



Manual de Referência de Scripts do E3

Sumário

1	Introdução	6
1.1	Objetos	6
1.2	Scripts	7
1.3	Picks	12
1.4	Eventos Definidos pelo Usuário	20
2	Programando no E3	23
2.1	Ambiente de Programação	23
2.2	Declarando Variáveis	24
2.3	Obtendo referências aos objetos	24
2.4	Acessando Objetos	39
2.5	Trabalhando com Coleções	40
2.6	Comando Set	41
2.7	Eventos, Métodos e Propriedades Gerais dos Objetos	42
3	Drivers	52
3.1	Driver de Comunicação	52
3.2	Tag de Comunicação	63
3.3	Bloco de Comunicação	73
3.4	Elemento de Bloco de Comunicação	78
3.5	Driver de Comunicação OPC	85
3.6	Tag de Comunicação OPC	90
3.7	Bloco de Comunicação OPC	100
3.8	Elemento de Bloco de Comunicação OPC	106
4	Objetos em Tempo de Execução do Servidor	116
4.1	Servidor	116
4.2	Pastas de Aplicativo	118
5	Servidor de Dados	120
5.1	Pasta de Dados	120
5.2	Tag Contador	120
5.3	Tag Demo	122
5.4	Tag Interno	125
5.5	Tag Timer	127
6	Telas e Objetos de Tela	129
6.1	Tela	129
6.2	Objetos de Tela	141
7	ActiveX	177
7.1	Métodos Comuns	177
7.2	Propriedades Comuns	178
7.3	Microsoft Forms	180
7.4	E2Controls	245
8	Quadros e Divisores	272
8.1	Métodos do Divisor	272
8.2	Propriedades do Divisor	277

8.3 Propriedades do Quadro	279
9 Associações	280
9.1 Métodos Comuns	280
9.2 Propriedades Comuns	283
9.3 Conexão Simples	284
9.4 Conexão Bidirecional	285
9.5 Conexão Reversa	285
9.6 Conexão Digital	286
9.7 Conexão Analógica	288
9.8 Conexão Por Tabela	290
9.9 Conexão Múltipla	294
10 Viewer	296
10.1 Eventos	296
10.2 Métodos	297
10.3 Propriedades	315
10.4 Objeto _top	322
11 Banco de Dados	329
11.1 Métodos	329
11.2 Propriedades	329
12 Alarmes	333
12.1 Configuração de Alarmes	333
12.2 Áreas de Alarme	333
12.3 Fontes de Alarme	335
12.4 Servidor de Alarmes	351
13 E3Alarm	359
13.1 Eventos	359
13.2 Métodos	360
13.3 Propriedades	366
14 Históricos	374
14.1 Métodos	374
14.2 Propriedades	375
15 Storage	380
15.1 Métodos	380
15.2 Propriedades	381
15.3 Campo do Storage	385
16 Consultas	387
16.1 Eventos	387
16.2 Métodos	387
16.3 Propriedades	392
17 E3Browser	399
17.1 Eventos	399
17.2 Métodos	400
17.3 Propriedades	401
17.4 Campos do E3Browser	408
18 E3Chart	410

18.1	Eventos	410
18.2	Métodos	411
18.3	Propriedades	414
18.4	Coleção de Penas	429
18.5	Penas	431
18.6	Coleção de Eixos	442
18.7	Eixos	444
18.8	Coleção de Consultas	451
18.9	Legenda	453
18.10	Colunas da Legenda	457
19	E3Playback	460
19.1	Métodos	460
19.2	Propriedades	461
20	Fórmulas	462
20.1	Métodos	462
20.2	Propriedades	466
21	Bibliotecas de Usuário	468
21.1	XControls e XObjects	468
21.2	Propriedades de um EllipseX	470
22	Relatórios	471
22.1	Eventos	471
22.2	Métodos	474
22.3	Propriedades	478
22.4	Layout	479
22.5	Seção	484
22.6	Objetos	495
23	Dúvidas Mais Frequentes	519

CAPÍTULO

1 Introdução

Os **Scripts** são módulos de linguagem de programação nos quais se pode criar procedimentos associados a eventos específicos, permitindo uma maior flexibilidade no desenvolvimento de aplicações. Cada objeto (item de um aplicativo) do E3 possui uma lista de eventos previamente definidos, mas também é possível definir novos eventos próprios do usuário.

1.1 Objetos

Os **Objetos** são componentes de software reutilizáveis que permitem maximizar o uso e aumentar a qualidade e produtividade em seus aplicativos.

Um objeto no E3 encapsula ou contém três diferentes partes (propriedades, métodos e eventos) que podem ser manipuladas para a utilização das vantagens de sua funcionalidade na aplicação.

Propriedades definem atributos de um objeto, como a aparência de um objeto de Tela ou o valor inicial de um objeto quando o aplicativo é iniciado.

Métodos são funções que realizam uma ação específica dentro ou com um objeto.

Eventos são notificações geradas por um objeto em resposta a alguma ocorrência em particular, como um clique de mouse ou uma mudança no valor de um tag, entre outros.

Uma das principais características dos objetos e das linguagens orientadas a objeto é a capacidade de herança entre eles, ou seja, estes podem herdar as características de um ou mais objetos, tendo as mesmas funcionalidades específicas. Assim, pode-se ter vários objetos trabalhando em conjunto para prover características de um outro objeto derivado.

Tome como exemplo o objeto E3Chart. Ele é composto internamente de vários objetos, como títulos, legendas, escalas, divisões, consultas e penas. Note que cada objeto contribui para a funcionalidade do todo: as escalas ajudam a localizar os valores dos pontos no E3Chart; as legendas, a identificar a pena e seus valores; e a pena realiza o desenho dos valores no E3Chart.

Através da manipulação de objetos dentro do E3Chart, pode-se criar duas instâncias deste objeto bastante diferentes entre si. Para se manipular um objeto específico, deve-se acessá-lo através de uma hierarquia. Se os dois objetos E3Chart estiverem na mesma Tela, deve-se acessar primeiro a Tela, depois o E3Chart desejado, depois uma de suas propriedades ou objetos-filhos. Quando há vários objetos de um mesmo tipo, geralmente eles podem ser acessados através de uma coleção (*collection*). Uma coleção é um objeto especial que gerencia um conjunto de objetos semelhantes. Um exemplo no objeto E3Chart é a coleção Pens, que permite o acesso a todas as penas do E3Chart.

1.2 Scripts

A linguagem que o E3 Studio usa em seus scripts é o **VBScript**, um subconjunto da linguagem Visual Basic® desenvolvida pela Microsoft. O VBScript possui um interpretador rápido, leve e portátil, desenvolvido para uso em navegadores para a Internet e outras aplicações que usam *ActiveX Controls, Automation Servers e Java Applets*.

Conforme visto anteriormente, os scripts são associados a eventos de um objeto. Entretanto, para facilitar e aumentar a velocidade de desenvolvimento, o E3 já incorpora algumas ações mais comuns que poderiam ser realizadas com scripts através de assistentes chamados **Picks**. Pode-se portanto definir que um determinado evento executará um script, um *pick* ou uma combinação deles, numa sequência também pré-definida.

Cada *view* do E3 Studio apresenta pelo menos duas abas na parte inferior: **Design** e **Scripts**, com exceção dos objetos Banco de Dados e Servidor de Alarmes, que não possuem a aba **Design**. Os objetos e seus filhos podem ser manipulados na aba **Design**; para manipular seus scripts, utilize a aba **Scripts**. Os botões disponíveis nesta última são:

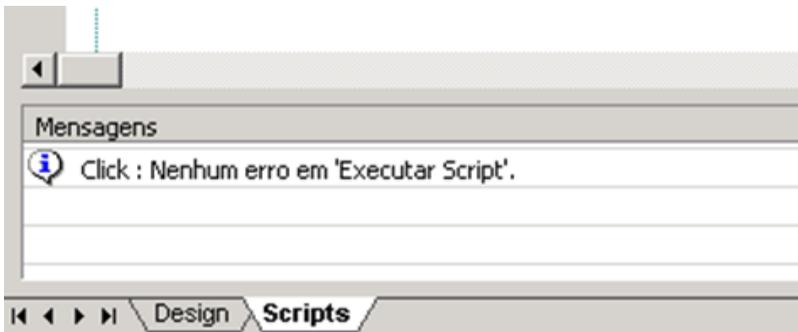
Opções disponíveis na aba Scripts

CAMPO	ÍCONE	AÇÃO/FUNÇÃO
Lista de Objetos		Seleciona o objeto cujo script será manipulado.
Lista de Eventos		Seleciona o evento a ser aplicado ao objeto.
Script		Adiciona um script associado ao evento.
Pick Abrir Tela		Adiciona o pick Abrir Tela .
Pick Abrir Tela Modal		Adiciona o pick Abrir Tela Modal .

CAMPO	ÍCONE	AÇÃO/FUNÇÃO
Pick Executar Aplicação		Adiciona o pick Executar Aplicação .
Pick Carregar Valor		Adiciona o pick Carregar Valor .
Pick Inverter Valor		Adiciona o pick Inverter Valor .
Pick Imprimir Relatório		Adiciona o pick Imprimir Relatório .
Remover script/pick selecionado		Remove o script ou o pick selecionado na Lista de Ações.
Mover script/pick selecionado para cima		Move a ação selecionada para cima, na ordem da Lista de Ações para o evento.
Mover script/pick selecionado para baixo		Move a ação selecionada para baixo, na ordem da Lista de Ações para o evento.
AppBrowser		Abre a janela do AppBrowser.
Procurar		Busca por ocorrências de um determinado texto.
Procurar anterior		Seleciona na lista de resultados a ocorrência anterior.
Procurar próxima		Seleciona na lista de resultados a ocorrência seguinte.
Substituir		Substitui as ocorrências encontradas por outro texto especificado.
Criar evento do usuário		Cria um evento de usuário.
Remover evento do usuário		Remove o evento de usuário selecionado.
Editar evento do usuário		Edita o evento de usuário selecionado.
Compila o script selecionado		Compila o script selecionado, mostrando os erros no painel Mensagens.
Compilar todos os scripts deste evento		Compila todos os scripts associados ao evento.
Compila todos os eventos		Compila todos os eventos associados ao

CAMPO	ÍCONE	AÇÃO/FUNÇÃO
deste objeto		objeto.

A ordem de execução das ações é de cima para baixo. Para alterar a ordem, pode-se utilizar os botões  e . Utilize o botão  para verificar se não há erros no script especificado para o evento. As mensagens de erro do compilador são apresentadas no painel **Mensagens**, na parte inferior da aba **Scripts**. Dê um duplo-clique no erro caso deseje que ele seja selecionado no script.

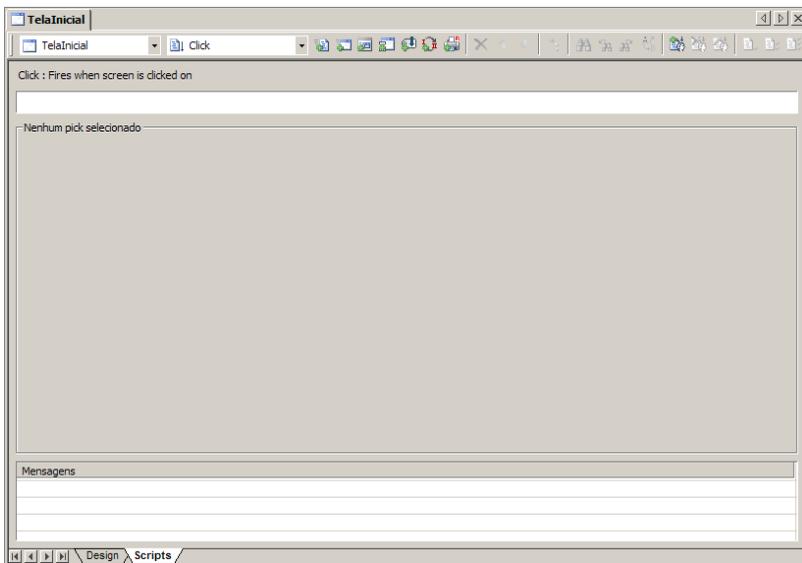


Mensagem do compilador

1.2.1 Adicionando um Script

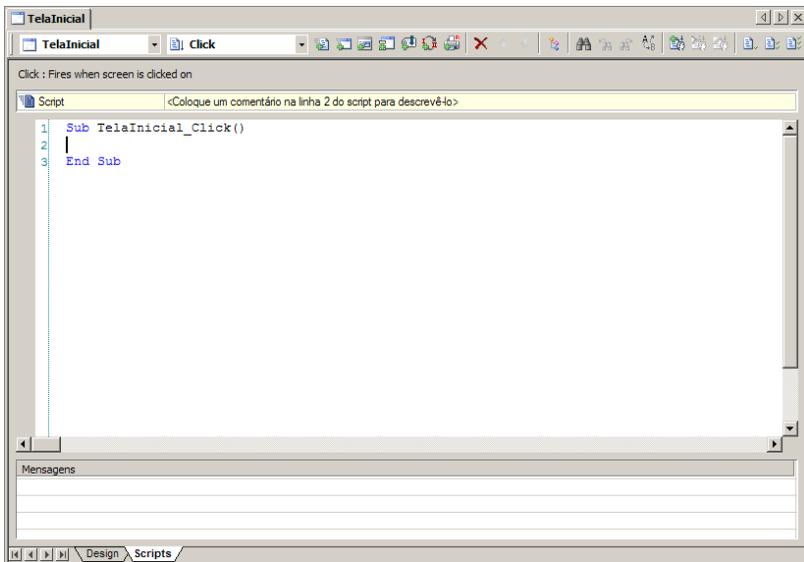
Para adicionar um script a um objeto, siga estes passos:

1. Selecione o objeto para o qual se deseja criar o script e clique na aba **Scripts**.



Aba Scripts

2. Clique sobre o ícone . O editor de scripts é aberto, conforme a figura a seguir.



Adicionando um script ao objeto

3. Entre com os comandos VBScript desejados na caixa de edição de texto.

NOTA: Utilize o caractere de sublinhado quando desejar acrescentar uma quebra de linha para deixar o código mais legível. O caractere de sublinhado indica que o código continua na linha subsequente.

Por exemplo:

```
If intTemperaturaCaldeira3 > 120 and _  
    intTemperaturaCaldeira4 > 120 Then  
    bEnviarAlarme = True  
    bAlarmeLigado = True  
End If
```

Cada evento pode ter vários scripts e vários *picks* associados, chamados **Ações do Evento**. A lista das ações pode ser vista na parte superior da janela de edição de scripts. Cada objeto pode possuir qualquer quantidade de eventos com scripts ou *picks* associados.

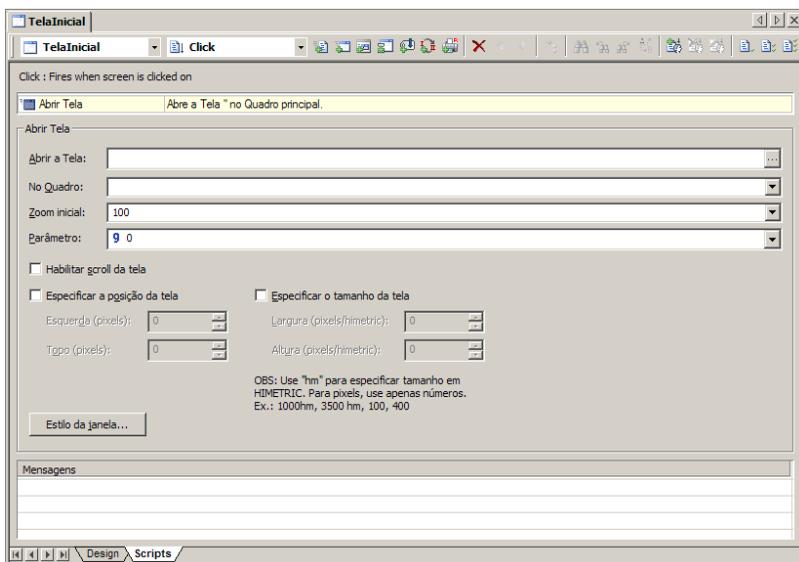
NOTA: Clicando com o botão direito do mouse sobre qualquer destas ações descritas acima abre um menu que possibilita recortar, copiar e colar scripts e picks entre eventos.

1.3 Picks

Os **Picks** implementam uma forma mais amigável de realizar procedimentos mais comuns, poupando tempo de configuração. Entre eles, estão ações como troca de Tela ou atribuição de valores, bastante comuns na criação de um projeto. A seguir são descritos os *picks* disponíveis na aba **Scripts**.

1.3.1 Abrir Tela

Abre uma determinada Tela ou Quadro.



Configurações do pick Abrir Tela

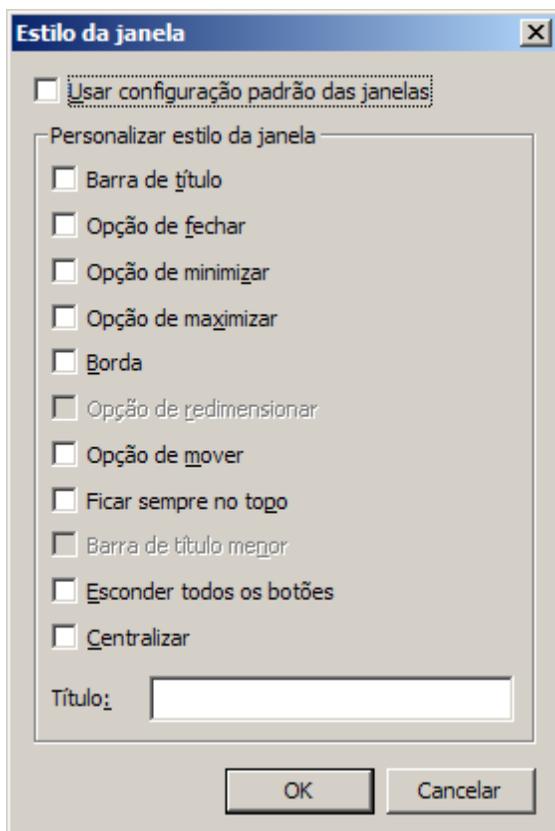
Opções disponíveis no pick Abrir Tela

CAMPO	DESCRIÇÃO
Abrir a tela	Indica qual Tela deve ser aberta.
No quadro	Indica o Quadro onde a Tela será visualizada. Se estiver em branco, será o Quadro principal (_top).
Zoom inicial	Define o zoom da Tela, quando mostrada.

CAMPO	DESCRIÇÃO
Parâmetro	Indica um parâmetro a ser passado na chamada da Tela.
Habilitar scroll da tela	Habilita o uso das barras de rolagem na Tela.
Especificar posição da tela	Indica a posição da Tela, em pixels.
Especificar tamanho da tela	Indica o tamanho da Tela, em pixels ou HIMETRIC.
Estilo da janela	Chama a caixa de diálogo Estilo de janela .

1.3.1.1 Caixa de Diálogo Estilo de Janela

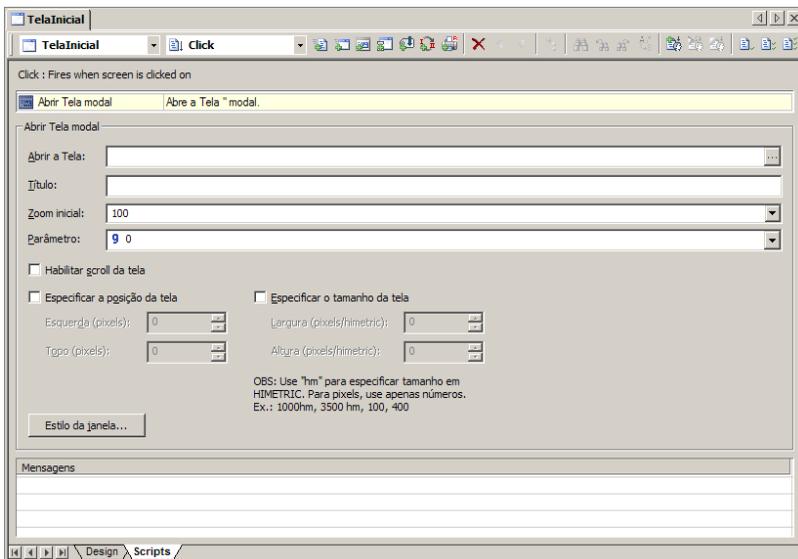
Permite configurar o estilo da janela a ser mostrada, definindo título e a disponibilidade das bordas e botões de fechar, maximizar e minimizar, entre outros. Se a opção **Usar configuração padrão das janelas** for marcada, o sistema desabilita as opções desta janela e assume a configuração padrão do Viewer, conforme o que pode ser visto na aba **Visualizador** das propriedades do objeto Viewer.



Caixa de diálogo Estilo de Janela

1.3.2 Abrir Tela Modal

Abre uma Tela Modal, isto é, uma Tela que não permite a interação do usuário com outras Telas enquanto esta estiver ativa.



Configurações do pick Abrir Tela Modal

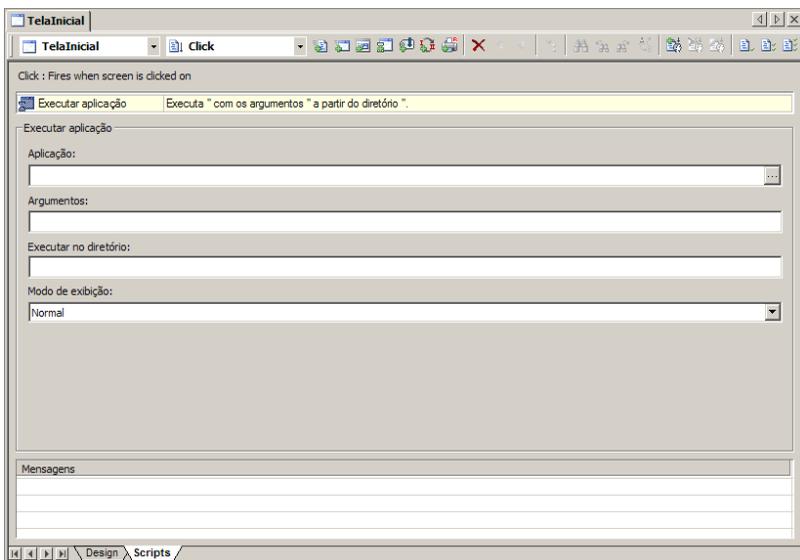
Opções disponíveis no pick Abrir Tela Modal

CAMPO	DESCRIÇÃO
Abrir a tela	Seleciona a Tela a ser aberta.
Título	Define o título da janela. Esse texto será concatenado com o nome da Tela.
Zoom inicial	Define o zoom da Tela, quando mostrada.
Parâmetro	Indica um parâmetro a ser passado na chamada da Tela.
Habilitar scroll da tela	Habilita o uso das barras de rolagem na Tela.
Especificar a posição da tela	Determina a posição (em pixels) do quadro na Tela, a partir do canto superior esquerdo da Tela.

CAMPO	DESCRIÇÃO
Especificar o tamanho da tela	Determina a largura e a altura (em pixels ou HIMETRIC) da Tela.
Estilo da janela	Permite configurar o estilo da janela a ser mostrada, definindo título e a disponibilidade das bordas e botões de fechar, entre outros (ver a Caixa de Diálogo Estilo de Janela).

1.3.3 Executar Aplicação

Executa uma aplicação específica.



Configurações do pick Executar Aplicação

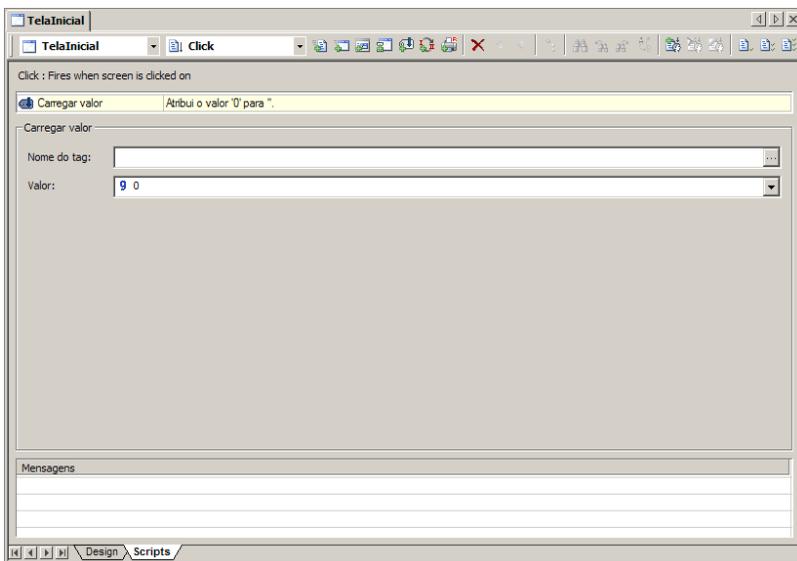
Opções disponíveis no pick Executar Aplicação

CAMPO	DESCRIÇÃO
Aplicação	Apertando-se o botão  pode-se navegar no disco para indicar o arquivo da aplicação a ser executada.
Argumentos	Permite especificar argumentos para a chamada da aplicação.

CAMPO	DESCRIÇÃO
Executar no diretório	Determina o diretório de trabalho da aplicação que será executada.
Modo de exibição	Determina o tipo de janela de execução da aplicação: normal, minimizado ou maximizado.

1.3.4 Carregar Valor

Carrega um valor em um tag.



Configurações do pick Carregar Valor

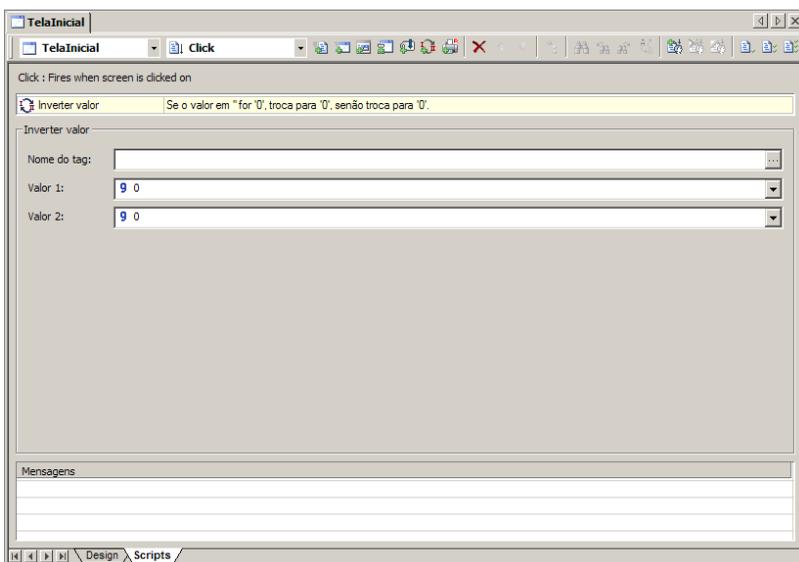
Opções disponíveis no pick Carregar Valor

CAMPO	DESCRIÇÃO
Nome do tag	Especifica o nome do tag no qual será carregado o valor. Pode-se escolher o tag no AppBrowser, apertando-se o botão  .
Valor	Determina o valor que será carregado no tag. Pode-se escolher o tipo de dados apertando a seta para baixo.

1.3.5 Inverter Valor

Permite inverter o valor de um tag. Se o valor do tag for igual a *Valor1*, então o tag recebe *Valor2*. Se o valor do tag for igual a *Valor2*, então o tag recebe *Valor1*. Se o valor do tag não for igual a *Valor1* nem *Valor2*, o tag recebe *Valor1*.

É possível colocar tantos picks **Inverter Valor** quanto o desejado. Isso permite a verificação de múltiplos valores para um mesmo tag ou até mesmo para diversos tags em um mesmo evento.



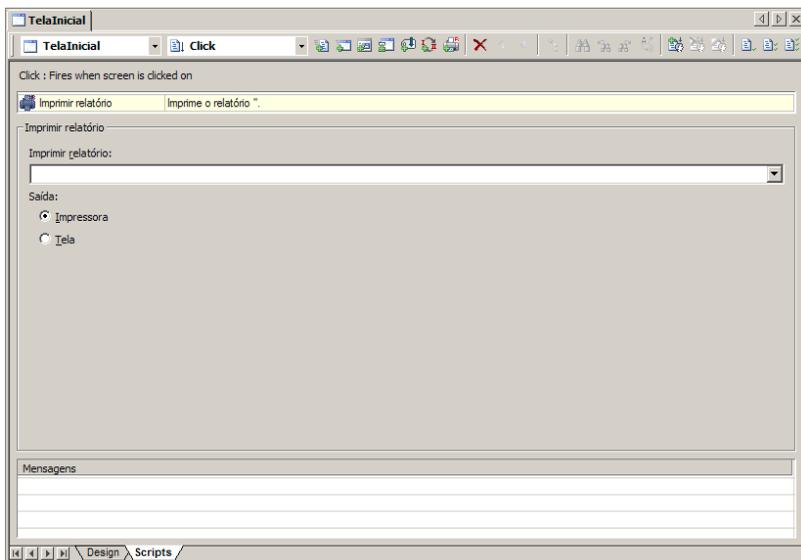
Configurações do pick Inverter Valor

Opções disponíveis no pick Inverter Valor

CAMPO	DESCRIÇÃO
Nome do tag	Apertando-se o botão  o E3 abre o AppBrowser, para a escolha do tag desejado.
Valor 1	Determina o primeiro valor a ser comparado. Se o valor do tag for igual a <i>Valor1</i> , então o tag recebe <i>Valor2</i> .
Valor 2	Determina o segundo valor a ser comparado. Se o valor do tag for igual a <i>Valor2</i> , então o tag recebe <i>Valor1</i> .

1.3.6 Imprimir Relatório

Permite imprimir um relatório na tela ou na impressora.



Configurações do pick Imprimir Relatório

Opções disponíveis no pick Imprimir Relatório

CAMPO	DESCRIÇÃO
Imprimir relatório	Permite escolher o relatório a ser impresso.
Saída	Determina o tipo de saída do relatório: <ul style="list-style-type: none">• Impressora: Envia o relatório para a impressora. Corresponde ao método Print.• Tela: Faz uma prévia de impressão de relatório na tela. Corresponde ao método PrintPreview.

1.4 Eventos Definidos pelo Usuário

Apesar do E3 vir com uma extensa gama de eventos, muitas vezes o usuário pode desejar criar um evento específico para sua aplicação.

Um exemplo para a utilização de eventos definidos pelo usuário seria a realização de um cálculo ou tarefa mais complexa em um objeto, quando o evento gerador vem de um outro tag ou propriedade.

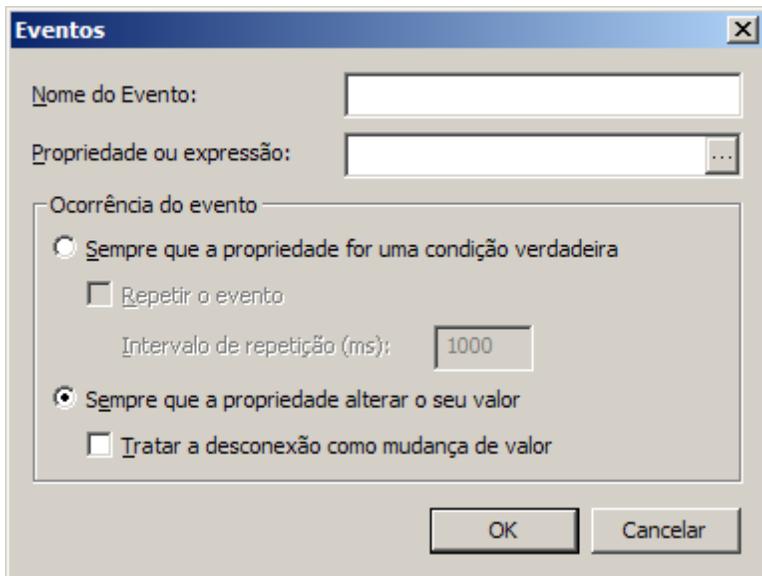
Apesar de se poder criar e executar este mesmo tipo de trabalho a partir do próprio tag ou da propriedade geradora do evento, existem vantagens em manter-se o script junto ao objeto que sofrerá a ação deste. Dentre elas, podemos citar o trabalho adicional necessário para fazer um objeto apontar para outro, além de facilidades de manutenção, pois se por algum motivo for necessário modificar ou apagar o tag ou a propriedade que seja o evento gerador, não é necessário modificar um segundo objeto.

Outra vantagem é o fato de que se o tag gerador do evento for apagado, o objeto não perderá o script, basta que se indique uma outra fonte geradora de evento.

A geração de eventos internos facilita ainda a criação de bibliotecas, pois cada vez que um componente de biblioteca é inserido em uma aplicação, ele traz consigo os scripts e cálculos que podem ser necessários, diminuindo o trabalho de configuração.

Para gerar um novo evento interno em um objeto, siga estes procedimentos:

1. Clique no botão **Criar evento do usuário** , ou então em **Criar novo evento** , na Lista de Eventos. O E3 abrirá uma janela para a definição das propriedades do evento.



Janela para a adição de eventos definidos pelo usuário

Opções disponíveis na janela Eventos

OPÇÃO	DESCRIÇÃO
Nome do Evento	Nome que identifica o evento.
A propriedade/expressão	Expressão geradora do evento. Pode ser copiada através do AppBrowser, pressionando-se o botão  .
For uma condição verdadeira	Indica que o evento será do tipo etOnEvent (o evento ocorre no momento que a expressão for verdadeira) ou etWhileEvent (o evento ocorre ciclicamente, em intervalos pré-definidos).
Repetir o evento a cada ... ms	Quando preenchido, indica que o evento será etWhileEvent . Indica o ciclo de repetição do evento em milissegundos, isto é, de quanto em quanto tempo ele irá ocorrer, enquanto a expressão geradora for verdadeira.
Alterar o seu valor	Indica que o evento será etOnValueChanged , isto é, o evento irá ocorrer sempre que a expressão geradora

OPÇÃO	DESCRIÇÃO
	mudar o valor.
Tratar a desconexão como mudança	Indica se a conexão ou desconexão da expressão geradora do evento deverá ser tratada como mudança.

2. Clique em **OK** para completar o processo e inserir o evento. Ele irá aparecer na lista de eventos.
3. Para alterar este evento, selecione-o e clique em **Editar evento do usuário** . A janela anterior será aberta novamente para a edição de dados do evento.
4. Para apagar este evento, selecione-o e clique em **Remover evento do usuário** .

IMPORTANTE: Quando se clica em Remover evento do usuário, os scripts deste evento serão perdidos.

Apesar de a maioria dos aspectos do VBScript se aplicarem à programação de scripts no E3, algumas particularidades devem ser destacadas no que diz respeito às implementações do conceito de orientação a objetos no sistema.

O **VBScript** é uma linguagem baseada no Visual Basic que traz a capacidade de *scripting* (roteirização) para aplicações que rodam no sistema operacional Windows.

O VBScript troca informações com as aplicações utilizando a tecnologia *ActiveX Scripting*. Com o *ActiveX Scripting*, navegadores e outras aplicações cliente como o E3 Viewer podem compilar scripts e chamar funções, dentre outros procedimentos. Isso possibilita que scripts desenvolvidos para uma aplicação ou biblioteca que devam ser executados na interface gráfica possam ser executados tanto no E3 Viewer quanto em um navegador de Internet, sem a necessidade de qualquer adaptação do aplicativo.

Mais informações sobre o VBScript podem ser obtidas no manual **Guia de Referência do VBScript**, no Grupo de Programas do E3.

2.1 Ambiente de Programação

O ambiente de programação de scripts no E3 Studio pode ser acessado através do clique com o botão direito do mouse sobre qualquer objeto, escolhendo-se a opção **Propriedades**. Na aba **Scripts** da *view* do objeto, pode-se ver a caixa combo onde se define qual evento será o gerador do script. Como visto no capítulo anterior, existem dois tipos de eventos em um objeto do E3, os pré-definidos e os definidos pelo usuário.

Os eventos pré-definidos variam de objeto para objeto, dependendo da sua utilização e funcionalidade. Um objeto de tela, por exemplo, possui eventos relacionados à interface gráfica, como **OnClick** (chamado ao clicar sobre o objeto) ou **OnDbClick** (chamado com um duplo clique); já um objeto como um Driver de Comunicação possui eventos relacionados à comunicação, como **OnCommError** (chamado quando há um erro de comunicação). Pode-se também definir outros eventos para o objeto, como visto anteriormente.

Quando um script é associado a um evento em um objeto, o campo de digitação apresenta uma declaração de procedimento (*procedure*), cuja definição é automática

e composta do seguinte texto:

```
Sub NomeDoObjeto_NomeDoEvento()  
End Sub
```

Onde `NomeDoObjeto` é o nome do objeto associado e `NomeDoEvento` é o nome do evento em questão. Os comandos do script deverão estar entre essas duas linhas.

Para ajudar na digitação do script, pode-se utilizar o AppBrowser. Ao se escolher o método ou propriedade desejada, pode-se acessar o botão **Copiar**. O tag, propriedade ou método escolhidos serão inseridos na posição do cursor no quadro de edição do script. O local do cursor é mostrado através da animação com uma seta piscante.

2.2 Declarando Variáveis

É possível declarar variáveis de duas maneiras, implicitamente ou explicitamente.

Para declarar uma variável implicitamente, basta usar seu nome no script. Automaticamente, a variável será criada e inicializada com o valor da atribuição, ou permanecerá EMPTY (vazia, sem nenhum valor), caso não receba nenhum valor antes de ser utilizada.

Esta é uma prática rápida, porém se o script for muito extenso, isto pode causar confusão e a criação de mais de uma variável com o mesmo nome, gerando *bugs* no script.

Para declarar variáveis explicitamente, utiliza-se o comando **Dim**, como no exemplo:

```
Dim Temperatura
```

Pode-se declarar múltiplas variáveis separando cada nome de variável com uma vírgula. Por exemplo:

```
Dim Esquerda, Direita, Topo, Base
```

Em virtude dos scripts no E3 serem todos associados a um objeto em particular, as variáveis são sempre locais, válidas apenas para o escopo do script. Para ter-se variáveis públicas ou globais, deve-se criar um tag interno e utilizá-lo para armazenar o valor desejado.

2.3 Obtendo referências aos objetos

Uma das características mais importantes ao se trabalhar com scripts dentro do E3 é considerar a separação existente entre os processos que são executados no servidor e aqueles executados na interface do cliente (E3 Viewer). Para se trabalhar com scripts, pode-se manipular:

- Objetos do servidor através do servidor.

- Objetos do servidor através do(s) E3 Viewer(s).
- Objetos do E3 Viewer através do mesmo E3 Viewer.

No entanto, não se pode manipular diretamente:

- Objetos do E3 Viewer através do servidor (só é possível através da criação de eventos no E3 Viewer, ligados a variáveis que estão no servidor).
- Objetos de um E3 Viewer a partir de outro E3 Viewer (só é possível através da criação de eventos ligados a variáveis que estão no servidor).

Tais limitações são decorrentes do fato de que, por definição, existe uma independência entre o que cada uma das estações E3 Viewer está fazendo ou visualizando e o servidor e vice-versa. Por isso, todas as atividades, tanto do servidor como do E3 Viewer, necessitam ser coordenadas de forma assíncrona ou através de eventos para operarem de forma harmoniosa.

Logo, devido a esta independência, ao se criar um script, primeiro deve-se obter uma referência correta dos objetos que se deseja manipular, ou seja, é necessário que o objeto seja primeiramente encontrado nos vários módulos do E3.

Vale repetir que, ao se editar um script, o usuário poderá utilizar o AppBrowser, que permite copiar para o script o caminho de um método ou propriedade de forma completa, auxiliando-o na criação de scripts.

Portanto, para acessar os objetos externos que estão sendo manipulados em um script, são utilizadas algumas diretivas básicas. Por exemplo, para se manipular o valor de um Tag de Comunicação, o caminho é **Servidor - Driver - Pasta (se houver) - Tag**. Já se o objetivo é manipular um botão na Tela, o caminho é **Viewer - Quadro (se houver) - Tela - Botão**.

Existem basicamente três localizações de origem de scripts, do ponto de vista da metodologia para acesso dos objetos:

- Servidor (E3 Server).
- Telas e Quadros (E3 Viewer).
- ElipseX (bibliotecas): podem ser XObjects (rodam no servidor) e XControls (rodam no E3 Viewer).

2.3.1 Acessando Propriedades do Servidor

Para se acessar um objeto que está sendo executado no servidor a partir de um Objeto de Tela ou um EclipseX, deve-se usar a diretiva **Application.GetObject**.

A palavra **Application** representa a aplicação como um todo, e o método **GetObject** procura no servidor por um objeto dentro de **Application** com o nome fornecido.

Exemplo:

```
Sub Button1_Click()  
    Application.GetObject("Driver1")._  
        Item("tag001").AllowRead = False  
End Sub
```

ou ainda

```
Sub Button1_Click()  
    Application.GetObject("Driver1.tag001").AllowRead = False  
End Sub
```

O método **Item** foi utilizado para, a partir da referência de **Driver1**, localizar o **tag001**, pois o **Driver** é uma coleção de **Tags**. Depois de localizado o objeto, suas propriedades e funções podem ser acessadas livremente.

Caso alguma outra operação tivesse que ser realizada com o **Driver1** ou **tag001**, outra alternativa para o script acima seria:

```
Sub Retangulo1_Click()  
    Set obj = Application.GetObject("Driver1")  
    obj.Item("tag001").AllowRead = False  
    obj.Item("tag002").AllowRead = False  
End Sub
```

Neste caso, a variável **obj** está apontando para o objeto **Driver1** e na próxima vez que se quiser acessar algum objeto que descende de **Driver1** dentro do script, pode-se utilizar a variável **obj** diretamente. Isso traz um ganho de performance, já que cada chamada do método **GetObject** faz um acesso ao servidor. Através dessa técnica, chamadas desnecessárias ao servidor são evitadas. Este exemplo usa o comando **Set**, que será explicado mais adiante. Note que a utilização de variáveis também torna o código mais claro e de modificação mais fácil. Caso seja necessário alterar o objeto no qual queremos executar comandos, basta mudar a linha de atribuição dessa variável.

A palavra **Application** nos scripts pode indicar tanto funções que são executadas no E3 Viewer quanto no servidor. No caso, o objeto **Application** sabe de antemão quais funções devem ser executadas tanto para um quanto para outro caso. Não é possível, entretanto, executar funções de E3 Viewer dentro do servidor, assim como também não é possível executar funções de servidor dentro do E3 Viewer.

2.3.2 Acessando Propriedades do Studio

Para acessar um objeto de servidor qualquer num script que roda no Studio pode-se usar a diretiva **Application.GetObject**. A palavra **Application** representa a aplicação como um todo, e o método **GetObject** procura no Domínio carregado no Studio por um objeto de servidor com o caminho fornecido. Exemplo (o evento **CustomConfig** é disparado no Studio):

```
Sub XControl1_CustomConfig
    Application.GetObject("Dados.TagDemo1").DocString = "Documentação"
End Sub
```

2.3.3 Acessando Propriedades do Servidor de dentro do Servidor

Caso seja necessário acessar as propriedades de um Tag a partir de outro, origem e destino estão no servidor ligados via um módulo-pai que é **Driver1**.

Nesta situação, deve ser usada a declaração **Parent**. Isto faz com que seja acessado primeiro o objeto-pai onde está o script, para depois descer-se na hierarquia em busca de outro elemento.

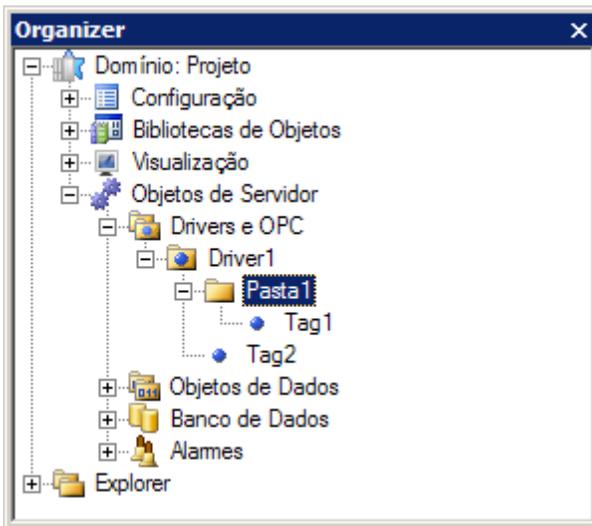


Driver1 é o objeto pai de Tag1 e de Tag2

Exemplo:

```
Sub Tag1_OnRead()
    Parent.Item("Tag2").AllowRead = False
End Sub
```

Estando-se dentro de um grupo, e desejando-se acessar o mesmo Tag2, pode-se aninhar vários comandos **Parent**.



Pasta1 é o objeto pai de Tag1

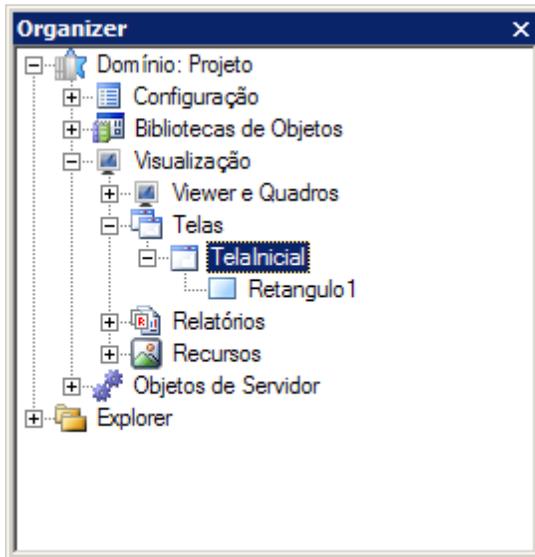
Exemplo:

```
Sub Tag1_OnRead()  
    Parent.Parent.Item("Tag2").AllowRead = False  
End Sub
```

2.3.4 Acessando Objetos de uma Tela a partir de um Script na Tela

Deve-se usar somente o método `Item`, já que os objetos são filhos da Tela. Exemplo:

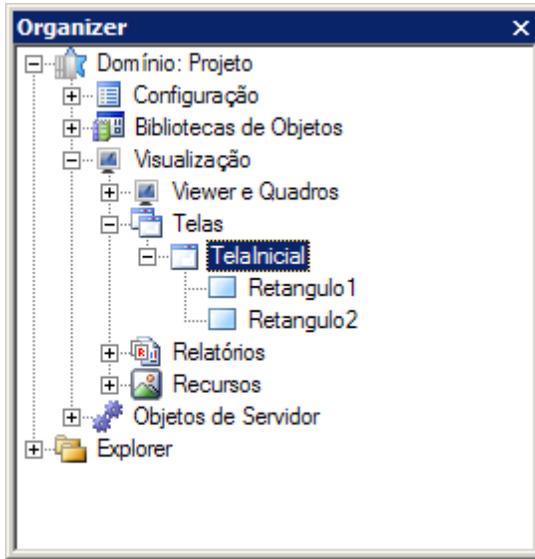
```
Sub Tela1_OnPreShow(vArg)
    Item("Retangulo1").Visible = True
End Sub
```



Retangulo1 é um item da TelaInicial

2.3.5 Acessando Objetos de uma Tela a partir de um Script em outro Objeto na Tela

Pode-se usar o método **Parent** ou o método **Screen**.



TelaInicial é o objeto pai de Retangulo1 e Retangulo2

Exemplo:

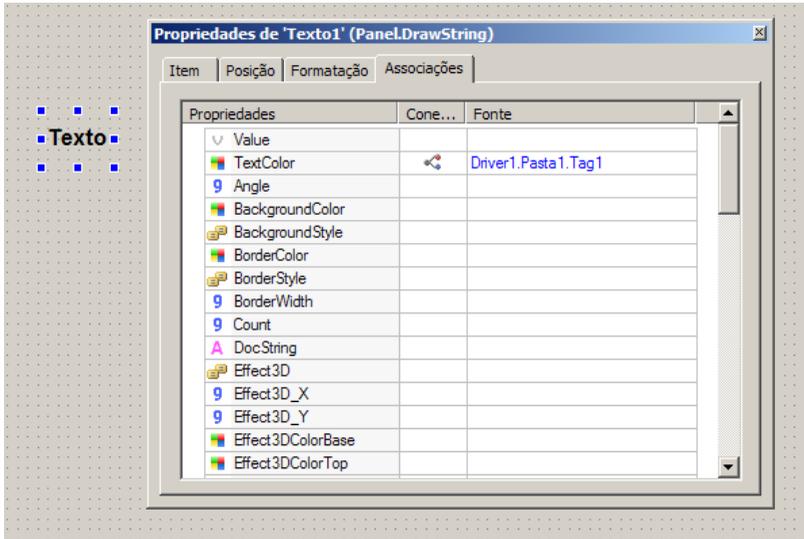
```
Sub Retangulo1_Click()  
    Parent.Item("Retangulo2").Visible = True  
End Sub
```

2.3.6 Modificando a Tela ou Objetos de Tela a partir do Servidor

A modificação de um comportamento qualquer em uma Tela só pode ser feita a partir de associações (o servidor reporta automaticamente para os Viewers todas as mudanças das variáveis escolhidas), ou via busca explícita do Viewer por informações no servidor. Toda a operação de associação da interface gráfica é feita do cliente para o servidor e não do servidor para o cliente. Assim, não é possível modificar Telas ou objetos a partir do servidor via scripts, pois cada cliente de dados é uma cópia diferente das Telas.

Um exemplo prático é mudar a cor de um texto na Tela para verde quando um tag for

ligado (valor 1) e para vermelho quando for desligado (valor 0). Neste caso, deve-se simplesmente criar uma associação digital entre a propriedade **TextColor** do **Text01** com o **Tag1**. Associações são preferíveis devido à rapidez de execução e simplicidade de manutenção e construção da aplicação.

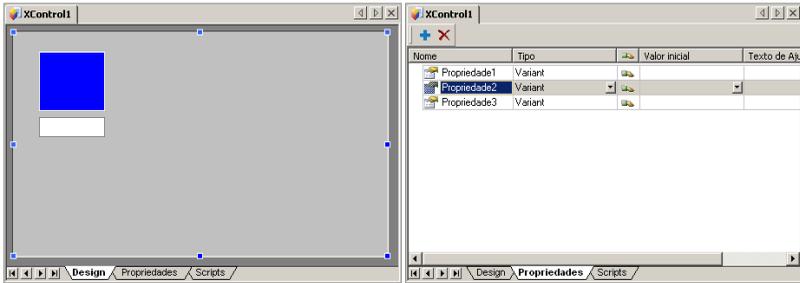


Associando a cor do texto ao valor de Tag1

Uma outra forma de executar o procedimento anterior é criar um script no Viewer que fique verificando constantemente se o **Tag1** mudou ou não de valor, para então mudar a cor do texto. Este tipo de script é possível de ser realizado, mas degrada muito a performance da aplicação. Por isto, esta prática **não** é aconselhável.

2.3.7 Acessando Objetos de um ElipseX a partir do próprio ElipseX

Na criação de um ElipseX, pode-se declarar propriedades (XProperties) e inserir objetos, que podem ser objetos de Tela (XControl) ou objetos do servidor (XObject). Para se acessar as XProperties através de scripts, basta acessar o nome da propriedade diretamente.



Acessando objetos de um ElipseX a partir do próprio ElipseX

Por exemplo, na figura anterior temos o `XControl1` com a propriedade **Propriedade1**, e os objetos `Texto1` e `Retangulo1`.

A propriedade **Propriedade1**, do tipo **Booleano**, pode ser acessada com a linha a seguir:

```
Sub XControl1_OnStartRunning()  
    XControl1.Propriedade1 = True  
End Sub
```

ou ainda

```
Sub XControl1_OnStartRunning()  
    Propriedade1 = True  
End Sub
```

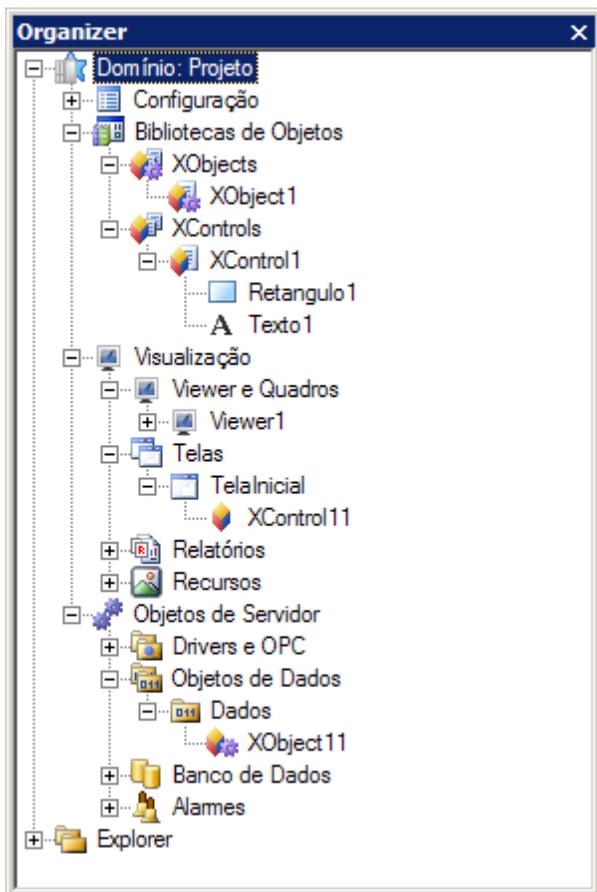
Se o ElipseX possui objetos internos, então é possível utilizar o método **Item** para obter uma referência destes objetos. Por exemplo:

```
Sub XControl1_OnStartRunning()  
    Item("Texto1").Value = "motor"  
    Item("Retangulo1").ForeColor = RGB(212, 208, 20)  
End Sub
```

2.3.8 Acessando Objetos de um ElipseX Externamente

O acesso externo a um objeto ElipseX só pode ser feito através de suas propriedades, utilizando suas instâncias criadas. Não é possível acessar objetos internos diretamente.

Se o ElipseX em questão for um XControl, ele se comporta como um objeto de Tela. Por exemplo, na seguinte aplicação:



XControl (exemplo)

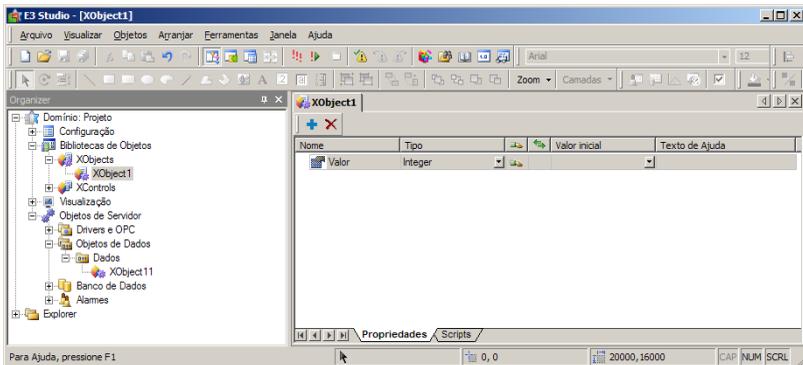
Para alterar a propriedade **Propriedade1** do XControl pode-se fazer o seguinte script no botão:

```
Sub CommandButton1_Click()
    Screen.Item("XControl11").Propriedade1 = True
End Sub
```

Ou ainda

```
Sub CommandButton1_Click()
    Parent.Item("XControl11").Propriedade1 = True
End Sub
```

No caso de um XObject, deve-se inserí-lo em um Servidor de Dados:



XObject (exemplo)

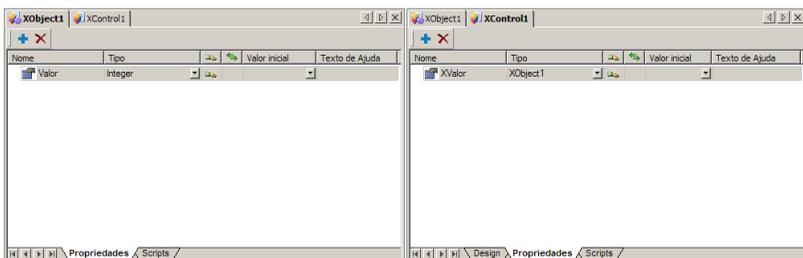
Um script para alterar a propriedade **Valor** do XObject seria

```
Sub CommandButton1_Click()
    Application.GetObject("Dados.XObject11").valor = 123
End Sub
```

Ou ainda

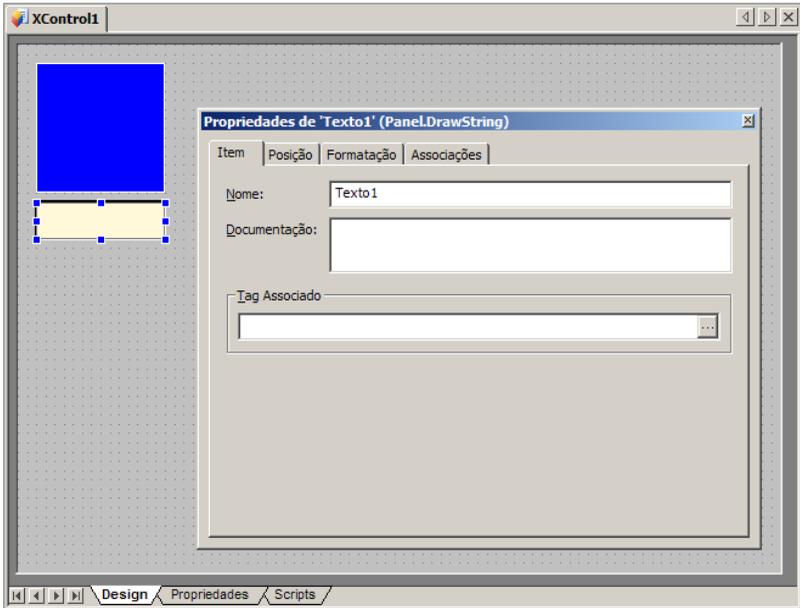
```
Sub CommandButton1_Click()
    Application.GetObject("Dados").Item("XObject11").valor = 123
End Sub
```

Pode-se ainda ter um XControl acessando um XObject, através de uma XProperty. Por exemplo, a figura mostra um XControl chamado XControl1 que possui uma propriedade **XValor** do tipo **XObject1**, que é o nome do XObject criado.



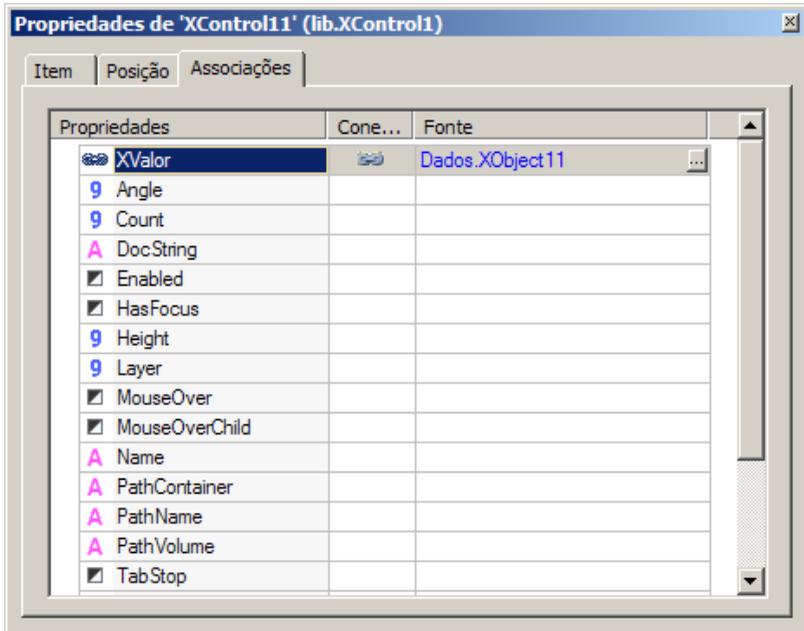
XControl acessando um XObject (exemplo)

Faz-se, por exemplo, uma associação do valor do objeto `Texto1` com a propriedade `Valor` do `XObject1`. Isso é feito através da propriedade `XValor`, criada no `XControl1`. Assim, o valor da propriedade `Valor` do `XObject1` será mostrada no objeto `Texto1` do `XControl1`.



Propriedade Valor

No projeto, o vínculo da instância `XObject11` à instância `XControl11` pode ser feito através de uma associação na propriedade `XValor`.

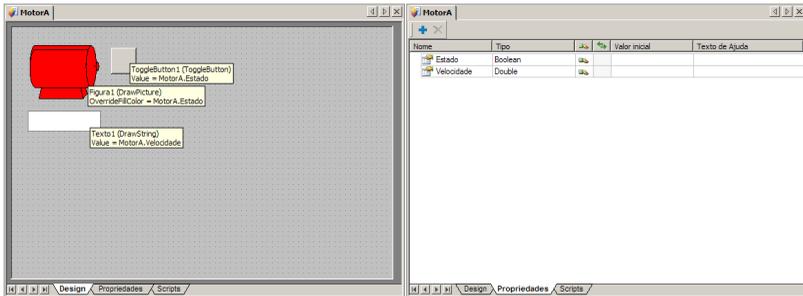


XValor (associação)

2.3.8.1 Exemplo de Criação de um ElipseX

Suponha que determinada aplicação necessite supervisionar e comandar 10 motores. Cada motor precisa ser representado por um desenho na Tela, que exibe a cor verde quando estiver operando, e vermelho quando estiver desligado. Também deve ser permitido o comando do motor na Tela, enviando instruções para ligar e desligar. Sua velocidade também deve ser exibida.

Uma possibilidade é a criação de um XControl chamado **MOTOR**, com as propriedades **Estado** igual a Boolean e **Velocidade** igual a Double, conforme a figura a seguir:

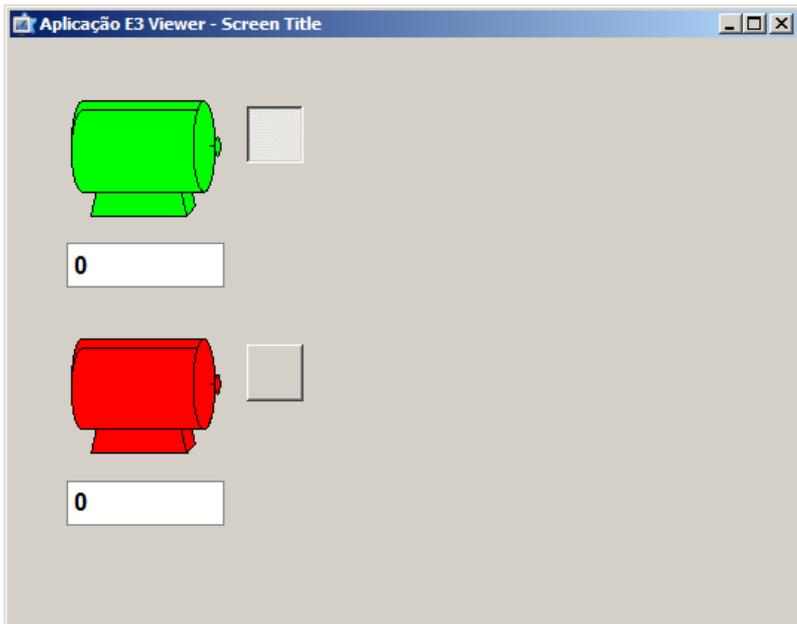


Definição do XControl

1. Para a indicação da cor, a propriedade **OverrideFillColor** do motor deve estar associada à propriedade **Estado** do XControl, através de uma associação digital. Configure a propriedade **OverrideFillMode** para **2 - SolidFill**.
2. Para exibir a velocidade, a propriedade **Value** do Display deve ser associada à propriedade **Velocidade** do XControl.
3. O Botão Liga-Desliga troca o valor da propriedade **Estado** através de uma associação simples.

Note que:

- As associações dentro da biblioteca são internas, e seu formato é **Nome_do_Controlo.Nome_da_Propriedade**.
- O objeto, depois de inserido na Tela, deverá ter essas propriedades associadas aos tags verdadeiros, para cada um dos motores.
- Uma associação de um tag para a propriedade **Estado** terá que ser feita para cada **MOTOR A**.

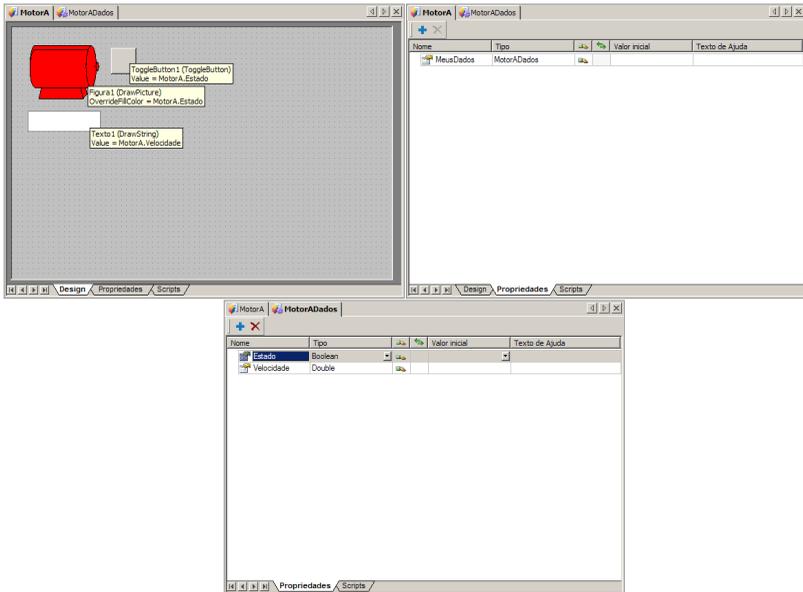


Viewer

Outra possibilidade, mais abrangente, utiliza um XObject para o motor. Desta forma, todas as informações relativas aos motores residem em objetos que ficam no servidor. Assim, podemos construir vários tipos de interface para o motor (XControls) que trazem do servidor, através do XObject, somente as informações necessárias.

Desta maneira, o objeto **MOTOR**A teria que ser modificado para apontar para um XObject, ao invés de declarar em si mesmo todas as propriedades.

1. Crie um XObject chamado **MOTOR**ADados, e declare nele as propriedades **Estado** e **Velocidade**.
2. Crie um XControl **MOTOR**A que possua apenas uma propriedade, chamada **MeusDados**, do tipo **MOTOR**ADados.
3. **MOTOR**ADados deverá ser inserido numa pasta de dados no servidor, e corresponderá a cada um dos motores. **MOTOR**A por sua vez irá apontar para o **MOTOR**ADados desejado, não sendo necessária a criação de novos tags.



Configuração na view do XObject

4. A propriedade **Estado**, associada à propriedade **OverrideFillColor** do motor, fica **MotorA.MeusDados.Estado**.
5. A propriedade **Velocidade**, associada ao Display, fica **MotorA.MeusDados.Velocidade**.

2.4 Acessando Objetos

Seguindo o conceito de encapsulamento de programação orientada a objetos, métodos e propriedades ficam associados a seus objetos de origem. Isto significa que sempre se deve indicar o objeto do qual se está acessando o método ou propriedade.

2.4.1 Propriedades

Para referenciar as propriedades de um objeto, deve-se usar o método **GetObject** do E3. A sintaxe é a seguinte:

```
Application.GetObject("<objeto>").<propriedade>
```

onde *<objeto>* é o nome do objeto e *<propriedade>*, a propriedade desejada.

Exemplo:

```
Application.GetObject("Dados.TempTanque2").Type
```

Para facilitar a digitação, é aconselhável sempre utilizar o AppBrowser, que já traz a sintaxe correta.

2.4.1.1 Propriedade Value

No E3, muitos objetos possuem uma propriedade em comum chamada **Value**. Neste caso específico, pode-se acessar esta propriedade utilizando-se o próprio nome do objeto:

```
Button1 = False
```

que é equivalente a:

```
Button1.Value = False
```

2.4.2 Métodos

A sintaxe a seguir exemplifica a chamada de um método que não precisa de parâmetros:

```
Application.GetObject("<objeto>").<metodo>
```

onde *<objeto>* é o objeto em questão e *<metodo>*, o método desejado.

Se o método aceita parâmetros, use a sintaxe a seguir:

```
Application.GetObject("<objeto>").<metodo>(<parametro>)
```

onde *<parametro>* é o parâmetro a ser passado para o método. Quando houver mais de um parâmetro, utilize vírgulas para separá-los.

Se o método retorna um resultado, e for desejável guardá-lo, então os parâmetros devem ser colocados obrigatoriamente entre parênteses:

```
<V> =Application.GetObject("<objeto>").<metodo>(<parametro>)
```

onde *<V>* é a variável que irá receber o resultado do método.

2.5 Trabalhando com Coleções

Uma coleção (ou *collection*) é um objeto que gerencia um conjunto de objetos similares. Os objetos contidos em uma coleção são referenciados por índices, semelhante à referência de arrays.

Pode-se adicionar ou remover objetos individuais de uma coleção, conforme o exemplo a seguir:

```

Sub CommandButton1_Click()
    ' Adiciona uma pena no objeto E3Chart1
    Screen.Item("E3Chart1").Pens.Add "Pena"
End Sub

Sub CommandButton2_Click()
    ' Remove a primeira pena
    Screen.Item("E3Chart1").Pens.Remove 0
End Sub

```

NOTA: o primeiro objeto em uma coleção possui o índice 1.

Todas as coleções possuem uma propriedade em comum chamada **Count**, que é o número de objetos (ou filhos) existentes. Exemplo:

```

Sub CommandButton1_Click()
    ' Mostra uma caixa de diálogo com o número de penas
    MsgBox Screen.Item("E3Chart1").Pens.Count
End Sub

```

2.5.1 Acessando Objetos com o Método Item

Toda coleção possui um método **Item**, que pode ser usado para acessar qualquer objeto dentro da coleção. O método **Item** aceita um parâmetro *Item*, que pode ser um número (inteiro positivo) ou o nome do objeto dentro da coleção que se deseja acessar.

Os exemplos a seguir são de ajustes da cor da segunda pena do objeto E3Chart:

```

Sub CommandButton1_Click()
    ' Modifica a cor da terceira pena
    Screen.Item("E3Chart1").Pens.Item(2).Color = RGB(212, 208, 20)
End Sub

```

Ou ainda

```

Sub CommandButton1_Click()
    ' Modifica a cor da pena com nome "Pena2"
    Screen.Item("E3Chart1").Pens.Item("Pena2").Color = RGB(212, 208, 20)
End Sub

```

Os comandos anteriores são equivalentes, o primeiro indicando o índice da pena na coleção, e o segundo indicando o nome.

2.6 Comando Set

O VBScript implementa o conceito de polimorfismo das linguagens orientadas a objeto, permitindo que uma variável do tipo Variant assumam a forma de um objeto qualquer, através do comando **Set**. Deste modo, a variável funciona como um ponteiro do objeto desejado, permitindo acessar seus métodos e propriedades. Exemplo:

```

Sub CommandButton1_Click()
    Set E3Chart = Screen.Item("E3Chart1")
    E3Chart.Pens.Item(2).Color = RGB(212, 208, 20)
End Sub

```

Neste exemplo foi feita a mesma tarefa que na seção anterior, porém a parte referente a como chegar até o objeto específico havia sido omitida. Sem o comando **Set**, a mesma chamada teria que ser:

```

Screen.Item("E3Chart1").Pens.Item(2).Color = RGB(212, 208, 20)

```

Aparentemente, não existe vantagem neste caso, pois pode-se fazer tudo em uma única linha de código. Porém, se logo abaixo no mesmo script, outras operações forem necessárias, o processo se torna mais simples e rápido se a chamada do método **Item** não tiver sido colocada em todas as linhas.

```

Sub CommandButton1_Click()
    ' Exemplo ruim
    Screen.Item("E3Chart1").Pens.Item(0).Color = RGB(212, 208, 20)
    Screen.Item("E3Chart1").Pens.Item(1).Color = RGB(200, 208, 20)
    Screen.Item("E3Chart1").Pens.Item(2).Color = RGB(100, 208, 20)
End Sub

Sub CommandButton2_Click()
    ' Exemplo melhor
    Set Penas = Screen.Item("E3Chart1").Pens
    Penas.Item(0).Color = RGB(212, 208, 20)
    Penas.Item(1).Color = RGB(200, 208, 20)
    Penas.Item(2).Color = RGB(100, 208, 20)
End Sub

```

2.7 Eventos, Métodos e Propriedades Gerais dos Objetos

Esta seção contém informações sobre eventos, métodos e propriedades gerais dos objetos.

2.7.1 Eventos

Os **Eventos** são ocorrências relacionadas a um objeto, que permitem disparar ações programadas. Existem basicamente, dois tipos de eventos: físicos (ou externos) e internos. Os eventos físicos são, por exemplo, ações do usuário. No caso do usuário digitar algo no teclado, a informação relevante pode ser a tecla pressionada, ou se o usuário apontar e clicar com o mouse, a informação relevante é a posição do cursor e o status dos botões. Os eventos internos são, por exemplo, mudanças de valor de uma variável (tag) no sistema. Como o tag pode ser associado a um dispositivo externo, diz-se que os eventos internos podem ter associações físicas, como a mudança da temperatura de uma câmara, por exemplo.

2.7.1.1 Variáveis de Evento

Variáveis de Evento são criadas quando o evento é iniciado. Para serem utilizadas, devem ser associadas a parâmetros na chamada do script do evento.

O exemplo a seguir é a chamada de um procedimento associado ao evento **KeyDown** de **AlgumObjeto**.

```
Sub AlgumObjeto_KeyDown(KeyCode, Shift)
```

Note que na chamada tem-se duas variáveis, *KeyCode* e *Shift*. O E3 irá atribuir valores a essas variáveis automaticamente no momento da ocorrência do evento. Nesse caso, *KeyCode* irá receber o código da tecla pressionada e *Shift* será verdadeiro ou falso, conforme a tecla SHIFT esteja pressionada ou não.

2.7.1.2 OnStartRunning

OnStartRunning()

Ocorre tão logo um objeto seja iniciado. Exemplo (**Meses** é um Tag Interno e utiliza o evento **OnStartRunning** para inicializar o vetor):

```
Sub Meses_OnStartRunning()  
    Value = Array ("Janeiro", "Fevereiro", "Março", "Abril",  
                  "Maio", "Junho", "Julho", "Agosto", "Setembro", "Outubro",_  
                  "Novembro", "Dezembro")  
End Sub
```

NOTA: Para acessar este array, é necessário copiar a propriedade **Value** para uma variável local.

2.7.1.3 OnStopRunning

OnStopRunning()

Ocorre quando termina a execução de uma instância deste objeto. Utilize o evento **OnStopRunning** para fazer operações de finalização para o objeto. Exemplo:

```
Sub TagInterno1_OnStopRunning()  
    ' Quando finaliza o objeto TagInterno1  
    ' atribui False ao TagInterno2  
    Set tag2 = Application.GetObject("Dados.TagInterno2")  
    tag2.Value = False  
End Sub
```

2.7.2 Métodos

Neste capítulo são listados os diversos métodos pré-definidos no E3, agrupando-os por tipos de objetos, iniciando pelos métodos padrão dos objetos da aplicação.

Cada entrada mostra o nome do método com seus respectivos parâmetros, na sintaxe correta, e um exemplo de utilização do método.

2.7.2.1 Chamadas de Métodos

Muitos métodos pré-definidos possuem parâmetros, que podem (ou devem) ser passados na chamada do método. Para isso, o VBScript possui uma regra que deve ser seguida: se o método for utilizado em uma atribuição, seus parâmetros devem estar entre parênteses. Por exemplo, veja esta chamada do método **GetObject**:

```
obj = Application.GetObject("dados.tag001")
```

Já se o método for chamado sozinho, deve-se retirar os parênteses. Por exemplo, veja esta chamada do método **SetVariableValue**:

```
Screen.Item("Consulta").SetVariableValue Valor, 12
```

Os parênteses usados nas citações de métodos neste manual servem apenas como indicativo para diferenciá-los das propriedades. Nos scripts, deve-se seguir esta mesma regra.

2.7.2.2 Activate

Activate()

Ativa um objeto que está inativo no momento. Exemplo:

```
Sub CommandButton1_Click()  
Dim obj, tag  
Set obj = Application.GetObject("Dados")  
' Cria o objeto novo e deixa desativado (False).  
Set tag = obj.AddObject("DemoTag", False)  
' Inicializa os parâmetros do objeto novo.  
tag.Name = "tag001"  
tag.Type = 3  
' Ativa o objeto (coloca em execução).  
tag.Activate()  
End Sub
```

2.7.2.3 AddObject

AddObject(*ClassName*, *Activate*)

O método **AddObject** adiciona um novo objeto à aplicação. Este método possui o parâmetro *ClassName*, que indica o tipo de objeto que será criado.

Por exemplo, para se criar um retângulo na Tela, o parâmetro *strClassName* deve ser "DrawRect". O objeto criado fica contido no objeto que chamou o método **AddObject** e pode ser acessado através do método **Item**.

O parâmetro *Activate* é opcional e indica se o objeto será ativado após a criação. Quando o objeto estiver ativado, os links e os scripts ficam habilitados. Se o objeto for criado com *Activate* em False, mais tarde ele pode ser ativado pelo método **Activate**.

O objeto só será criado se ele for de um tipo compatível com o objeto que o contém. Para se ter certeza que o objeto foi criado pode-se utilizar o método **IsObject**. Exemplo:

```
Sub CommandButton_Click()
    ' Acrescenta um retângulo (DrawRect) na tela.
    xc.Visible = True
    Set Retangulo = Screen.AddObject("DrawRect", True)
    Retangulo.X = 1000
    Retangulo.Y = 1000
    Retangulo.Width = 10000
    Retangulo.Height = 10000
End Sub
```

NOTA: Apenas os objetos que possuem a opção **Inserir** no menu podem acessar este método.

2.7.2.4 Deactivate

Deactivate()

Este método desativa um objeto criado ou previamente ativado pelo método **Activate**. Pode-se desativar um objeto quando for preciso fazer uma configuração prévia (iniciação de propriedades, por exemplo), ou quando se quiser fazer testes nos quais o objeto não pode estar presente e ativo. Exemplo:

```
Sub CommandButton1_Click()
    Dim obj, novo
    Set obj = Application.GetObject("Dados")
    Set novo = obj.AddObject("DemoTag", True)
    ' Desativa o objeto.
    novo.Deactivate()
End Sub
```

2.7.2.5 DeleteObject

DeleteObject(*ChildName*)

Apaga do projeto o objeto especificado. O parâmetro *ChildName* é um **String** (ignora maiúsculas e minúsculas) que indica o objeto filho que se deseja apagar. O método retorna True caso tenha conseguido apagar o objeto, ou False caso o objeto filho não exista.

Para se apagar um objeto a partir de uma referência a um elemento, utiliza-se o método **DeleteObject** do objeto pai. Exemplo:

```
Sub CommandButton1_Click()  
    Set obj = Application.GetObject("Dados")  
    If obj.DeleteObject("Tag001") Then  
        MsgBox("Tag apagado com sucesso!")  
    Else  
        MsgBox("Falha em apagar: o tag não existe.")  
    End If  
End Sub
```

NOTA: Apenas os objetos que possuem a opção **Inserir** no menu podem acessar este método.

2.7.2.6 GetChildObject

GetChildObject(*ObjectPath*)

O método **GetChildObject** retorna uma referência para o objeto filho apontado pelo parâmetro *ObjectPath*. Com isso é possível acessar todas as propriedades e métodos deste objeto, semelhante ao funcionamento do método **GetObject**. O método falhará se o caminho apontado por *ObjectPath* contiver uma propriedade ou método ao final. O caminho apontado pelo objeto filho não é um caminho desde a raiz (o arquivo .prj) e sim um caminho sempre a partir do objeto onde o método é chamado.

NOTA: Este método **NÃO** existe no objeto Application do servidor e nem nas Pastas de Aplicativo, porém existe no objeto Application do Viewer, sendo acessível mesmo num Viewer em modo Somente Leitura.

2.7.2.7 GetObject

GetObject(*ObjectPath*)

O método **GetObject** retorna a referência do objeto especificado em *ObjectPath*. Isto permite acessar todas as propriedades ou métodos do objeto. Esta é uma prática bastante comum na programação de scripts no E3. Ela facilita a manipulação de objetos e deixa o código mais inteligível. Exemplo:

```
Sub CommandButton1_Click()  
    ' Atribui o valor 20 à propriedade Value do objeto  
    ' TagInterno1 que está em Dados.  
    Set tag = Application.GetObject("Dados.TagInterno1")  
    tag.Value = 20  
End Sub
```

2.7.2.8 Item

Item(*ItemId*)

Retorna a referência para o objeto filho *ItemId* do objeto que chamou o método. O método **Item** pode buscar um objeto tanto pelo nome quanto pelo índice (inteiro, de 1 até o especificado na propriedade **Count**). Se o índice ou o nome especificado for válido, o método **Item** retorna a referência do objeto. Caso contrário, o método retorna um erro de "Parâmetro Inválido". Exemplo:

```
Sub Tela1_Click()  
    ' Atribui a obj a referência para o objeto filho Botao1  
    ' de Tela1.  
    Set obj = Item("Botao1")  
    ' Configura a propriedade BackColor de obj, ou seja,  
    ' de Botao1.  
    obj.BackColor = RGB(255, 0, 0)  
End Sub
```

2.7.2.9 Save

Save()

Este método salva o objeto especificado que foi modificado em tempo de execução (*runtime*). Os objetos-filho também serão salvos conforme as especificações do objeto-pai. Este método não é válido para os objetos Tela e Viewer. Exemplo:

```
Sub CommandButton1_Click()  
    Set area = Application.GetObject("ConfigAlarms")._  
    AddObject("Area", True)  
    Application.GetObject("ConfigAlarms").Save()  
End Sub
```

NOTA: As modificações feitas em *runtime* e salvas no objeto só serão visíveis no E3 Studio após a atualização do projeto, que pode ser feita clicando com o botão direito do mouse sobre o nome do projeto e selecionando a opção **Atualizar**.

2.7.3 Propriedades

Todo objeto possui **Propriedades**, que servem para guardar informações a respeito de suas características. Por exemplo, um objeto do tipo Retângulo possui a propriedade **Name**, que contém seu nome, e as propriedades **Width** e **Height**, que guardam a sua largura e a sua altura, respectivamente, entre outras propriedades.

Neste capítulo, listamos todas as propriedades dos objetos do E3. Cada entrada traz o nome da propriedade, descrição e, quando aplicável, um exemplo de seu uso. O primeiro tópico (Propriedades Padrão) introduz as propriedades presentes em todos os objetos do E3. Os tópicos seguintes trazem as propriedades específicas de cada objeto ou grupo de objetos mencionados em seu título. Em alguns destes tópicos, o primeiro item é também de propriedades comuns, neste caso, propriedades comuns a todos os objetos do grupo sendo descrito (por exemplo, no tópico **Propriedades dos Objetos de Tela**, o primeiro item é **Propriedades Comuns dos Objetos de Tela**, onde estão agrupadas todas as propriedades comuns a este grupo de objetos).

As propriedades são identificadas por um ícone que indica o tipo de dado suportado em seu conteúdo. Estes são:

Tipos de Dados disponíveis

ÍCONE	DADO	DESCRIÇÃO
	Booleano	Retorna Verdadeiro (True) ou Falso (False).
	Numérico	Retorna positivo, negativo, inteiro ou double, a ser definido pela propriedade.
	Data	Retorna uma data em formato Gregoriano (desde 1899).
	Texto	Retorna um texto.
	Variant	Retorna um tipo Variável, que pode assumir vários formatos.
	Cor	Retorna uma cor no formato RGB.
	Link	Retorna uma associação entre objetos.
	Enum	Retorna um conjunto determinado de valores.

Algumas propriedades podem propagar seus valores à mesma propriedade em seus objetos-filhos. Neste caso, são chamadas de propriedades propagáveis. Pode-se, no entanto, forçar a propriedade do objeto-filho a se comportar de modo diferente.

NOTA: O E3 utiliza para definição de coordenadas e espessura o sistema HIMETRIC. Neste sistema, cada unidade lógica equivale a um milésimo de centímetro; ou seja, cada 1000 unidades equivalem a 1 centímetro. Assim, este é o padrão adotado na descrição das propriedades do E3, quando aplicável.

2.7.3.1 Application

☞ A propriedade **Application** retorna o objeto de aplicação relacionado a este objeto. Com o objeto aplicação é possível, por exemplo, buscar outros objetos presentes no aplicativo. Exemplo:

```
Sub Tela1_Click()  
    ' Quando clica na tela, faz busca por um tag  
    Dim obj  
    Set obj = Application._  
        getObject("Dados.TagInterno1")  
    obj.Value = 100  
End Sub
```

2.7.3.2 Count

9 Retorna o número de objetos-filhos (itens) que o objeto possui. Esta propriedade funciona em conjunto com o método **Item**. Se o objeto não possuir filhos, o valor retornado será 0. Exemplo:

```
Sub Tela1_Click()  
    ' Num clique de tela, faz a busca por todos  
    ' os objetos da tela e ajusta a propriedade  
    ' ForegroundColor para vermelho  
    Dim obj  
    For i = 1 To Count  
        Set obj = Item(i) ' Pega objeto filho  
        obj.ForegroundColor = RGB(255, 0, 0)  
    Next  
End Sub
```

2.7.3.3 DocString

A Texto livre que tem por objetivo possibilitar a documentação das funcionalidades ou características do objeto pelos programadores do projeto. Exemplo:

```
Sub CommandButton1_Click()  
    Docstring = "Este botão ativa o condensador do sistema."  
    MsgBox Docstring  
End Sub
```

2.7.3.4 Links

 Retorna um objeto que é uma coleção das conexões (ou associações) de um objeto qualquer do E3. Esta propriedade é acessível apenas em tempo de execução.

2.7.3.5 Name

A Identifica cada um dos objetos presentes no sistema. Alterar esta propriedade implica em modificar todas as outras propriedades ou scripts que utilizam este objeto. Não é aconselhável a alteração desta propriedade em tempo de execução. Exemplo:

```
Sub CommandButton9_Click()  
    MsgBox "o nome da Tela é " & (Screen.Name)  
End Sub
```

2.7.3.6 Parent

 Retorna o objeto-pai deste objeto. Sendo assim, se um objeto está inserido na Tela, a propriedade **Parent** retorna o objeto Tela. Da mesma forma, se um Tag Interno está inserido diretamente abaixo de um Servidor de Dados, a propriedade **Parent** do Tag Interno apontará para o Servidor de Dados. Exemplo:

```
Sub Retangulo1_Click()  
    ' Ao clicar no Retangulo1, troca a cor do Retangulo2  
    ' Ambos objetos estão na mesma tela  
    Parent.Item("Retangulo2").ForegroundColor = RGB(255, 0, 0)  
End Sub
```

2.7.3.7 PathContainer

A Retorna um **String** contendo o caminho do objeto que contém o objeto atual, incluindo as pastas. Este valor é determinado apenas no momento em que for requisitado, por isso não é recomendado criar associações para esta propriedade.

2.7.3.8 PathName

A Identifica o caminho do objeto no sistema. Esta propriedade é acessível apenas em tempo de execução. Exemplo:

```
Sub CommandButton9_Click()  
    MsgBox "o path da Tela é " & (Screen.PathName)  
End Sub
```

2.7.3.9 PathVolume

A Retorna um **String** contendo o nome do arquivo .prj ou .lib que contém o objeto. No E3 Studio, a propriedade retorna o caminho completo do projeto ou biblioteca (c:\folder\folder\volume.prj). Em tempo de execução, os objetos que rodam no Viewer retornam sempre um **String** vazio. Já os que rodam no servidor retornam o caminho relativo do projeto ou biblioteca, conforme ele esteja armazenado no Domínio (volume.prj). Este valor é determinado apenas no momento em que for requisitado, por isso não é recomendado criar associações para esta propriedade.

CAPÍTULO

3

Drivers

Esta seção contém informações sobre eventos, métodos e propriedades dos seguintes objetos: Driver de Comunicação, Tag de Comunicação, Bloco de Comunicação, Elemento de Bloco de Comunicação, Driver de Comunicação OPC, Tag de Comunicação OPC, Bloco de Comunicação OPC e Elemento de Bloco de Comunicação OPC.

3.1 Driver de Comunicação

Esta seção contém informações sobre eventos, métodos e propriedades do objeto Driver de Comunicação.

3.1.1 Eventos

Esta seção contém informações sobre os eventos do objeto Driver de Comunicação.

3.1.1.1 AfterStart

AfterStart()

Ocorre depois que o driver de comunicação iniciou a comunicação. É comum fazer-se um script para este evento utilizando o método **Write** para fazer configurações em equipamentos. Exemplo:

```
Sub Driver1_AfterStart()  
    ' Após iniciada a comunicação manda escrever valores  
    ' para o equipamento/dispositivo  
    write 0, 2, 55, 2, 33.4  
    write 0, 3, 55, 20, "Metal"  
End Sub
```

3.1.1.2 AfterStop

AfterStop()

Ocorre depois que o driver finalizou a comunicação. Utilize o evento **AfterStop** para fazer alguma ação que seja necessária após terminada a comunicação do driver.

3.1.1.3 BeforeStart

BeforeStart()

Ocorre quando o driver está prestes a iniciar a comunicação. Utilize o evento **BeforeStart** para fazer alguma ação que seja necessária antes de iniciar a comunicação, como por exemplo a configuração de parâmetros do driver. Exemplo:

```
Sub Driver1_BeforeStart()  
    ' Faz a inicialização dos parâmetros do driver antes  
    ' de iniciar a comunicação  
    P1 = 0  
    P2 = 20  
    P3 = 80  
    P4 = 0  
End Sub
```

3.1.1.4 BeforeStop

BeforeStop()

Ocorre quando o driver está prestes a finalizar a comunicação. Utilize o evento **BeforeStop** para fazer alguma ação que seja necessária antes que a comunicação seja terminada, como por exemplo, escrever ou ler valores do equipamento ou dispositivo, antes que a comunicação não esteja mais disponível.

3.1.1.5 OnCommError

OnCommError(*EvtType, Size, Element, N1, N2, N3, N4*)

Ocorre quando algum erro de escrita ou leitura é detectado pelo driver de comunicação. Utilize o evento **OnCommError** para saber quando ocorreu uma falha de escrita ou leitura no driver. As variáveis de evento recebem informações sobre o erro ocorrido. Com estes valores, é possível rastrear quais tags estão apresentando problemas de comunicação.

Variáveis do evento OnCommError

NOME	DESCRIÇÃO
EvtType	<p>Informa qual o tipo de operação que o driver estava efetuando quando ocorreu o erro, conforme as opções a seguir:</p> <ul style="list-style-type: none">• 0: Erro de leitura de um só elemento (Size = 1). Param1 é N1, Param2 é N2, Param3 é N3 e Param4 é N4.• 1: Erro de escrita de um só elemento (Size = 1). Param1 é N1,

NOME	DESCRIÇÃO
	<p>Param2 é N2, Param3 é N3 e Param4 é N4.</p> <ul style="list-style-type: none"> • 2: Erro de leitura de um bloco (bloco de comunicação). Size é determinado pelo número de elementos do bloco. Param1 é N1, Param2 é N2, Param3 é N3 e Param4 é N4. • 3: Erro de escrita de um bloco (bloco de comunicação). Size é determinado pelo número de elementos do bloco. Param1 é N1, Param2 é N2, Param3 é N3 e Param4 é N4.
Size	Quantidade de valores sendo escritos ou lidos.
Element	Índice do elemento que estava sendo lido ou escrito, dentro do bloco.
N1	Parâmetro 1 da operação de leitura/escrita que gerou o erro.
N2	Parâmetro 2 da operação de leitura/escrita que gerou o erro.
N3	Parâmetro 3 da operação de leitura/escrita que gerou o erro.
N4	Parâmetro 4 da operação de leitura/escrita que gerou o erro.

Exemplo:

```

Sub Driver1_OnCommError(Type, Size, Element, N1, N2, N3, N4)
  Application.GetObject("Dados.TagInterno1").Value = _
    Application.GetObject("Dados.TagInterno1").Value + 1
  Application.GetObject("Dados.EvtType").Value = EvtType
  Application.GetObject("Dados.Size").Value = Size
  Application.GetObject("Dados.Element").Value = Element
  Application.GetObject("Dados.N1").Value = N1
  Application.GetObject("Dados.N2").Value = N2
  Application.GetObject("Dados.N3").Value = N3
  Application.GetObject("Dados.N4").Value = N4
End Sub

```

3.1.1.6 OnCommErrorEx

OnCommErrorEx(*ErrorInfo*)

Ocorre logo após a execução do método **OnCommError**.

Informações do parâmetro *ErrorInfo*

NOME	DESCRIÇÃO
ErrorInfo.EvtType	Indica o tipo de operação que causou o erro: <ul style="list-style-type: none">• 0: leitura de tag.• 1: escrita em tag.• 2: leitura de bloco.• 3: escrita em bloco.
ErrorInfo.Size	Tamanho do bloco que ocasionou o erro (se for um tag, o Size é 1).
ErrorInfo.Element	Índice do elemento de bloco que causou o erro.
ErrorInfo.Nx	Parâmetros Nx ou Bx (x= 1, 2, 3, 4) da operação que causou o erro.
ErrorInfo.ParamDevice	Parâmetro ParamDevice (string) da operação que causou o erro.
ErrorInfo.ParamItem	Parâmetro ParamItem (string) da operação que causou o erro.

3.1.1.7 OnTagRead

OnTagRead(*Tag*)

Ocorre na leitura de um tag, sempre que um novo valor ou um erro for retornado pelo driver de comunicação. Ou seja, se o valor ou a qualidade do tag não mudarem, o evento não será disparado. Para que o evento funcione, a propriedade **EnableDriverEvent** deve obrigatoriamente estar habilitada. Além disso, a propriedade **PercentDeadband** também pode exercer influência sobre a ocorrência do evento, caso a propriedade **EnableDeadband** esteja habilitada. Exemplo:

```

Sub Tags_OnTagRead(Tag)
  Set Obj = Application.GetObject("Dados1.TagName")
  Obj.Value = Tag.Name
  Set Obj = Application.GetObject("Dados1.TagRead")
  Obj.Value = True
  Set Obj = Application.GetObject("Dados1.TagType")
  Obj.Value = TypeName(Tag)
End Sub

```

3.1.1.8 OnTagWrite

OnTagWrite(Tag, Succeeded, User)

Ocorre quando uma escrita for disparada em qualquer tag do driver.

Variáveis do evento OnTagWrite

NOME	DESCRIÇÃO
Tag	Uma referência para o objeto do Tag que está sofrendo a escrita. Por exemplo, pode-se acessar a propriedade do tag com a sintaxe Tag.DocString .
Succeeded	Um valor Booleano que indica o sucesso da escrita (True = sucesso; False = falha).
User	Texto que retorna o usuário que fez a escrita.

3.1.2 Métodos

Esta seção contém informações sobre os métodos do objeto Driver de Comunicação.

3.1.2.1 Write

Write(N1, N2, N3, N4, Value, [WriteSyncMode])

Faz a escrita síncrona de um dado no equipamento. Este método retorna um booleano que indica o sucesso ou não da operação.

Os parâmetros *N1* a *N4* correspondem aos parâmetros **N** do driver. O parâmetro *Value* define o valor a ser escrito no driver. O parâmetro *WriteSyncMode* permite utilizar o modo de escrita diferente do utilizado no driver. As opções disponíveis para este parâmetro são: **0** - usa o modo de escrita configurado no driver; **1** - faz uma escrita síncrona; **2** - faz uma escrita assíncrona (sem confirmação). Se o parâmetro for omitido, será também utilizado o modo de escrita configurado no driver. Para maiores detalhes sobre estes parâmetros, consulte a documentação do driver. Exemplo:

```

Sub Botao1_Click()
    Dim val
    ' Quando clica no botão faz uma escrita no driver
    Set driver = Application.GetObject("Driver1")
    driver.write 4, 5, 1, 0, 55.5
    ' Faz a escrita do valor 55.5
End Sub

```

3.1.2.2 WriteEx

WriteEx(*N1, N2, N3, N4, Value, Timestamp, Quality, [WriteStatus], [WriteSyncMode]*)

Faz a escrita de um dado no equipamento. Este método retorna um booleano que indica o sucesso ou não da operação.

Os parâmetros *N1* a *N4* correspondem aos parâmetros **N** do driver. O parâmetro *Value* define o valor a ser escrito no driver. Para maiores detalhes sobre estes parâmetros, consulte a documentação do driver.

Os parâmetros *Timestamp*, *Quality* e *WriteStatus* são opcionais. Se omitidos, o comportamento do método é o mesmo do método **Write**. *Timestamp* especifica a data/hora a ser escrita no tag (se suportada pelo equipamento). Se omitido, é assumida a data/hora do momento da operação de escrita. *Quality* indica a qualidade (de 0 a 255). Se omitido, é assumida a qualidade Boa (192). *WriteStatus* recebe um valor retornado pelo driver, que indica o status da escrita (se suportado pelo driver, de acordo com sua documentação própria).

O parâmetro *WriteSyncMode* permite utilizar o modo de escrita diferente do utilizado no driver. As opções disponíveis para este parâmetro são: **0** - usa o modo de escrita configurado no driver; **1** - faz uma escrita síncrona; **2** - faz uma escrita assíncrona (sem confirmação). Se o parâmetro for omitido, será também utilizado o modo de escrita configurado no driver. Exemplo:

```

Dim status
If Driver1.writeEx(0, 0, 0, 0, 100, , , status) Then
    MsgBox "Escrita feita com sucesso, status = " & status
Else
    MsgBox "Escrita falhou, status = " & status
End If

```

3.1.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto Driver de Comunicação.

3.1.3.1 DriverLocation

A Define qual será o driver utilizado pelo driver de comunicação para efetuar a comunicação com o equipamento ou dispositivo. Esta propriedade aceita um string com o caminho completo do driver. Após isso, a propriedade **DriverName** mudará para a descrição do driver. Esta propriedade não pode ser modificada após iniciada a comunicação. O valor padrão desta propriedade é vazio. Exemplo:

```
Sub Driver1_BeforeStart()  
    Configura o DriverLocation como "c:\drivers\Driver1.dll"  
    DriverLocation = "c:\drivers\" & Name & ".dll"  
End Sub
```

3.1.3.2 DriverName

A A propriedade **DriverName** contém o string que descreve o driver associado ao objeto Driver de Comunicação. Para isso, deve-se antes configurar a propriedade **DriverLocation**. Esta propriedade é somente para leitura. O valor padrão desta propriedade é vazio. Exemplo:

```
Sub Driver1_BeforeStart()  
    MsgBox DriverName  
End Sub
```

3.1.3.3 EnableReadGrouping

Permite otimizações de leitura (agrupamento automático de tags). Esta propriedade não pode ser alterada em tempo de execução. Seu valor padrão é False (não permite otimizações de leitura).

3.1.3.4 P1

9 Utilize a propriedade **P1** para configurar o driver. Consulte a documentação do driver para sua correta parametrização. Esta propriedade não pode ser modificada após iniciada a comunicação. Exemplo:

```
Sub Driver1_BeforeStart()  
    ' Driver1 é um objeto do tipo Driver de Comunicação  
    DriverLocation = "c:\driver\plc.dll"  
    P1 = 2  
    P2 = 1  
    P3 = 9600  
End Sub
```

3.1.3.5 P2

9 Utilize a propriedade **P2** para configurar o driver. Consulte a documentação do driver para sua correta parametrização. Esta propriedade não pode ser modificada após iniciada a comunicação. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Driver1_BeforeStart()  
    ' Driver1 é um objeto do tipo Driver de Comunicação  
    DriverLocation = "c:\driver\plc.d11"  
    P1 = 2  
    P2 = 1  
    P3 = 9600  
End Sub
```

3.1.3.6 P3

9 Utilize a propriedade **P3** para configurar o driver. Consulte a documentação do driver para sua correta parametrização. Esta propriedade não pode ser modificada após iniciada a comunicação. Exemplo:

```
Sub Driver1_BeforeStart()  
    ' Driver1 é um objeto do tipo Driver de Comunicação  
    DriverLocation = "c:\driver\plc.d11"  
    P1 = 2  
    P2 = 1  
    P3 = 9600  
End Sub
```

3.1.3.7 P4

9 Utilize a propriedade **P4** para configurar o driver. Consulte a documentação do driver para sua correta parametrização. Esta propriedade não pode ser modificada após iniciada a comunicação. Exemplo:

```
Sub Driver1_BeforeStart()  
    ' Driver1 é um objeto do tipo Driver de Comunicação  
    DriverLocation = "c:\driver\plc.d11"  
    P1 = 2  
    P2 = 1  
    P3 = 9600  
    P4 = 500  
End Sub
```

3.1.3.8 ParamDevice

A Define o endereço do equipamento acessado pelo driver. Esta propriedade é herdada pelos tags filhos do driver, que podem sobrescrever este valor, se necessário.

3.1.3.9 ReadRetries

9 Indica o número de re-tentativas de leitura do driver em caso de erro. Se for configurada para 2, por exemplo, indica que o driver vai tentar duas vezes mais uma comunicação falhada, sem contar a tentativa original.

3.1.3.10 ShareMaximum

9 A propriedade **ShareMaximum** define o número máximo de drivers de comunicação que serão agrupados em um I/O Server compartilhado. Esta propriedade só é utilizada se a propriedade **ShareServer** estiver habilitada. Exemplo:

```
' Este driver não será compartilhado
ShareServer = False
ShareMaximum = <qualquer valor>
' Todos os drivers serão agrupados no mesmo IOserver
' Não define um limite
ShareServer = True
ShareMaximum = 0
' Agrupa cada 5 drivers em um IOserver
ShareServer = True
ShareMaximum = 5
```

3.1.3.11 ShareServer

☑ Se a propriedade **ShareServer** estiver configurada para True, isto significa que este driver irá compartilhar sua execução entre os demais objetos Driver de Comunicação que possuam o mesmo string em **DriverLocation**. Isto implica que somente o primeiro driver de comunicação configurado executará a inicialização da comunicação. Todos os demais objetos Driver de Comunicação compartilhados ignorarão todos os parâmetros de configuração **P1** até **P4**, e também outras configurações. Do contrário, se a propriedade estiver configurada para False, o driver não compartilhará qualquer tipo de comunicação com outros objetos Driver de Comunicação. Esta propriedade não pode ser modificada após iniciada a comunicação. O valor padrão desta propriedade é False.

3.1.3.12 WriteFeedbackMode

📄 Esta propriedade permite controlar a confirmação (ou *feedback*) das escritas em tags. É aplicável somente a tags que possam ser lidos, ou seja, possuam a propriedade **AllowRead** em True. Através desta propriedade, é possível tornar mais imediata a leitura de tags que recebem escritas. Esta propriedade possui as seguintes opções de configuração:

Opções disponíveis para WriteFeedbackMode

OPÇÃO	DESCRIÇÃO
0 - wfWaitNextRead	A leitura do tag será feita normalmente no próximo <i>scan</i> .
1 - wfImmediateReadAfterWrite	Após cada escrita, uma leitura de confirmação será feita o mais cedo possível.
2 - wfTrustWriteSuccess	Se o driver indicar sucesso na escrita, o valor escrito é assumido diretamente pelo tag, sem lê-lo do PLC.

O valor padrão desta propriedade é **1 - wfImmediateReadAfterWrite**. Aplicações anteriores à existência desta propriedade assumem o valor **0 - wfWaitNextRead** quando carregadas. Exemplo:

```

Sub CommandButton1_Click()
Dim modo
modo = Application.GetObject("Driver1").WriteFeedbackMode
MsgBox modo
Select case modo
Case 0
    MsgBox "A leitura do tag será feita no próximo scan."
Case 1
    MsgBox "Após cada escrita, uma leitura de confirmação _
        será feita o mais cedo possível."
Case 2
    MsgBox "Se o driver indicar sucesso na escrita, _
        o valor escrito é assumido diretamente pelo tag, _
        sem lê-lo do PLC."
End Select
End Sub
    
```

NOTA: Quando for usado o modo **2 - wfTrustWriteSuccess**, o *timestamp* e a qualidade podem ficar errados, já que numa escrita bem sucedida, o valor é assumido pelo tag sem buscar o Timestamp e a Qualidade no PLC. Além disso, o próprio valor assumido pode ter um pequeno desvio devido a qualquer tipo de arredondamento que possa ocorrer no driver ou no PLC. Deve ser levado em conta ainda que alguns drivers ou protocolos podem indicar sucesso, mesmo quando a escrita falhou. Por tudo isso, os outros modos (**1 - wfImmediateReadAfterWrite** ou **0 - wfWaitNextRead**) devem ser preferidos sempre que possível.

3.1.3.13 WriteRetries

9 Indica o número de re-tentativas de escrita do driver em caso de erro. Se for igual a 2, indica que o driver vai tentar duas vezes mais uma comunicação falhada, sem contar a tentativa original.

3.1.3.14 WriteSyncMode

 Determina como as escritas serão enviadas para o IO Server (modo síncrono ou assíncrono). Esta propriedade possui as seguintes opções de configuração:

Opções disponíveis para WriteSyncMode

OPÇÃO	DESCRIÇÃO
0 - wsmDefault	Modo síncrono (padrão).
1 - wsmSync	Modo síncrono. Sempre que um valor é escrito em algum tag o E3Run envia a escrita para o IO Server e aguarda o retorno da escrita.
2 - wsmAsyncUnconfirmed	Modo assíncrono sem confirmação. Todas as escritas são enviadas para o IO Server sem aguardar o retorno, e assume-se sempre que a escrita funcionou. Quando em modo assíncrono, os métodos de escrita dos tags (Write , WriteEx) sempre retornam True imediatamente, e o status da escrita (nos métodos que retornam este status) fica sempre vazio (Empty). O evento OnTagWrite do driver é executado assim que a escrita é enviada para o IO Server, e o parâmetro <i>Succeeded</i> fica sempre em True.

As escritas assíncronas serão executadas pelo IO Server assim que o driver ficar disponível (quando a leitura atual tiver acabado). Se várias escritas assíncronas forem enviadas para o IO Server, o driver só retomará as leituras depois que todas as escritas assíncronas forem executadas.

3.2 Tag de Comunicação

Esta seção contém informações sobre eventos, métodos e propriedades do objeto Tag de Comunicação.

3.2.1 Eventos

Esta seção contém informações sobre os eventos do objeto Tag de Comunicação.

3.2.1.1 OnRead

OnRead()

Ocorre quando é efetuada uma leitura do tag pelo driver. Utilize o evento **OnRead** quando for necessário efetuar alguma operação logo após algum dado ser modificado no tag, como por exemplo, as propriedades **Value**, **Quality** ou **TimeStamp**. Este evento é gerado por uma leitura em *background*. Exemplo:

```
Sub CommTag1_OnRead()  
    ' Ao ler o tag, atribui o seu valor ao tag InternalTag1  
    Set obj = Application.GetObject("DataServer1.InternalTag1")  
    obj = Value ' valor de CommTag1  
End Sub
```

3.2.2 Métodos

Esta seção contém informações sobre os métodos do objeto Tag de Comunicação.

3.2.2.1 Write

Write([WriteSyncMode])

Faz a escrita do valor corrente do tag de comunicação no equipamento. Normalmente, este comando de script só é usado quando a propriedade **AllowWrite** deste objeto for False.

O parâmetro *WriteSyncMode* permite utilizar o modo de escrita diferente do utilizado no driver. As opções disponíveis para este parâmetro são: **0** - usa o modo de escrita configurado no driver; **1** - faz uma escrita síncrona; **2** - faz uma escrita assíncrona (sem confirmação). Se o parâmetro for omitido, será também utilizado o modo de escrita configurado no driver.

Para maiores detalhes, consulte a documentação do driver. Este método retorna um booleano que indica o sucesso ou não da operação.

3.2.2.2 WriteEx

WriteEx(Value, Timestamp, Quality, [WriteStatus], [WriteSyncMode])

Faz a escrita de um valor no equipamento. Todos os seus parâmetros são opcionais; se omitidos, o comportamento do método é igual ao do método **Write**. Este método retorna um booleano que indica o sucesso ou não da operação.

O parâmetro *Value* define o valor a ser escrito no driver. O tipo de dado depende do driver; se omitido, é assumido o valor corrente do tag. *Timestamp* especifica a data/hora a ser escrita no tag (se suportada pelo equipamento). Se omitido, é assumida a data/hora do momento da operação de escrita. *Quality* indica a qualidade (de 0 a 255). Se omitido, é assumida a qualidade Boa (192). *WriteStatus* recebe um valor retornado pelo driver, que indica o status da escrita (se suportado pelo driver, de acordo com sua documentação própria).

O parâmetro *WriteSyncMode* permite utilizar o modo de escrita diferente do utilizado no driver. As opções disponíveis para este parâmetro são: **0** - usa o modo de escrita configurado no driver; **1** - faz uma escrita síncrona; **2** - faz uma escrita assíncrona (sem confirmação). Se o parâmetro for omitido, será também utilizado o modo de escrita configurado no driver. Exemplo:

```
Sub Tag1_OnRead()  
    ' O método WriteEx pode ser usado para transferir  
    ' valores de variáveis de um driver para outro.  
    Application.GetObject("Driver2.Tag")._  
        WriteEx Value, TimeStamp, Quality  
End Sub
```

3.2.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto Tag de Comunicação.

3.2.3.1 AdviseType

 Controla o modo **Advise**. As opções disponíveis são:

Opções disponíveis para a propriedade AdviseType

OPÇÃO	DESCRIÇÃO
0 - AlwaysInAdvise	O tag é mantido atualizado se a propriedade AllowRead for True.
1 - AdviseWhenLinked	O tag só é atualizado se AllowRead for True e o tag estiver associado a algum objeto ativo, por exemplo, um Display

OPÇÃO	DESCRIÇÃO
	de uma Tela aberta, um Alarme habilitado, entre outros. A associação do tag para esse propósito pode ser feita nas seguintes propriedades: Value , RawValue , TimeStamp , Quality e Bit00 até Bit31 do Tag de Comunicação.

Exemplo:

```
Sub CommandButton3_Click()
    MsgBox Application._
    Application.GetObject("Driver1.Tag1").AdviseType
End Sub
```

3.2.3.2 AllowRead

Configure esta propriedade para definir se este tag deverá ou não ser lido pelo driver de comunicação. Se esta propriedade estiver configurada para True, o driver automaticamente atualizará as propriedades **Value** e **Bits** (de **Bit00** até **Bit31**) deste objeto em intervalos de tempo definidos pela propriedade **Scan**. Caso contrário, este tag de comunicação não será lido. Esta propriedade pode ser modificada em execução. O valor padrão desta propriedade é True. Exemplo:

```
Sub Botao1_Click()
    ' Faz com que seja parada a leitura do tag
    Set obj = Application.GetObject("Driver1.tag")
    obj.AllowRead = False
End Sub
```

3.2.3.3 AllowWrite

Configure esta propriedade para definir se este tag deverá ou não ser escrito automaticamente quando for modificada a propriedade **Value** ou qualquer propriedade **Bit** (de **Bit00** até **Bit31**). Se esta propriedade estiver configurada para True, as modificações serão enviadas para o equipamento ou dispositivo associado ao driver de comunicação. Caso contrário, as modificações serão ignoradas. O valor padrão desta propriedade é True. Exemplo:

```
Sub Botao1_Click()
    ' Faz com que seja desabilitada a escrita do tag
    Set obj = Application.GetObject("Driver1.tag")
    obj.Allowwrite = False
End Sub
```

3.2.3.4 Bit00 - Bit31

■ As propriedades de bit **Bit00** até **Bit31** representam juntas os 32 bits do valor presente na propriedade **Value** do objeto tag de comunicação, sendo **Bit00** o bit menos significativo e **Bit31** o bit mais significativo. Modificar cada um destes bits implica na igual modificação da propriedade **Value** do tag, e vice-versa, mas isto somente ocorre quando a propriedade **UseBitFields** estiver configurada para True. O valor padrão destas propriedades é False.

3.2.3.5 DeviceHigh

9 A propriedade **DeviceHigh** define qual o valor máximo atingido pelo tag no equipamento ou dispositivo. Esta propriedade será utilizada para fazer o ajuste de escala do valor proveniente do equipamento ou dispositivo antes de ser atribuído à propriedade **Value**. Da mesma forma, no momento da escrita é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 1000. Exemplo:

```
Sub Tag_OnStartRunning()
' Faz o ajuste de escala de um tag de temperatura
' que varia de 0 a 255 no PLC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub
```

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.2.3.6 DeviceLow

9 A propriedade **DeviceLow** define o valor mínimo atingido por este tag no equipamento ou no dispositivo. Esta propriedade será utilizada para fazer o ajuste de escala do valor proveniente do equipamento ou do dispositivo antes de ser atribuído à propriedade **Value**. Da mesma forma, no momento da escrita é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. Exemplo:

```

Sub Tag_OnStartRunning()
' Faz o ajuste de escala de um tag de temperatura
' que varia de 0 a 255 no PLC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub

```

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala, isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.2.3.7 EnableDeadBand

■ A propriedade **EnableDeadBand** habilita ou desabilita a propriedade **PercentDeadBand**. Se esta propriedade estiver configurada para True, o valor do tag é atualizado somente se este mudar e o novo valor exceder o limite definido pela propriedade **PercentDeadBand**. Caso contrário, o tag é sempre atualizado e o limite de *deadband* (banda morta) não é verificado. Sempre mantenha a banda morta (*deadband*) habilitada, pois ela melhora a performance da aquisição de dados e processamento. Você deve desabilitar o *deadband* somente em casos onde o Driver de Comunicação estiver retornando eventos de tags digitais ou analógicos, e seja necessário processar estes eventos via script no evento **OnRead** do tag. O valor padrão desta propriedade é True.

3.2.3.8 EnableDriverEvent

■ Esta propriedade é utilizada para controlar a geração do evento **OnTagRead**, que ocorre no driver de comunicação que contém o bloco. Se a propriedade **EnableDriverEvent** do tag estiver configurada para True, será habilitada a geração do evento **OnTagRead** por este tag. Caso contrário, não ocorre. Os três tipos de elementos de comunicação (Tag de Comunicação, Bloco de Comunicação e Elemento de Bloco) podem gerar este evento. O evento ocorre no driver e não no bloco.

3.2.3.9 EnableScaling

■ A propriedade **EnableScaling** habilita ou desabilita a escala de valor para o valor enviado e recebido do equipamento ou do dispositivo. Se esta propriedade estiver configurada para True, todo valor proveniente do equipamento ou do dispositivo sofrerá ajuste de escala nas propriedades **DeviceHigh**, **DeviceLow**, **EUHigh** e **EULow**, antes de ser atribuído à propriedade **Value**. Caso contrário, nenhum ajuste de escala será efetuado em nenhum dos dois sentidos (escrita e leitura). Exemplo:

```

Sub Tag_OnStartRunning()
' Faz o ajuste de escala de um tag de temperatura
' que varia de 0 a 255 no PLC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub

```

3.2.3.10 EU

A Esta propriedade é utilizada para identificar a unidade de engenharia que o valor representa, como por exemplo, graus, metros, KW/h, etc. Exemplo:

```

Sub Tag_OnStartRunning()
' Faz o ajuste de escala de um tag de temperatura
' que varia de 0 a 255 no PLC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub

```

3.2.3.11 EUHigh

9 Define qual o valor máximo que deverá ser atribuído à propriedade **Value**, ajustando a escala ao valor do equipamento ou do dispositivo antes desta atribuição. Da mesma forma, no momento da escrita é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 1000. Exemplo:

```

Sub Tag_OnStartRunning()
' Faz o ajuste de escala de um tag de temperatura
' que varia de 0 a 255 no PLC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub

```

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.2.3.12 EULow

9 Define qual o valor mínimo que deverá ser atribuído à propriedade **Value**, ajustando a escala ao valor do equipamento ou do dispositivo antes desta atribuição. Da mesma forma, no momento da escrita, é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Tag_OnStartRunning()  
    ' Faz o ajuste de escala de um tag de temperatura  
    ' que varia de 0 a 255 no PLC, mas significam na  
    ' prática de 0 a 100 graus Celsius  
    EU = "Graus Celsius"  
    EnableScaling = True  
    DeviceHigh = 255  
    DeviceLow = 0  
    EULow = 100  
End Sub
```

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.2.3.13 N1

9 Especifica a variável do equipamento ou do dispositivo a que este tag estará associado. Consulte a documentação do driver para sua correta parametrização. Esta propriedade pode ser modificada após iniciada a comunicação. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Tag_OnStartRunning()  
    N1 = 10  
End Sub
```

3.2.3.14 N2

9 Especifica a variável do equipamento ou do dispositivo a que este tag estará associado. Consulte a documentação do driver para sua correta parametrização. Esta propriedade pode ser modificada após iniciada a comunicação. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Tag_OnStartRunning()  
    N2 = 3  
End Sub
```

3.2.3.15 N3

9 Especifica a variável do equipamento ou do dispositivo a que este tag estará associado. Consulte a documentação do driver para sua correta parametrização. Esta propriedade pode ser modificada após iniciada a comunicação. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Tag_OnStartRunning()
    N1 = 10
    N3 = 5
    N4 = 20
End Sub
```

3.2.3.16 N4

9 Especifica a variável do equipamento ou do dispositivo a que este tag estará associado. Consulte a documentação do driver para sua correta parametrização. Esta propriedade pode ser modificada após iniciada a comunicação. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Tag_OnStartRunning()
    N1 = 10
    N4 = 20
End Sub
```

3.2.3.17 ParamDevice

A Define o endereço do equipamento acessado pelo tag. Esta propriedade é herdada do driver, mas seu valor pode ser sobrescrito, se necessário.

3.2.3.18 ParamItem

A Identifica o dado que o tag acessa dentro de um equipamento.

3.2.3.19 PercentDeadBand

9 A propriedade **PercentDeadBand** determina a variação mínima do valor (banda morta) de um tag, para que seu valor no E3 seja atualizado. Este valor é especificado como um percentual da diferença entre as propriedades **DeviceHigh** e **DeviceLow**. Esta propriedade só será utilizada caso a propriedade **EnableDeadBand** do mesmo tag esteja configurada para True. Se a propriedade **PercentDeadband** for igual a 0, o tag não terá banda morta, e qualquer variação no seu valor será repassada para o E3. Caso contrário, só será enviado para o E3 um novo valor cuja diferença, em relação ao valor atual no E3, seja maior que a banda morta. O valor padrão desta propriedade é 0.

3.2.3.20 Quality

 A propriedade **Quality** informa qual a qualidade do valor contido na propriedade **Value**. Cada vez que o Driver atribui um novo valor para o Tag, ele também configura a qualidade daquele dado. Esta propriedade é somente para leitura. O valor padrão desta propriedade é 0 (Qualidade Ruim).

NOTA: Para maiores informações sobre qualidade, consulte o tópico **Propriedade Quality** ou o tópico **Qualidade no Manual do Usuário**.

3.2.3.21 RawValue

 Acessa o valor original do tag anterior à escala, ou seja, independentemente da configuração da propriedade **EnableScaling**. Então, se esta propriedade for False, as propriedades **Value** e **RawValue** apresentam o mesmo comportamento.

3.2.3.22 Scan

 Utilize a propriedade **Scan** para especificar o tempo de varredura que será utilizado pelo servidor para atualizar a propriedade **Value**. Esta propriedade é representada em milissegundos e pode ser modificada após iniciada a comunicação, sendo utilizada apenas quando a propriedade **AllowRead** estiver configurada para True. Ao configurar esta propriedade nos diversos tags presentes na aplicação, recomenda-se aumentar o valor da propriedade **Scan** para aqueles tags que não variam muito no equipamento ou no dispositivo, possibilitando assim que outros tags de maior prioridade sejam lidos com maior frequência, aumentando assim a performance geral do sistema. O valor padrão desta propriedade é 1000 (1 segundo). O valor do *scan* deve ser maior que 0. Exemplo:

```
Sub Tag_OnStartRunning()
  Scan = 1500
End Sub
```

3.2.3.23 TimeStamp

 A propriedade **TimeStamp** é atualizada sempre que ocorrer alguma mudança de valor ou estado nas propriedades **Value** ou **Quality**. Ele informa qual a data/hora associada tanto ao valor quanto à qualidade do tag de comunicação. Esta é uma propriedade somente para leitura. O valor padrão desta propriedade é 00:00:00.

3.2.3.24 UseBitFields

☑ Caso a propriedade **UseBitFields** esteja configurada para True, toda vez que o valor da propriedade **Value** for modificado, ele atualizará os bits referentes às propriedades **Bit00** até **Bit31**. Da mesma forma, atualizará o valor de **Value** toda vez que alguma das propriedades **Bit00** até **Bit31** mudar de valor e posteriormente enviará para o equipamento ou para o dispositivo, se a propriedade **AllowWrite** estiver configurada para True. Caso contrário, os bits não sofrerão nem acarretarão qualquer mudança. Esta propriedade pode ser atualizada após ter sido iniciada a comunicação.

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala, isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.2.3.25 Value

▼ Atualizada sempre que uma nova leitura válida de um valor do equipamento ou do dispositivo for feita, utilizando-se a parametrização **N1** até **N4**. O tipo desta variável (número inteiro, ponto flutuante, texto) depende do driver ao qual o tag está associado e da sua parametrização.

Esta propriedade somente será atualizada desta forma se a propriedade **AllowRead** estiver configurada para True, e quando não houver erros de comunicação (neste caso somente as propriedades **Quality** e **TimeStamp** serão atualizadas), de acordo com o tempo de varredura definido em **Scan**. Outra forma de utilizar esta propriedade é para escrever valores no equipamento ou no dispositivo, bastando para isto atribuir um novo valor à propriedade **Value** ou a algum dos bits **Bit00** até **Bit31**. Neste caso, a propriedade **AllowWrite** deve estar configurada para True.

Esta também é a propriedade padrão do objeto Tag de Comunicação. Sendo assim, uma referência por valor a um Tag de Comunicação não necessita obrigatoriamente explicitar a propriedade **Value** para acessar o valor. O valor padrão desta propriedade é vazio (sem valor). Exemplo:

```
Sub Botao1_Click()
    ' Acessa um tag e mostra o valor atual
    ' tag1 é um objeto tipo Tag de Comunicação
    Set obj = Application.GetObject("Driver de Comunicação1...
    tag1")
    MsgBox "Valor atual de tag1: " & obj.Value
    ' Isto também pode ser feito de outra forma,
    ' sem mostrar a propriedade Value, que é o padrão
    MsgBox "Valor atual de tag1: " & obj
End Sub
```

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.3 Bloco de Comunicação

Esta seção contém informações sobre eventos, métodos e propriedades do objeto Bloco de Comunicação.

3.3.1 Eventos

Esta seção contém informações sobre os eventos do objeto Bloco de Comunicação.

3.3.1.1 OnRead

OnRead()

Ocorre quando uma leitura do bloco de comunicação é efetuada pelo driver. Utilize o evento **OnRead** quando for necessário efetuar alguma operação logo após algum dado ter sido modificado no objeto bloco de comunicação, como por exemplo as propriedades **Quality**, **TimeStamp** ou ainda a propriedade **Value** de algum elemento do bloco. Exemplo:

```
Sub IOBloco1_OnRead()  
    ' Ao ler o bloco, atribui ao tag InternalTag1  
    ' o valor do elemento de bloco elm1  
    Set obj = Application.GetObject("DataServer1.InternalTag1")  
    Set elm = Application.GetObject("Driver1.IOBloco1.elm1")  
    obj.Value = elm.Value  
End Sub
```

3.3.2 Métodos

Esta seção contém informações sobre os métodos do objeto Bloco de Comunicação.

3.3.2.1 Write

Write([WriteSyncMode])

Faz a escrita do valor corrente do bloco de comunicação no equipamento. Normalmente, este comando de script só é usado quando a propriedade **AllowWrite** deste objeto for False.

O parâmetro *WriteSyncMode* permite utilizar o modo de escrita diferente do utilizado no driver. As opções disponíveis para este parâmetro são: **0** - usa o modo de escrita configurado no driver; **1** - faz uma escrita síncrona; **2** - faz uma escrita assíncrona (sem

confirmação). Se o parâmetro for omitido, será também utilizado o modo de escrita configurado no driver.

Para maiores detalhes, consulte a documentação do driver. Este método retorna um booleano que indica o sucesso ou não da operação.

3.3.2 WriteEx

WriteEx(Value, Timestamp, Quality, [WriteStatus], [WriteSyncMode])

Faz a escrita de um valor no equipamento. Todos os seus parâmetros são opcionais; se omitidos, o comportamento do método é igual ao do método **Write**. Este método retorna um booleano que indica o sucesso ou não da operação.

O parâmetro *Value* define o valor a ser escrito no driver. O tipo de dado depende do driver; se omitido, é assumido o valor corrente do tag. *Timestamp* especifica a data/hora a ser escrita no tag (se suportada pelo equipamento). Se omitido, é assumida a data/hora do momento da operação de escrita. *Quality* indica a qualidade (de 0 a 255). Se omitido, é assumida a qualidade Boa (192). *WriteStatus* recebe um valor retornado pelo driver, que indica o status da escrita (se suportado pelo driver, de acordo com sua documentação própria).

O parâmetro *WriteSyncMode* permite utilizar o modo de escrita diferente do utilizado no driver. As opções disponíveis para este parâmetro são: **0** - usa o modo de escrita configurado no driver; **1** - faz uma escrita síncrona; **2** - faz uma escrita assíncrona (sem confirmação). Se o parâmetro for omitido, será também utilizado o modo de escrita configurado no driver. Exemplo:

```
Sub Tag1_OnRead()  
    ' O método WriteEx pode ser usado para transferir  
    ' valores de variáveis de um driver para outro.  
    Application.GetObject("Driver2.Tag")._  
        WriteEx Value, TimeStamp, Quality  
End Sub
```

3.3.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto Bloco de Comunicação.

3.3.3.1 AdviseType

 Controla o modo Advise. As opções disponíveis são:

Opções disponíveis para AdviseType

OPÇÃO	DESCRIÇÃO
0 - AlwaysInAdvise	O tag é mantido atualizado se a propriedade AllowRead for True.
1 - AdviseWhenLinked	O tag só é atualizado se AllowRead for True e o tag estiver associado a algum objeto ativo, por exemplo, um Display de uma Tela aberta, um Alarme habilitado, entre outros. A associação do tag para esse propósito pode ser feita nas seguintes propriedades: Value , RawValue , Quality e de Bit00 a Bit31 de Elementos de Bloco, e Quality e TimeStamp de Blocos de Comunicação.

3.3.3.2 AllowRead

 Define se este bloco deverá ou não ser lido pelo driver de comunicação. Caso a propriedade **AllowRead** esteja habilitada, então o driver automaticamente atualizará os elementos de comunicação inseridos neste bloco, em intervalos de tempo definidos pela propriedade **Scan**. Caso contrário, este bloco de comunicação não será lido ou atualizado. Esta propriedade pode ser modificada em tempo de execução. O valor padrão desta propriedade é True. Exemplo:

```
Sub Botao1_Click()  
    Faz com que seja parada a leitura do bloco  
    Set obj = Application.GetObject("Driver1.bloco1")  
    obj.AllowRead = False  
End Sub
```

3.3.3.3 AllowWrite

Define se este bloco deverá ou não ser escrito quando for modificada a propriedade **Value** de seus Elementos do Bloco de Comunicação. Se esta propriedade estiver configurada para True, as modificações serão enviadas para o equipamento ou dispositivo associado ao Driver de Comunicação. Caso contrário as modificações serão ignoradas. Os elementos de comunicação não aceitarão valores, caso esta propriedade esteja configurada para False, a não ser que a propriedade **AllowRead** também esteja configurada para False. Exemplo:

```
Sub Botao1_Click()  
    Faz com que seja desabilitada a escrita do bloco  
    Set obj = Application.GetObject("Driver1.bloco1")  
    obj.Allowwrite = False  
End Sub
```

3.3.3.4 B1

9 Especifica a qual conjunto de dados no equipamento ou no dispositivo este tag estará associado. Consulte a documentação do driver para sua correta parametrização. Esta propriedade pode ser modificada após iniciada a comunicação. Exemplo:

```
Sub Bloco1_BeforeStart()  
  B1 = 2  
  B2 = 1  
  B3 = 9600  
End Sub
```

3.3.3.5 B2

9 Especifica a qual conjunto de dados no equipamento ou no dispositivo este tag estará associado. Consulte a documentação do driver para sua correta parametrização. Esta propriedade pode ser modificada após iniciada a comunicação. Exemplo:

```
Sub Bloco1_BeforeStart()  
  B1 = 2  
  B2 = 1  
  B3 = 9600  
End Sub
```

3.3.3.6 B3

9 Especifica a qual conjunto de dados no equipamento ou no dispositivo este tag estará associado. Consulte a documentação do driver para sua correta parametrização. Esta propriedade pode ser modificada após iniciada a comunicação. Exemplo:

```
Sub Bloco1_BeforeStart()  
  B1 = 2  
  B2 = 1  
  B3 = 9600  
End Sub
```

3.3.3.7 B4

9 Especifica a qual conjunto de dados no equipamento ou no dispositivo este tag estará associado. Consulte a documentação do driver para sua correta parametrização. Esta propriedade pode ser modificada após iniciada a comunicação. Exemplo:

```
Sub Bloco1_BeforeStart()  
  B1 = 2  
  B2 = 1  
  B3 = 9600  
  B4 = 524  
End Sub
```

3.3.3.8 EnableDeadBand

☑ A propriedade **EnableDeadBand** habilita ou desabilita a propriedade **PercentDeadBand**. Se esta opção estiver configurada para True, o valor do bloco é atualizado somente se este mudar e o novo valor exceder o limite definido pela propriedade **PercentDeadBand**. Caso contrário, o bloco é sempre atualizado e o limite de *deadband* (banda morta) não é verificado. Sempre que possível, é recomendado manter a *deadband* (banda morta) habilitada, pois ela melhora a performance da aquisição e do processamento de dados. Normalmente, a banda morta será desabilitada apenas para os tags que retornem valores que representem eventos que precisam ser tratados no evento **OnRead** do tag. O valor padrão desta propriedade é True.

3.3.3.9 EnableDriverEvent

☑ Esta propriedade é utilizada para controlar a geração do evento **OnTagRead**, que ocorre no driver de comunicação que contém o bloco. Se a propriedade **EnableDriverEvent** do tag estiver configurada para True, será habilitada a geração do evento **OnTagRead** por este tag. Caso contrário, não ocorre. Os três tipos de elementos de comunicação (Tag de Comunicação, Bloco de Comunicação e Elemento de Bloco) podem gerar este evento. O evento ocorre no driver e não no bloco.

3.3.3.10 ParamDevice

A Define o endereço do equipamento acessado pelo bloco. Esta propriedade é herdada do driver, mas seu valor pode ser sobrescrito, se necessário.

3.3.3.11 ParamItem

A Identifica o dado que o bloco acessa dentro de um equipamento.

3.3.3.12 Quality

9 A propriedade **Quality** informa qual a qualidade do valor contido na propriedade **Value**. Cada vez que o Driver atribui um novo valor para o Bloco, ele também configura a qualidade daquele dado. Esta propriedade é somente para leitura. O valor padrão desta propriedade é 0 (Qualidade Ruim).

NOTA: Para maiores informações sobre qualidade, consulte o tópico **Propriedade Quality** ou o tópico **Qualidade** no **Manual do Usuário**.

3.3.3.13 Scan

9 Especifica qual tempo de varredura será utilizado pelo servidor para atualizar o bloco. Esta propriedade é representada em milissegundos e pode ser modificada após iniciada a comunicação, sendo utilizada apenas quando a propriedade **AllowRead** estiver habilitada. Ao configurar esta propriedade nos diversos blocos presentes na aplicação, recomenda-se aumentar o valor da propriedade **Scan** para aqueles blocos que não variam muito no equipamento ou no dispositivo, possibilitando assim que outros blocos de maior prioridade sejam lidos com maior frequência, aumentando assim a performance geral do sistema. O valor padrão desta propriedade é 1000 (1 segundo). O valor do *scan* deve ser maior que zero. Exemplo:

```
Sub Bloco de comunicacao1_BeforeStart()  
    Scan = 152  
End Sub
```

3.3.3.14 Size

9 Define o tamanho do conjunto de valores deste bloco. Consulte a documentação do driver utilizado para conhecer qual o limite para esta propriedade, de acordo com os parâmetros *B1* até *B4*. Criando os elementos de bloco, possibilita-se o acesso aos valores lidos e também permite a escrita de valores para o equipamento ou dispositivo. Esta propriedade não pode ser modificada após a comunicação ser iniciada. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Bloco de comunicacao1_BeforeStart()  
    Size = 10  
End Sub
```

3.3.3.15 TimeStamp

 A propriedade **TimeStamp** é atualizada sempre que ocorrer alguma mudança de valor ou de estado nas propriedades **Value** ou **Quality**. Ele informa qual a data/hora associada tanto ao valor quanto à qualidade do Bloco de Comunicação. Esta é uma propriedade somente para leitura. O valor padrão desta propriedade é 00:00:00.

3.4 Elemento de Bloco de Comunicação

Esta seção contém informações sobre métodos e propriedades do objeto Elemento de Bloco de Comunicação. Este objeto não possui eventos associados.

3.4.1 Métodos

Esta seção contém informações sobre os métodos do objeto Elemento de Bloco de Comunicação.

3.4.1.1 Write

Write([WriteSyncMode])

Faz a escrita do valor corrente do elemento de bloco no equipamento. Normalmente, este comando de script só é usado quando a propriedade **AllowWrite** deste objeto for **False**.

O parâmetro *WriteSyncMode* permite utilizar o modo de escrita diferente do utilizado no driver. As opções disponíveis para este parâmetro são: **0** - usa o modo de escrita configurado no driver; **1** - faz uma escrita síncrona; **2** - faz uma escrita assíncrona (sem confirmação). Se o parâmetro for omitido, será também utilizado o modo de escrita configurado no driver.

Para maiores detalhes, consulte a documentação do driver. Este método retorna um booleano que indica o sucesso ou não da operação.

3.4.1.2 WriteEx

WriteEx(Value, Timestamp, Quality, [WriteStatus], [WriteSyncMode])

Faz a escrita de um valor no equipamento. Todos os seus parâmetros são opcionais; se omitidos, o comportamento do método é igual ao do método **Write**. Este método retorna um booleano que indica o sucesso ou não da operação.

O parâmetro *Value* define o valor a ser escrito no driver. O tipo de dado depende do driver; se omitido, é assumido o valor corrente do tag. *Timestamp* especifica a data/hora a ser escrita no tag (se suportada pelo equipamento). Se omitido, é assumida a data/hora do momento da operação de escrita. *Quality* indica a qualidade (de 0 a 255). Se omitido, é assumida a qualidade Boa (192). *WriteStatus* recebe um valor retornado pelo driver, que indica o status da escrita (se suportado pelo driver, de acordo com sua documentação própria).

O parâmetro *WriteSyncMode* permite utilizar o modo de escrita diferente do utilizado no driver. As opções disponíveis para este parâmetro são: **0** - usa o modo de escrita configurado no driver; **1** - faz uma escrita síncrona; **2** - faz uma escrita assíncrona (sem confirmação). Se o parâmetro for omitido, será também utilizado o modo de escrita configurado no driver. Exemplo:

```

Sub Tag1_OnRead()
    ' O método WriteEx pode ser usado para transferir
    ' valores de variáveis de um driver para outro.
    Application.GetObject("Driver2.Tag")._
        WriteEx Value, TimeStamp, Quality
End Sub

```

3.4.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Elemento de Bloco de Comunicação.

3.4.2.1 Bit00 - Bit31

As propriedades de bit **Bit00** até **Bit31** representam juntas os 32 bits do valor presente na propriedade **Value** do elemento de bloco, sendo **Bit00** o bit menos significativo e **Bit31** o bit mais significativo. Modificar cada um destes bits implica na igual modificação da propriedade **Value** do elemento, e vice-versa, mas isto somente ocorre quando a propriedade **UseBitFields** estiver configurada para True. O valor padrão destas propriedades é False.

3.4.2.2 DeviceHigh

A propriedade **DeviceHigh** define o valor máximo atingido pelo elemento de bloco no equipamento ou no dispositivo. Esta propriedade será utilizada para fazer o ajuste de escala do valor proveniente do equipamento ou do dispositivo antes de ser atribuído à propriedade **Value**. Da mesma forma, no momento da escrita é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 1000. Exemplo:

```

Sub Elemento_OnStartRunning()
    ' Faz o ajuste de escala de um elemento temperatura
    ' que varia de 0 a 255 no PLC, mas significam na
    ' prática de 0 a 100 graus Celsius
    EU = "Graus Celsius"
    EnableScaling = True
    DeviceHigh = 255
    DeviceLow = 0
    EUHigh = 100
    EULow = 0
End Sub

```

3.4.2.3 DeviceLow

A propriedade **DeviceLow** define qual o valor mínimo atingido por este elemento no equipamento ou no dispositivo. Esta propriedade será utilizada para fazer o ajuste de escala do valor proveniente do equipamento ou do dispositivo antes de ser atribuído à propriedade **Value**. Da mesma forma, no momento da escrita, é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta

propriedade é 0. Exemplo:

```
Sub Elemento_OnStartRunning()  
    ' Faz o ajuste de escala de um elemento temperatura  
    ' que varia de 0 a 255 no PLC, mas significam na  
    ' prática de 0 a 100 graus Celsius  
    EU = "Graus Celsius"  
    EnableScaling = True  
    DeviceHigh = 255  
    DeviceLow = 0  
    EUHigh = 100  
    EULow = 0  
End Sub
```

3.4.2.4 EnableDriverEvent

■ Esta propriedade é utilizada para controlar a geração do evento **OnTagRead**, que ocorre no driver de comunicação que contém o bloco. Se a propriedade **EnableDriverEvent** do tag estiver configurada para True, será habilitada a geração do evento **OnTagRead** por este tag. Caso contrário, não ocorre. Os três tipos de elementos de comunicação (Tag de Comunicação, Bloco de Comunicação e Elemento de Bloco) podem gerar este evento. O evento ocorre no driver e não no bloco.

3.4.2.5 EnableScaling

■ A propriedade **EnableScaling** habilita ou desabilita a escala de valor para o valor enviado e recebido do equipamento ou do dispositivo. Se esta propriedade estiver configurada para True, todo valor proveniente do equipamento ou do dispositivo sofrerá ajuste de escala conforme as propriedades **DeviceHigh**, **DeviceLow**, **EUHigh** e **EULow** antes de ser atribuído à propriedade **Value**. Caso contrário, nenhum ajuste de escala será efetuado em nenhum dos dois sentidos (escrita e leitura). O valor padrão desta propriedade é False. Exemplo:

```
Sub Elemento_OnStartRunning()  
    ' Faz o ajuste de escala de um elemento temperatura  
    ' que varia de 0 a 255 no PLC, mas significam na  
    ' prática de 0 a 100 graus Celsius  
    EU = "Graus Celsius"  
    EnableScaling = True  
    DeviceHigh = 255  
    DeviceLow = 0  
    EUHigh = 100  
    EULow = 0  
End Sub
```

3.4.2.6 EU

A Esta propriedade é utilizada para identificar qual unidade de engenharia seu valor representa, como por exemplo: graus, metros, KW/h, etc. Exemplo:

```

Sub ElementodeBloco1_OnStartRunning()
' Faz o ajuste de escala de um tag de temperatura
' que varia de 0 a 255 no PLC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub

```

3.4.2.7 EUHigh

9 Define qual o valor máximo que deverá ser atribuído à propriedade **Value**, ajustando a escala ao valor do equipamento ou do dispositivo antes desta atribuição. Da mesma forma, no momento da escrita é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 1000. Exemplo:

```

Sub Elemento_OnStartRunning()
' Faz o ajuste de escala de um elemento temperatura
' que varia de 0 a 255 no PLC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub

```

3.4.2.8 EULow

9 Define qual o valor mínimo que deverá ser atribuído à propriedade **Value**, ajustando a escala ao valor do equipamento ou do dispositivo antes desta atribuição. Da mesma forma, no momento da escrita, é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 0. Exemplo:

```

Sub Elemento_OnStartRunning()
' Faz o ajuste de escala de um elemento temperatura
' que varia de 0 a 255 no PLC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EULow = 100
EULow = 0
End Sub

```

3.4.2.9 Index

9 Especifica a posição que o elemento de bloco ocupa entre os elementos configurados pela propriedade **Size** do bloco de comunicação no qual ele está inserido. Esta propriedade aceita valores de 0 até um número menor do que a valor definido em **Size**. Por exemplo: configurando um bloco de comunicação para ter **Size** igual a 20, o número máximo válido para a propriedade **Index** será 19 e o mínimo será 0. Esta propriedade pode ser alterada após iniciada a comunicação. O valor padrão desta propriedade é 0, porém ao mapear os elementos de um bloco o E3 Studio automaticamente ajusta o parâmetro **Index** para um valor de acordo com o especificado. Exemplo:

```
Sub Elemento_OnStartRunning()  
    Index = 15  
End Sub
```

3.4.2.10 PercentDeadBand

9 A propriedade **PercentDeadBand** determina a variação mínima do valor (banda morta) de um elemento de bloco, para que seu valor no E3 seja atualizado. Este valor é especificado como um percentual da diferença entre as propriedades **DeviceHigh** e **DeviceLow**. Esta propriedade só será utilizada caso a propriedade **EnableDeadBand** do mesmo elemento esteja configurada para True. Se a propriedade **PercentDeadband** for igual a 0, o elemento de bloco não terá banda morta, e qualquer variação no seu valor será repassada para o E3. Caso contrário, só será enviado para o E3 um novo valor cuja diferença, em relação ao valor atual no E3, seja maior que a banda morta. O valor padrão desta propriedade é 0.

3.4.2.11 Quality

9 A propriedade **Quality** informa qual a qualidade do valor contido na propriedade **Value**. Cada vez que o Driver atribui um novo valor para o Elemento, ele também configura a qualidade daquele dado. Esta propriedade é somente para leitura. O valor padrão desta propriedade é 0 (Qualidade Ruim).

NOTA: Para maiores informações sobre qualidade, consulte o tópico **Propriedade Quality** ou o tópico **Qualidade** no **Manual do Usuário**.

3.4.2.12 RawValue

▼ Acessa o valor original do elemento anterior à escala, ou seja, independentemente da configuração da propriedade **EnableScaling**. Então, se esta propriedade for **False**, as propriedades **Value** e **RawValue** apresentam o mesmo comportamento.

3.4.2.13 UseBitFields

☑ Se esta propriedade estiver configurada para **True**, toda vez que o valor da propriedade **Value** for modificada, ele atualizará os bits referentes às propriedades **Bit00** até **Bit31**. Da mesma forma, atualizará o valor de **Value** toda vez que alguma das propriedades **Bit00** até **Bit31** mudar de valor e posteriormente enviará para o equipamento ou para o dispositivo se a propriedade **AllowWrite** do bloco de comunicação estiver configurada para **True**. Caso contrário, os bits não sofrerão nem acarretarão qualquer mudança. Esta propriedade pode ser atualizada após ter sido iniciada a comunicação. O valor padrão desta propriedade é **False**.

3.4.2.14 Value

▼ Atualizada sempre que uma nova leitura válida de um valor do equipamento ou do dispositivo for feita, utilizando-se a parametrização **B1** até **B4** do Bloco de Comunicação onde este elemento estiver inserido, e considerando-se sua posição na tabela lida através da propriedade **Index**. O tipo desta variável (número inteiro, ponto flutuante, texto) depende do driver ao qual o elemento está associado e da sua parametrização.

Esta propriedade somente será atualizada desta forma se a propriedade **AllowRead** do objeto Bloco de Comunicação ao qual o elemento pertence estiver configurada para **True**, e quando não houver erros de comunicação (neste caso somente as propriedades **Quality** e **TimeStamp** serão atualizadas), de acordo com o tempo de varredura definido em **Scan**. Outra forma de utilizar esta propriedade é para escrever valores no equipamento ou no dispositivo, bastando para isto atribuir um novo valor à propriedade **Value** ou a algum dos bits **Bit00** a **Bit31**. Neste caso, a propriedade **AllowWrite** do objeto Bloco de Comunicação deve estar configurada para **True**.

Esta também é a propriedade padrão do objeto Elemento de Bloco. Sendo assim, uma referência por valor a um Elemento de Bloco não necessita obrigatoriamente explicitar a propriedade **Value** para acessar o valor. Caso esta propriedade não esteja sendo atualizada, verifique se a propriedade **Index** está corretamente configurada. O valor padrão desta propriedade é vazio (sem valor). Exemplo:

```

Sub Botao1_Click()
' Acessa um elemento e mostra o valor atual
' elm1 é um objeto tipo Bloco de comunicação
Set obj = Application.GetObject("Driverde_
Comunicacao1.Bloco1.elm1")
MsgBox "Valor atual de elm1: " & obj.Value
' Isto também pode ser feito de outra forma,
' sem mostrar a propriedade Value, que é padrão
MsgBox "Valor atual de elm1: " & obj
End Sub

```

3.5 Driver de Comunicação OPC

Esta seção contém informações sobre os eventos e propriedades do objeto Driver de Comunicação OPC. Este objeto não possui métodos associados.

3.5.1 Eventos

Esta seção contém informações sobre os eventos do objeto Driver de Comunicação OPC.

3.5.1.1 OnTagRead

OnTagRead(Tag)

Ocorre na leitura de um tag, sempre que um novo valor ou um erro for retornado pelo Servidor OPC. Ou seja, se o valor ou a qualidade do tag não mudarem, o evento não será disparado. Para que este evento ocorra, a propriedade **EnableDriverEvent** do tag deve obrigatoriamente estar habilitada.

3.5.1.2 OnTagWrite

OnTagWrite(Tag, Succeeded, User)

Ocorre quando uma escrita for disparada em qualquer tag do Driver. Para que o evento ocorra, a propriedade **EnableDriverEvent** do tag deve obrigatoriamente estar habilitada. Se a escrita for assíncrona, o evento **OnTagWrite** será gerado apenas quando o servidor enviar a resposta indicando o efetivo sucesso ou não da escrita.

Variáveis do evento OnTagWrite

NOME	DESCRIÇÃO
Tag	Uma referência para o objeto Tag que está sofrendo a escrita. Por exemplo, pode-se acessar a propriedade do tag usando a sintaxe Tag.DocString .

Succeeded	Um valor Booleano que indica sucesso ou falha na escrita.
User	Texto que retorna o usuário que fez a escrita. Atualmente tem sempre o valor "System".

3.5.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Driver de Comunicação OPC.

3.5.2.1 Compatibility

 Permite controlar o uso das interfaces padrão OPC pelo Cliente OPC do E3. As opções disponíveis são as seguintes:

- **0 - AnyVersion:** funcionamento normal (recomendado), o OPC driver irá comunicar com servidores DA 2.0xe 1.0a (dando preferência ao uso de interfaces 2.0x).
- **1 - Version10A:** permite forçar a comunicação no padrão DA 1.0a para servidores que suportam DA 2.0xe 1.0a.
- **2 - Version20:** permite forçar a comunicação apenas no padrão OPC DA 2.0.

Essa propriedade não pode ser modificada quando a comunicação do cliente OPC estiver habilitada (tanto no E3 Studio, quanto em tempo de execução).

NOTA: Apenas em último caso o driver deve ser configurado num valor diferente de **0 - AnyVersion** (valor padrão). Esta propriedade é de uso avançado, e se aplica restritamente para contornar alguma situação de possível incompatibilidade com algum servidor OPC específico.

3.5.2.2 ReconnectPeriod

 Controla o período da conexão com o Servidor OPC. Se a conexão for perdida, o driver pára e reinicia até que esta ação retorne sucesso. O período é configurado em milissegundos e quando o valor é configurado para 0, a reconexão é desabilitada. Pelo fato do Driver OPC ser parado e iniciado, os eventos **BeforeStart** e **AfterStop** serão gerados. Quando a conexão for perdida, todos os tags relacionados serão desconectados do seu estado atual (bad/quality/null value). Exemplo:

```

Sub DriverOPCL_AfterStart()
    Application.GetObject("DriverOPCL.GrupoOPCL")._
        ReconnectPeriod = 0
End Sub

```

3.5.2.3 ServerID

A Determina o servidor ao qual o Driver OPC deve se conectar. Apesar do valor padrão da propriedade ser vazio, se o campo estiver vazio, o objeto OPC não irá conectar. Esta propriedade só pode ser modificada quando o Driver OPC não está conectado. Exemplo:

```

Sub CommandButton1_Click()
    Set Opc = Application.GetObject("DriverOPCL")
    Opc.Deactivate
    Opc.ServerId = "EclipseSCADA.OPCSvr.1"
    Opc.ServerMachine = "\\server2"
    Opc.Activate
End Sub

```

3.5.2.4 ServerMachine

A A propriedade **ServerMachine** determina o endereço da estação onde está rodando o servidor OPC. Para aplicações que estão rodando localmente, esta propriedade pode ficar em branco (padrão). Caso contrário, será necessário especificar o caminho (por exemplo, \\NomeServidor). Esta propriedade só pode ser modificada quando o driver OPC estiver desconectado. Exemplo:

```

Sub CommandButton1_Click()
    Set Opc = Application.GetObject("DriverOPCL")
    Opc.Deactivate
    Opc.ServerId = "EclipseSCADA.OPCSvr.1"
    Opc.ServerMachine = "\\server2"
    Opc.Activate
End Sub

```

3.5.2.5 ServerName

A A propriedade **ServerName** retorna o nome ou a descrição do servidor OPC. Esta propriedade é diferente da propriedade **ServerID**, que é um código. Exemplo:

```

Sub Driver OPC1_AfterStart()
    MsgBox _
        Application.GetObject("DriverOPCL.GrupoOPCL").ServerName
End Sub

```

3.5.2.6 ServerStatus

 Determina o estado da conexão com o servidor OPC. As opções disponíveis são as seguintes:

Opções disponíveis para ServerStatus

OPÇÃO	DESCRIÇÃO
-1 - ServerStatus_Unknown	O Driver OPC está conectado ao servidor OPC mas o estado não é informado ou o cliente OPC possui a propriedade ReconnectPeriod em 0.
0 - ServerStatus_NoConnected	O Driver OPC não está conectado ao servidor OPC. Isso acontece quando, por exemplo, o Driver OPC não está ativo, ou a conexão não foi estabelecida por algum problema qualquer.

Os valores a seguir são informados apenas quando a propriedade **ReconnectPeriod** é diferente de 0. Esta passagem de período especificando o estado é buscada no servidor. Caso o estado não seja informado corretamente, a propriedade pode manter o valor igual a -1, ou a desconexão pode ser detectada neste caso, levando a propriedade **ServerStatus** para 0. Os valores são baseados nos cinco status padrão definidos para servidores OPC.

Opções disponíveis para ReconnectPeriod diferente de zero

OPÇÃO	DESCRIÇÃO
1 - ServerStatus_Running	O servidor está rodando normalmente.
2 - ServerStatus_Failed	O servidor não está rodando. Um erro não especificado ocorreu no servidor.
3 - ServerStatus_NoConfig	O servidor está rodando, porém sem informações de configuração.
4 - ServerStatus_Suspended	O servidor foi suspenso temporariamente.
5 - ServerStatus_Test	O servidor está em Modo de Testes.

Exemplo:

```
Sub CommandButton1_Click()  
    Dim status  
    status = Application.GetObject("DriverOPC1").ServerStatus  
    MsgBox "O status do driver é " & status  
    Select Case status  
    Case -1  
        MsgBox "O DriverOPC está conectado ao servidor OPC mas_  
            seu estado não foi informado."  
    Case 0  
        MsgBox "O DriverOPC não está conectado ao servidor OPC"  
    Case 1  
        MsgBox "O servidor está rodando normalmente"  
    Case 2  
        MsgBox "O servidor não está rodando"  
    Case 3  
        MsgBox "O servidor está rodando mas sem informações de_  
            configuração"  
    Case 4  
        MsgBox "O servidor foi suspenso temporariamente"  
    Case 5  
        MsgBox "O servidor está em Modo de Testes"  
    End Select  
End Sub
```

NOTA: Para obter o comportamento equivalente a uma propriedade booleana, o recomendado é usar **ServerStatus** diferente de 0 (zero). Isso basicamente diferencia apenas entre haver a conexão ou não, desconsiderando estados mais específicos do servidor. Além disso, dessa forma, a expressão independe do uso da propriedade **ReconnectPeriod** diferente de 0 (zero).

3.5.2.7 WriteFeedbackMode

 Esta propriedade permite controlar a confirmação (ou *feedback*) das escritas em tags. As opções de configuração dessa propriedade estão descritas na tabela a seguir.

Opções disponíveis para a propriedade WriteFeedbackMode

OPÇÃO	DESCRIÇÃO
0 - wfWaitNextRead	Após a escrita, aguarda normalmente a próxima leitura.
1 - wfImmediateReadAfterWrite	Força uma leitura assíncrona de dispositivo logo após cada escrita.
2 - wfTrustWriteSuccess	O valor escrito é assumido pelo tag imediatamente, caso a escrita tenha funcionado.

O valor padrão dessa propriedade é **0 - wfWaitNextRead** para aplicativos criados antes da implementação desta propriedade, e **1 - wfImmediateReadAfterWrite** para aplicativos criados a partir de sua implementação.

NOTAS:

- A propriedade **WriteFeedbackMode** do Driver OPC não pode ser alterada com o objeto ativo.
- Na opção **2 - wfTrustWriteSuccess**, no caso de escritas assíncronas, o valor é assumido no tag após o agendamento da escrita, se a operação foi bem sucedida. Porém, se a escrita falhar posteriormente, o valor no tag pode ficar errado. No caso da escrita síncrona, o valor é assumido logo após o término da escrita, se houver sucesso.
- Veja também a propriedade **WriteFeedbackMode** do objeto Driver de Comunicação, cujo comportamento é semelhante.

3.6 Tag de Comunicação OPC

Esta seção contém informações sobre as propriedades do objeto Tag de Comunicação OPC. Este objeto não possui eventos nem métodos associados.

3.6.1 Propriedades

Esta seção contém as propriedades do objeto Tag de Comunicação OPC.

3.6.1.1 AdviseType



Controla o modo **Advise**. As opções disponíveis são:

Opções disponíveis para AdviseType

OPÇÃO	DESCRIÇÃO
0 - AlwaysInAdvise	O tag é mantido atualizado se a propriedade AllowRead do Tag OPC for True e a propriedade Enabled do Grupo OPC também for True.
1 - AdviseWhenLinked	O tag só é atualizado se a propriedade AllowRead do Tag OPC e a propriedade Enabled do Grupo OPC forem True, e o tag estiver associado a algum objeto ativo, por

OPÇÃO	DESCRIÇÃO
	<p>exemplo, um Display de uma Tela aberta, um Alarme habilitado, entre outros. A associação do tag para esse propósito pode ser feita nas seguintes propriedades: Value, RawValue, TimeStamp, Quality e Bit00 até Bit31 de Tags OPC.</p>

Exemplo:

```
Sub CommandButton3_Click()
    MsgBox Application._
        GetObject("DriverOPC.GrupoOPC.TagOPC1").AdviseType
End Sub
```

3.6.1.2 AllowRead

Define se este tag deverá ou não ser lido pelo driver OPC. Caso a propriedade esteja configurada para True, o driver automaticamente atualizará as propriedades **Value** e Bits (de **Bit00** até **Bit31**) deste objeto em intervalos de tempo. Caso contrário, este tag OPC não será lido. Esta propriedade pode ser modificada em tempo de execução. O valor padrão desta propriedade é True. Exemplo:

```
Sub Botao1_Click()
    ' Faz com que seja parada a leitura do tag
    Set obj = Application.GetObject("Driver1.tag")
    obj.AllowRead = False
End Sub
```

3.6.1.3 AllowWrite

Define se este tag deverá ou não ser escrito automaticamente quando a propriedade **Value** ou qualquer propriedade Bit (de **Bit00** até **Bit31**) for modificada. Se a propriedade estiver em True, as modificações serão enviadas para o equipamento ou dispositivo associado ao driver OPC; do contrário, as modificações serão ignoradas. O valor padrão desta propriedade é True. Exemplo:

```
Sub Botao1_Click()
    ' Faz com que seja desabilitada a escrita do tag
    Set obj = Application.GetObject("Driver1.tag")
    obj.Allowwrite = False
End Sub
```

3.6.1.4 Bit00 - Bit31

☑ As propriedades de bit **Bit00** até **Bit31** representam juntas os 32 bits do valor presente na propriedade **Value** do objeto Tag de Comunicação OPC, sendo **Bit00** o bit menos significativo e **Bit31** o bit mais significativo. Modificar cada um destes bits implica na igual modificação da propriedade **Value** do tag, e vice-versa, mas isto somente ocorre quando a propriedade **UseBitFields** estiver configurada para True. O valor padrão destas propriedades é False.

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.6.1.5 DataType

📄 Propriedade somente de leitura. Determina o tipo de dado associado ao tag OPC (ver tabela a seguir).

Opções disponíveis para DataType

OPÇÃO	DESCRIÇÃO
0 - _Undefined	Valor unidimensional não definido (Vazio).
1 - _Null	Valor nulo.
2 - _Integer	Valor unidimensional inteiro de 16 bits sinalizado.
3 - _Long	Valor unidimensional inteiro de 32 bits sinalizado.
4 - _Single	Valor unidimensional de ponto flutuante de 32 bits.
5 - _Double	Valor unidimensional de ponto flutuante de 64 bits.
6 - _Currency	Valor unidimensional monetário com 4 casas decimais.
7 - _Date	Valor de data/hora.
8 - _String	Valor literal - Texto.
9 - _Object	Valor unidimensional de referência a um objeto.
10 - _Error	Valor unidimensional de código de erro.

OPÇÃO	DESCRIÇÃO
11 - _Boolean	Valor unidimensional booleano (verdadeiro ou falso).
12 - _Variant	Dados de qualquer tipo usados para objetos e outros valores para os quais o tipo de dados é desconhecido.
13 - _UnkObject	Valor unidimensional de referência a um objeto.
14 - _Decimal	Valor unidimensional de ponto flutuante de 96 bits.
36 - _Record	Valor unidimensional de gravação.
16 - _Char	Valor unidimensional inteiro de 8 bits.
17 - _Byte	É utilizado para criação de DLLs e para OLE. Ocupa 1 byte de memória.
18 - _Word	Valor unidimensional inteiro de 16 bits.
19 - _Dword	Valor unidimensional inteiro de 32 bits.
20 - _LongLong	Valor unidimensional inteiro de 64 bits sinalizado.
21 - _DDWord	Valor unidimensional inteiro de 64 bits.
22 - _Integer_	Números sem decimais, cujo intervalo de ocorrência deverá ser de -32.768 a 32.767. Ocupa 2 bytes.
23 - _UInteger	Número inteiro sem sinal (no caso equivale a um DWORD), que vai de 0 a 4294967295 ($2^{32} - 1$).
8194 - _ArrInteger	Array unidimensional de valores integer (inteiros).
8195 - _ArrLong	Array unidimensional de valores inteiros de 32 bits sinalizados.
8196 - _ArrSingle	Array unidimensional de ponto flutuante de 32 bits.
8197 - _ArrDouble	Array unidimensional de valor de ponto flutuante de 64 bits.
8198 - _ArrCurrency	Array unidimensional de valor monetário com 4 casas decimais.
8199 - _ArrDate	Array unidimensional de valor de data/hora.

OPÇÃO	DESCRIÇÃO
8200 - _ArrString	Array unidimensional de valor literal - Texto.
8201 - _ArrObject	Array unidimensional de valor de referência a um objeto.
8202 - _ArrError	Array unidimensional de valor de código de erro.
8203 - _ArrBoolean	Array unidimensional de valor booleano (verdadeiro ou falso).
8204 - _ArrVariant	Array de dados de qualquer tipo usados para objetos e outros valores para os quais o tipo de dados é desconhecido.
8205 - _ArrUnkObject	Array unidimensional de valor de referência a um objeto.
8206 - _ArrDecimal	Array unidimensional de valor de ponto flutuante de 96 bits.
8228 - _ArrRecord	Array unidimensional de valor record.
8208 - _ArrChar	Array unidimensional de valor char.
8209 - _ArrByte	Array unidimensional de bytes, que são valores utilizados para criação de DLLs e para OLE. Ocupa 1 byte de memória.
8210 - _ArrWord	Array unidimensional de valor inteiro de 32 bits.
8211 - _ArrDWord	Array unidimensional de valor inteiro de 32 bits.
8212 - _ArrLongLong	Array unidimensional de valor inteiro de 16 bits.
8213 - _ArrDDWord	Array unidimensional de valor inteiro de 32 bits sinalizado.
8214 - _ArrInteger_	Array unidimensional de Inteiro de 16 bits sinalizado
8215 - _ArrUInteger	Array unidimensional de número inteiro sem sinal (no caso equivale a um DWORD), que vai de 0 a 4294967295 ($2^{32} - 1$).

3.6.1.6 DeviceHigh

9 A propriedade **DeviceHigh** define o valor máximo atingido por um tag no equipamento ou no dispositivo. Esta propriedade será utilizada para fazer o ajuste de escala do valor proveniente do equipamento ou do dispositivo antes de ser atribuído à propriedade **Value**. Da mesma forma, no momento da escrita, é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** estiver configurada para True, senão, os valores permanecem os mesmos. O valor padrão desta propriedade é 1000. Exemplo:

```
Sub Tag_OnStartRunning()  
    ' Faz o ajuste de escala de um tag de temperatura  
    ' que varia de 0 a 255 no PLC, mas significam na  
    ' prática de 0 a 100 graus Celsius  
    EU = "Graus Celsius"  
    EnableScaling = True  
    DeviceHigh = 255  
    DeviceLow = 0  
    EUHigh = 100  
    EULow = 0  
End Sub
```

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.6.1.7 DeviceLow

 A propriedade **DeviceLow** define o valor mínimo atingido por um tag no equipamento ou no dispositivo. Esta propriedade será utilizada para fazer o ajuste de escala do valor proveniente do equipamento ou do dispositivo antes de ser atribuído à propriedade **Value**. Da mesma forma, no momento da escrita, é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True, senão as configurações permanecem as mesmas. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Tag_OnStartRunning()  
    ' Faz o ajuste de escala de um tag de temperatura  
    ' que varia de 0 a 255 no PLC, mas significam na  
    ' prática de 0 a 100 graus Celsius  
    EU = "Graus Celsius"  
    EnableScaling = True  
    DeviceHigh = 255  
    DeviceLow = 0  
    EUHigh = 100  
    EULow = 0  
End Sub
```

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.6.1.8 EnableDriverEvent

☑ Esta propriedade é utilizada para controlar a geração do evento **OnTagRead**, que ocorre no Driver de Comunicação OPC que contém o tag. Se a propriedade **EnableDriverEvent** do tag estiver configurada para True, a cada leitura vinda do servidor OPC, seja com erro ou não, será gerado o evento **OnTagRead** no objeto Driver OPC que contem este tag. Caso contrário, o evento não ocorre. Também quando a propriedade **EnableDriverEvent** tiver o valor True, a cada escrita que seja enviada para o servidor OPC, será gerado o evento **OnTagWrite** no objeto OPCDriver que contem a tag. Se a escrita for assíncrona, o evento OnTagWrite será gerado apenas quando o servidor enviar a resposta indicando o efetivo sucesso ou não da escrita. Nesse caso, o evento será gerado apenas se a propriedade **EnableDriverEvent** for True nesse momento, e não no momento do envio da escrita. O valor padrão dessa propriedade é False.

3.6.1.9 EnableScaling

☑ Se esta propriedade estiver configurada para True, todo valor proveniente do equipamento ou do dispositivo sofrerá ajuste de escala conforme as propriedades **EUHigh** e **EULow** antes de ser atribuído à propriedade **Value**. Caso a propriedade **EnableScaling** tenha o valor configurado para False, nenhum ajuste de escala será efetuado em nenhum dos dois sentidos (tanto para escrita como para leitura). O valor padrão desta propriedade é False. Exemplo:

```
Sub Tag_OnStartRunning()  
    ' Faz o ajuste de escala de um tag de temperatura  
    ' que varia de 0 a 255 no PLC, mas significam na  
    ' prática de 0 a 100 graus Celsius  
    EU = "Graus Celsius"  
    EnableScaling = True  
    DeviceHigh = 255  
    DeviceLow = 0  
    EUHigh = 100  
    EULow = 0  
End Sub
```

3.6.1.10 EU

A Identifica a unidade de engenharia representada pelo valor, como por exemplo, graus, metros, KW/h, etc. O valor desta propriedade é vazio. Exemplo:

```
Sub Tag_OnStartRunning()  
    ' Faz o ajuste de escala de um tag de temperatura  
    ' que varia de 0 a 255 no PLC, mas significam na  
    ' prática de 0 a 100 graus Celsius  
    EU = "Graus Celsius"  
    EnableScaling = True  
    DeviceHigh = 255  
    DeviceLow = 0  
    EUHigh = 100  
    EULow = 0  
End Sub
```

3.6.1.11 EUHigh

9 Define qual o valor máximo que deverá ser atribuído à propriedade **Value**, ajustando a escala ao valor do equipamento ou do dispositivo antes desta atribuição. Da mesma forma, no momento da escrita é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 1000. Exemplo:

```
Sub Tag_OnStartRunning()  
    ' Faz o ajuste de escala de um tag de temperatura  
    ' que varia de 0 a 255 no PLC, mas significam na  
    ' prática de 0 a 100 graus Celsius  
    EU = "Graus Celsius"  
    EnableScaling = True  
    DeviceHigh = 255  
    DeviceLow = 0  
    EUHigh = 100  
    EULow = 0  
End Sub
```

NOTA: Os valores dos bits (propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.6.1.12 EULow

9 Define qual o valor mínimo que deverá ser atribuído à propriedade **Value**, ajustando a escala ao valor do equipamento ou do dispositivo antes desta atribuição. Da mesma forma, no momento da escrita, é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Tag_OnStartRunning()  
    ' Faz o ajuste de escala de um tag de temperatura  
    ' que varia de 0 a 255 no PLC, mas significam na  
    ' prática de 0 a 100 graus Celsius  
    EU = "Graus Celsius"  
    EnableScaling = True  
    DeviceHigh = 255  
    DeviceLow = 0  
    EUHigh = 100  
    EULow = 0  
End Sub
```

NOTA: Os valores dos bits (propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.6.1.13 ItemID

A A propriedade **ItemID** determina o caminho que identifica o tag OPC no servidor OPC ao qual o driver OPC se conecta. A definição desse caminho é flexível e depende do servidor específico. Normalmente os servidores especificam um espaço de ID com itens hierárquicos, do tipo **ItemPai.ItemFilho.Tag1**. Esta propriedade funciona como uma chave única para os dados, considerando o *onde* ou o *quê* permite o servidor OPC se conectar à fonte dos dados. Apesar de o valor padrão ser vazio, é necessário especificar algum valor para que o tag OPC seja válido. Exemplo:

```
Sub TagOPC_Click()  
    MsgBox ItemID  
End Sub
```

3.6.1.14 Quality

9 A propriedade **Quality** informa qual a qualidade do valor contido na propriedade **Value**. Cada vez que o Driver atribui um novo valor para o Tag, ele também configura a qualidade daquele dado. Esta propriedade é somente para leitura. O valor padrão desta propriedade é 0 (Qualidade Ruim).

NOTA: Para maiores informações sobre qualidade, consulte o tópico **Propriedade Quality** ou o tópico **Qualidade no Manual do Usuário**.

3.6.1.15 RawValue

V Acessa o valor original do tag anterior à escala, ou seja, independentemente da configuração da propriedade **EnableScaling**. Então, se esta propriedade for **False**, as propriedades **Value** e **RawValue** apresentam o mesmo comportamento.

3.6.1.16 SyncWrite

☑ A propriedade **SyncWrite** determina o tipo de escrita usada por um tag OPC. Se a propriedade estiver configurada para **True**, a escrita será síncrona, isto é, o driver OPC espera o resultado da escrita do servidor. Caso contrário, a escrita é assíncrona, isto é, o valor do tag OPC será enviado e o processamento do OPC Driver continuará imediatamente. O valor padrão desta propriedade é **False**.

NOTA: No modo assíncrono (propriedade configurada para **False**), a performance da comunicação tende a ser melhor, porém no modo síncrono (propriedade configurada para **True**), o sucesso da operação de escrita é verificado e informado.

3.6.1.17 TimeStamp

 A propriedade **TimeStamp** é atualizada sempre que ocorrer alguma mudança de valor ou de estado nas propriedades **Value** ou **Quality**. Ele informa qual a data/hora associada tanto ao valor quanto à qualidade do tag OPC. Esta é uma propriedade somente para leitura. O valor padrão desta propriedade é 00:00:00.

3.6.1.18 UseBitFields

 Caso a propriedade **UseBitFields** tenha o valor True, toda vez que o valor da propriedade **Value** for modificada ele atualizará os bits referentes às propriedades **Bit00** até **Bit31**. Da mesma forma atualizará o valor de **Value** toda vez que alguma das propriedades **Bit00** até **Bit31** mudar de valor e posteriormente enviará para o equipamento ou dispositivo se a propriedade **AllowWrite** estiver configurada para True. Caso contrário, se a propriedade **UseBitFields** estiver configurada para False, os bits não sofrerão qualquer mudança. Esta propriedade pode sofrer atualização após ter sido iniciada a comunicação. O valor padrão desta propriedade é False. Exemplo:

```
Sub TagOPC1_Click()  
    UseBitFields = True  
End Sub
```

NOTA: Os valores dos bits (Propriedades **Bit00** até **Bit31**) não são afetados pelo ajuste de escala. Isto é, representam os bits do valor lido do equipamento, antes da conversão.

3.6.1.19 Value

 A propriedade **Value** é atualizada quando é feita uma nova leitura válida de um valor do equipamento ou do dispositivo utilizando sua parametrização, porém o tipo desta variável (número inteiro, ponto flutuante, texto) depende do driver OPC, ao qual ele está associado e da sua parametrização. Esta propriedade somente irá ser atualizada desta forma se a propriedade **AllowRead** estiver configurada para True e quando não houver erros de comunicação (neste caso somente as propriedades **Quality** e **TimeStamp** são atualizadas), mas de acordo com o tempo de varredura definido no grupo ao qual ele pertence. Outra forma de utilizar esta propriedade é escrever valores no equipamento ou dispositivo. Basta para isso, atribuir um novo valor para a propriedade **Value** ou algum dos bits **Bit00** até **Bit31**, desde que a propriedade **AllowWrite** esteja configurado para True. Esta propriedade também é a propriedade padrão do objeto tag OPC. Sendo assim, uma referência por valor a um objeto tag OPC não necessita obrigatoriamente explicitar a propriedade **Value** para acessar o valor. O valor padrão desta propriedade é vazio. Exemplo:

```

Sub Botao1_Click()
' Acessa um tag e mostra o valor atual
' tag1 é um objeto tipo OPCTag
Set obj = Application._
GetObject("Driverde_Comunicacao1.tag1")
MsgBox "valor atual de tag1: " & obj.Value
' Sem mostrar a propriedade Value, que é padrão
MsgBox "valor atual de tag1: " & obj
End Sub

```

3.7 Bloco de Comunicação OPC

Esta seção contém informações sobre as propriedades do objeto Bloco de Comunicação OPC. Este objeto não possui eventos nem métodos associados.

3.7.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Bloco de Comunicação OPC.

3.7.1.1 AdviseType

 Controla o modo **Advise**. As opções disponíveis são:

Opções disponíveis para AdviseType

OPÇÃO	DESCRIÇÃO
0 - AlwaysInAdvise	O tag é mantido atualizado se a propriedade AllowRead do Bloco OPC for True e a propriedade Enabled do Grupo OPC também for True.
1 - AdviseWhenLinked	O tag só é atualizado se a propriedade AllowRead do Bloco OPC e a propriedade Enabled do Grupo OPC forem True, e o tag estiver associado a algum objeto ativo, por exemplo, um Display de uma Tela aberta, um Alarme habilitado, entre outros. A associação do tag para esse propósito pode ser feita nas seguintes propriedades: Value , RawValue , Quality e Bit00 até Bit31 de Elementos de Bloco OPC, e Quality e TimeStamp de Blocos OPC.

Exemplo:

```

Sub CommandButton3_Click()
MsgBox Application._
GetObject("DriverOPC.GrupoOPC.SCRIPT1").AdviseType
End Sub

```

3.7.1.2 AllowRead

☑ Configure esta propriedade para definir se este bloco deverá ou não ser lido pelo driver OPC. Caso a propriedade **AllowRead** esteja configurada para True, então o driver automaticamente atualizará as propriedades **Value** e Bits (de **Bit00** até **Bit31**) dos elementos do bloco OPC em intervalos de tempo. Caso contrário, este bloco OPC não será lido se a propriedade **AllowRead** estiver configurada para False. Esta propriedade pode ser modificada em tempo de execução. O valor padrão desta propriedade é True. Exemplo:

```
Sub Botao1_Click()  
    ' Faz com que seja parada a leitura do tag  
    Set obj = Application.GetObject("Driver1.tag")  
    obj.AllowRead = False  
End Sub
```

3.7.1.3 AllowWrite

☑ Configure esta propriedade para definir se este tag deverá ou não ser escrito automaticamente quando a propriedade **Value** ou qualquer propriedade Bit (de **Bit00** até **Bit31**) for modificada. Se a propriedade estiver em True, as modificações serão enviadas para o equipamento ou dispositivo associado ao Driver OPC, senão as modificações serão ignoradas. Estando esta propriedade configurada para True, então o driver automaticamente atualizará as propriedades **Value** e Bits (de **Bit00** até **Bit31**) deste objeto, em intervalos de tempo. Caso contrário, este Bloco OPC não será lido. O valor padrão desta propriedade é True. Exemplo:

```
Sub Botao1_Click()  
    Set obj = Application.GetObject("Driver1.tag")  
    obj.Allowwrite = False  
End Sub
```

3.7.1.4 DataType

📄 Propriedade somente de leitura. Determina o tipo de dado associado ao bloco OPC (ver tabela a seguir).

Opções disponíveis para DataType

OPÇÃO	DESCRIÇÃO
0 - _Undefined	Valor unidimensional não definido (Vazio).
1 - _Null	Valor nulo.
2 - _Integer	Valor unidimensional Inteiro de 16 bits sinalizado.
3 - _Long	Valor unidimensional inteiro de 32 bits sinalizado.

OPÇÃO	DESCRIÇÃO
4 - _Single	Valor unidimensional de ponto flutuante de 32 bits.
5 - _Double	Valor unidimensional de ponto flutuante de 64 bits.
6 - _Currency	Valor unidimensional monetário com 4 casas decimais.
7 - _Date	Valor de data/hora.
8 - _String	Valor literal - Texto.
9 - _Object	Valor unidimensional de referência a um objeto.
10 - _Error	Valor unidimensional de código de erro.
11 - _Boolean	Valor unidimensional booleano (verdadeiro ou falso).
12 - _Variant	Dados de qualquer tipo usados para objetos e outros valores para os quais o tipo de dados é desconhecido.
13 - _UnkObject	Valor unidimensional de referência a um objeto.
14 - _Decimal	Valor unidimensional de ponto flutuante de 96 bits.
36 - _Record	Valor unidimensional de gravação.
16 - _Char	Valor unidimensional inteiro de 8 bits.
17 - _Byte	É utilizado para criação de DLLs e para OLE. Ocupa 1 byte de memória.
18 - _Word	Valor unidimensional inteiro de 16 bits.
19 - _Dword	Valor unidimensional inteiro de 32 bits.
20 - _LongLong	Valor unidimensional inteiro de 64 bits sinalizado.
21 - _DDWord	Valor unidimensional inteiro de 64 bits.
22 - _Integer_	Números sem decimais, cujo intervalo de ocorrência deverá ser de -32.768 a 32.767. Ocupa 2 bytes.
23 - _UInteger	Número inteiro sem sinal (no caso equivale a um

OPÇÃO	DESCRIÇÃO
	DWORD), que vai de 0 a 4294967295 ($2^{32} - 1$).
8194 - _ArrInteger	Array unidimensional de valores integer (inteiros).
8195 - _ArrLong	Array unidimensional de valores inteiros de 32 bits sinalizados.
8196 - _ArrSingle	Array unidimensional de ponto flutuante de 32 bits.
8197 - _ArrDouble	Array unidimensional de valor de ponto flutuante de 64 bits.
8198 - _ArrCurrency	Array unidimensional de valor monetário com 4 casas decimais.
8199 - _ArrDate	Array unidimensional de valor de data/hora.
8200 - _ArrString	Array unidimensional de valor literal - Texto.
8201 - _ArrObject	Array unidimensional de valor de referência a um objeto.
8202 - _ArrError	Array unidimensional de valor de código de erro.
8203 - _ArrBoolean	Array unidimensional de valor booleano (verdadeiro ou falso).
8204 - _ArrVariant	Array de dados de qualquer tipo usados para objetos e outros valores para os quais o tipo de dados é desconhecido.
8205 - _ArrUnkObject	Array unidimensional de valor de referência a um objeto.
8206 - _ArrDecimal	Array unidimensional de valor de ponto flutuante de 96 bits.
8228 - _ArrRecord	Array unidimensional de valor record.
8208 - _ArrChar	Array unidimensional de valor char.
8209 - _ArrByte	Array unidimensional de bytes, que são valores utilizados para criação de DLLs e para OLE. Ocupa 1 byte de memória.

OPÇÃO	DESCRIÇÃO
8210 - _ArrWord	Array unidimensional de valor inteiro de 32 bits.
8211 - _ArrDWord	Array unidimensional de valor inteiro de 32 bits.
8212 - _ArrLongLong	Array unidimensional de valor inteiro de 16 bits.
8213 - _ArrDDWord	Array unidimensional de valor inteiro de 32 bits sinalizado.
8214 - _ArrInteger_	Array unidimensional de Inteiro de 16 bits sinalizado.
8215 - _ArrUInteger	Array unidimensional de número inteiro sem sinal (neste caso equivale a um DWORD), que vai de 0 a 4294967295 ($2^{32} - 1$).

3.7.1.5 EnableDriverEvent

■ Esta propriedade é utilizada para controlar a geração do evento **OnTagRead**, que ocorre no Driver de Comunicação OPC que contém o bloco. Se a propriedade **EnableDriverEvent** do bloco estiver configurada para True, a cada leitura vinda do servidor OPC, seja com erro ou não, será gerado o evento **OnTagRead** no objeto Driver OPC que contém este bloco. Caso contrário, o evento não ocorre. Também quando a propriedade **EnableDriverEvent** tiver o valor True, a cada escrita que seja enviada para o servidor OPC será gerado o evento **OnTagWrite** no objeto OPC Driver que contém o bloco. Se a escrita for assíncrona, o evento **OnTagWrite** será gerado apenas quando o servidor enviar a resposta indicando o efetivo sucesso ou não da escrita. Nesse caso, o evento será gerado apenas se a propriedade **EnableDriverEvent** for True nesse momento, e não no momento do envio da escrita. O valor padrão dessa propriedade é False.

3.7.1.6 ItemID

A propriedade **ItemID** determina o caminho que identifica o Bloco OPC no servidor ao qual o Driver OPC se conecta. A definição deste caminho é flexível e depende do servidor específico. Normalmente, os servidores especificam um espaço de ID com itens hierárquicos, do tipo **ItemPai.ItemFilho.Tag1**. A propriedade **ItemID** funciona como uma chave única para os dados, considerando o *onde* ou o *quê* que permite ao servidor OPC se conectar à fonte dos dados. Apesar de o valor padrão ser vazio, é necessário especificar algum valor para que o Bloco OPC seja válido. Exemplo:

```
Sub BlocoOPC1_OnStartRunning()
MsgBox ItemID
End Sub
```

3.7.1.7 Quality

 A propriedade **Quality** informa qual a qualidade do valor contido na propriedade **Value**. Cada vez que o Driver atribui um novo valor para o Bloco, ele também configura a qualidade daquele dado. Esta propriedade é somente para leitura. O valor padrão desta propriedade é 0 (Qualidade Ruim).

NOTA: Para maiores informações sobre qualidade, consulte o tópico **Propriedade Quality** ou o tópico **Qualidade no Manual do Usuário**.

3.7.1.8 Size

 Define o tamanho do conjunto de valores do bloco. Consulte a documentação do driver utilizado, para conhecer qual o limite para esta propriedade de acordo com os parâmetros *B1* até *B4*. Criando os elementos para o bloco, possibilita acessar os valores lidos e também permite a escrita de valores para o equipamento ou dispositivo. Esta propriedade não pode ser modificada após iniciada a comunicação. O valor padrão desta propriedade é 0. Exemplo:

```
Sub BlocoOPCL_OnStartRunning()
    Size = 12
End Sub
```

3.7.1.9 SyncWrite

 A propriedade **SyncWrite** determina o tipo de escrita usada por um Bloco OPC. Se a propriedade estiver configurada para True, a escrita será síncrona, isto é, o Driver OPC espera o resultado da escrita do servidor. Caso contrário, a escrita é assíncrona, isto é, o valor do OPCTag será enviado e o processamento do Driver OPC continuará imediatamente. O valor padrão desta propriedade é False.

NOTA: No modo assíncrono (propriedade desabilitada), a performance da comunicação tende a ser melhor, porém no modo síncrono (propriedade habilitada), o sucesso da operação de escrita é verificado e informado.

3.7.1.10 TimeStamp

 A propriedade **TimeStamp** é atualizada sempre que ocorrer alguma mudança de valor ou de estado nas propriedades **Value** ou **Quality**. Ele informa qual a data/hora associada tanto ao valor quanto à qualidade do Bloco OPC. Esta é uma propriedade somente para leitura. O valor padrão desta propriedade é 00:00:00.

3.8 Elemento de Bloco de Comunicação OPC

Esta seção contém informações sobre propriedades do objeto Elemento de Bloco de Comunicação OPC. Este objeto não possui eventos nem métodos associados.

3.8.1 Propriedades

Esta seção contém as propriedades do objeto Elemento de Bloco de Comunicação OPC.

3.8.1.1 Bit00 - Bit31

☑ As propriedades de bit **Bit00** até **Bit31** representam juntas os 32 bits do valor presente na propriedade **Value** do objeto Elemento de Bloco OPC, sendo **Bit00** o bit menos significativo e **Bit31** o bit mais significativo. Modificar cada um destes bits implica na igual modificação da propriedade **Value** do elemento, e vice-versa, mas isto somente ocorre quando a propriedade **UseBitFields** estiver configurada para True. O valor padrão destas propriedades é False.

3.8.1.2 DeviceHigh

9 A propriedade **DeviceHigh** define o valor máximo atingido por um elemento de bloco no equipamento ou no dispositivo. Esta propriedade será utilizada para fazer o ajuste de escala do valor proveniente do equipamento ou dispositivo antes de ser atribuído à propriedade **Value**. Da mesma forma, no momento da escrita é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 1000. Exemplo:

```
Sub Elemento_OnStartRunning()
' Faz o ajuste de escala de um elemento temperatura
' que varia de 0 a 255 no OPC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub
```

3.8.1.3 DeviceLow

9 A propriedade **DeviceLow** define o valor mínimo atingido por um elemento no equipamento ou no dispositivo. Esta propriedade será utilizada para fazer o ajuste de escala do valor proveniente do equipamento ou dispositivo antes de ser atribuído à propriedade **Value**. Da mesma forma, no momento da escrita, é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Elemento_OnStartRunning()
' Faz o ajuste de escala de um elemento temperatura
' que varia de 0 a 255 no OPC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub
```

3.8.1.4 EnableScaling

■ A propriedade **EnableScaling** habilita ou desabilita a escala de valor para o valor enviado e recebido do equipamento ou dispositivo. Se esta propriedade estiver configurada para True, todo valor proveniente do equipamento ou dispositivo sofrerá ajuste de escala conforme as propriedades **DeviceHigh**, **DeviceLow**, **EUHigh** e **EULow** antes de ser atribuído à propriedade **Value**. O mesmo ocorrerá quando uma escrita for necessária, é quando o valor presente em **Value** sofrerá ajuste de escala (sem com isso alterar a propriedade **Value**) e posteriormente será enviado para o equipamento ou dispositivo. Caso a propriedade **EnableScaling** tenha o valor False, nenhum ajuste de escala será efetuado em nenhum dos dois sentidos (escrita e leitura). O valor padrão desta propriedade é False. Exemplo:

```
Sub Elemento_OnStartRunning()
' Faz o ajuste de escala de um elemento temperatura
' que varia de 0 a 255 no OPC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub
```

3.8.1.5 EU

A Identifica a unidade de engenharia representada por seu valor, como por exemplo, graus, metros, KW/h, etc. Exemplo:

```

Sub Elemento_OnStartRunning()
' Faz o ajuste de escala de um tag de temperatura
' que varia de 0 a 255 no OPC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub

```

3.8.1.6 EUHigh

9 Define qual o valor máximo que deverá ser atribuído à propriedade **Value**, ajustando a escala ao valor do equipamento ou do dispositivo antes desta atribuição. Da mesma forma, no momento da escrita é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 1000. Exemplo:

```

Sub Elemento_OnStartRunning()
' Faz o ajuste de escala de um elemento temperatura
' que varia de 0 a 255 no OPC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub

```

3.8.1.7 EULow

9 Define qual o valor mínimo que deverá ser atribuído à propriedade **Value**, ajustando a escala ao valor do equipamento ou do dispositivo antes desta atribuição. Da mesma forma, no momento da escrita, é feita a operação inversa antes de enviar o valor para o driver. Esta conversão só ocorre quando a propriedade **EnableScaling** está configurada para True. O valor padrão desta propriedade é 0. Exemplo:

```

Sub Elemento_OnStartRunning()
' Faz o ajuste de escala de um elemento temperatura
' que varia de 0 a 255 no OPC, mas significam na
' prática de 0 a 100 graus Celsius
EU = "Graus Celsius"
EnableScaling = True
DeviceHigh = 255
DeviceLow = 0
EUHigh = 100
EULow = 0
End Sub

```

3.8.1.8 Index

9 Utilize esta propriedade para especificar a posição que um elemento ocupa entre os elementos configurados pela propriedade **Size** do objeto Bloco OPC no qual ele está inserido. Esta propriedade aceita valores de 0 até um número menor do que a valor definido em **Size**. Por exemplo, configurando um bloco OPC para ter **Size** igual a 20, o número máximo válido para a propriedade **Index** será 19 e o mínimo 0. Esta propriedade pode ser alterada após iniciada a comunicação. O valor padrão da propriedade é 0, porém ao mapear os elementos de um bloco o E3 Studio automaticamente configura o parâmetro **Index** para um valor de acordo com o especificado. Exemplo:

```
Sub Elemento1_OnRead()  
  MsgBox Index  
End Sub
```

3.8.1.9 Quality

9 A propriedade **Quality** representa o estado de qualidade do valor. O E3 utiliza o padrão OPC para informar essa propriedade. O campo é formado por uma palavra de 16 bits, sendo que os primeiros 8 bits são definidos na forma de 3 campos de bits: Qualidade, Substatus e Limite, nessa disposição: QQSSSSL. Qualidade pode ser ruim, incerta ou boa (ver tabela a seguir). Substatus tem informações complementares de Qualidade. Limite pode indicar informações de diagnóstico. Servidores que não suportam estas últimas informações, retornam zero nos bits correspondentes.

Os outros 8 bits (de maior ordem) estão disponíveis para uso de cada fabricante. Se esses bits forem utilizados, os bits de qualidade padrão ainda são utilizados para indicar quais deduções pode-se fazer sobre os dados retornados. Assim, é de responsabilidade do cliente interpretar o campo de informações específicas de qualidade para garantir que o servidor que está provendo essa informação usa as mesmas regras que o cliente.

Um driver ou servidor OPC que não suporta informação de qualidade sempre retorna o valor 192 (qualidade boa). Também é aceitável que um servidor retorne simplesmente qualidade ruim (0) ou boa (192) e não passe nenhuma informação nos bits de substatus e limite (retornar 0 nesses bits).

Uma lista dos valores possíveis de **Quality** e seu significado pode ser vista nas tabelas a seguir.

Faixas de valor para Quality

FAIXA	QUALIDADE	DESCRIÇÃO
0 - 63	RUIM	O valor não é utilizável pelas razões indicadas em substatus.

FAIXA	QUALIDADE	DESCRIÇÃO
64 - 127	INCERTA	A qualidade do valor é incerta pelas razões indicadas em substatus.
128 - 191	N/A	Não usado pelo padrão OPC.
192 - 255	BOA	A qualidade do valor é boa.

Opções possíveis para Limite

LIMITE	DESCRIÇÃO
Sem limite	O valor é livre para mover-se acima ou abaixo.
Baixo	O valor está limitado em algum valor inferior.
Alto	O valor está limitado em algum valor superior.
Constante	O valor é uma constante e não pode mover-se.

Substatus para Qualidade Ruim (0 - 63)

SUBSTATUS	DESCRIÇÃO	LIMITES			
		SEM	BAIXO	ALTO	CONSTANTE
Não especificado	O valor é ruim, mas nenhuma razão específica é conhecida.	000	001	002	003
Erro de configuração	Há algum problema de configuração no servidor. Por exemplo, o item em	004	005	006	007

SUBSTATUS	DESCRIÇÃO	LIMITES			
		SEM	BAIXO	ALTO	CONSTANTE
	questão pode ter sido apagado.				
Não conectado	A entrada deveria estar logicamente conectada a algo, mas não está. Esta qualidade pode refletir que não há valor disponível neste momento porque o mesmo pode não ter sido fornecido pela fonte de dados, por exemplo.	008	009	010	011
Falha de dispositivo	Uma falha de dispositivo foi detectada.	012	013	014	015
Falha de sensor	Uma falha de sensor foi detectada (o campo Limite pode prover informação adicional).	016	017	018	019
Último valor conhecido	Comunicação em falha, mas o último valor conhecido está disponível. A idade do valor pode ser determinada através da propriedade TimeStamp .	020	021	022	023

SUBSTATUS	DESCRIÇÃO	LIMITES			
		SEM	BAIXO	ALTO	CONSTANTE
Falha de comunicação	Comunicação em falha, mas não há valor conhecido disponível.	024	025	026	027
Fora de serviço	O bloco está fora de <i>scan</i> ou trancado.	028	029	030	031
N/A	Não usado.	032 - 063			

Substatus para Qualidade Incerta (64 - 127)

SUBSTATUS	DESCRIÇÃO	LIMITES			
		SEM	BAIXO	ALTO	CONSTANTE
Não especificado	O valor é incerto, mas nenhuma razão específica é conhecida.	064	065	066	067
Último valor utilizado	O dispositivo que estava providenciando o valor parou de fazê-lo. O valor deve ser considerado velho. Note que isso difere de um valor ruim com o substatus "Último Valor Conhecido". Esse estado é associado com uma falha de comunicação num valor	068	069	070	071

SUBSTATUS	DESCRIÇÃO	LIMITES			
		SEM	BAIXO	ALTO	CONSTANTE
	buscado possível de ser detectada. Este erro é associado à falha de alguma fonte externa que iria atualizar o valor em um período aceitável. A idade do dado pode ser determinada através da propriedade TimeStamp .				
N/A	Não usado.	072 - 079			
Sensor sem precisão	O valor está colado em um dos limites do sensor (nesse caso o campo Limite deve estar ajustado para 1 ou 2) ou o sensor está de algum modo conhecido fora de calibração (nesse caso, Limite deve ser 0).	080	081	082	083
Unidade de engenharia excedida	O valor retornado está fora dos limites definidos para esse parâmetro. Note que nesse caso o valor em Limite indica quais limites foram excedidos mas não necessariamente implica que o valor não possa ir além da faixa especificada.	084	085	086	087
Subnormal	O valor é derivado de múltiplas fontes e há um número menor do que o requerido de boas fontes.	088	089	090	091
N/A	Não usado.	092 - 127			

Substatus para Qualidade Boa (192 - 255)

SUBSTATUS	DESCRIÇÃO	LIMITE			
		SEM	BAIXO	ALTO	CONSTANTE
Não especificado	O valor é bom, mas nenhuma razão específica é conhecida.	192	193	194	195
Não utilizado	Não utilizado.	196 - 215			
Sobrescrita local	O valor foi sobrescrito. Tipicamente indica que a entrada foi desconectada e um valor entrado manualmente foi forçado.	216	217	218	219
Não especificado	O valor é bom, mas nenhuma razão específica é conhecida.	220 - 255			

3.8.1.10 RawValue

▼ Acessa o valor original do elemento anterior à escala, ou seja, independentemente da configuração da propriedade **EnableScaling**. Então, se esta propriedade for **False**, as propriedades **Value** e **RawValue** apresentam o mesmo comportamento.

3.8.1.11 UseBitFields

☑ Caso a propriedade **UseBitFields** tenha o valor **True**, toda vez que o valor da propriedade **Value** for modificada, ele atualizará os bits referentes às propriedades **Bit00** até **Bit31**. Da mesma forma, atualizará o valor de **Value** toda vez que alguma das propriedades **Bit00** até **Bit31** mudar de valor e posteriormente enviará para o equipamento ou dispositivo se a propriedade **AllowWrite** do bloco de comunicação

estiver configurada para True. Caso contrário, se a propriedade **UseBitFields** estiver configurada para False, os bits não sofrerão nem acarretarão qualquer mudança. Esta propriedade pode ser atualizada após ter sido iniciada a comunicação. O valor padrão desta propriedade é False.

3.8.1.12 Value

▼ Atualizada sempre que um novo valor é lido no servidor OPC, de acordo com a especificação da propriedade **ItemID** do Bloco OPC onde este objeto está inserido, e considerando a propriedade **Index** (que especifica a posição do elemento no vetor do bloco). O tipo desta propriedade (número inteiro, ponto flutuante, texto, etc.) depende do driver ao qual ele está associado e da sua parametrização.

Esta propriedade somente será atualizada desta forma se a propriedade **AllowRead** do objeto Bloco OPC ao qual o elemento pertence estiver configurada para True, e de acordo com o tempo de varredura definido na propriedade **Scan** do Grupo OPC que contém o bloco. Se a propriedade **AllowWrite** do objeto Bloco OPC estiver configurada em True, pode-se escrever valores no equipamento ou no dispositivo simplesmente atribuindo um novo valor para a propriedade **Value**.

Esta também é a propriedade padrão do objeto Elemento de Bloco OPC. Sendo assim, uma referência por valor a um objeto Elemento de Bloco OPC não necessita obrigatoriamente explicitar a propriedade **Value** para acessar o valor. Caso esta propriedade não esteja sendo atualizada, verifique se a propriedade **Index** está corretamente configurada (seu valor deve ser zero até o tamanho do Bloco OPC menos um). Exemplo:

```
Sub Botao1_Click()  
    ' Acessa um elemento e mostra o valor atual  
    ' elm1 é um objeto tipo Elemento de Bloco OPC  
    Set obj = Application.GetObject_  
        ("OPCDriver1.Grupo1.BlocoOPC1.elm1")  
    MsgBox "Valor atual de elm1: " & obj.Value  
    ' Isto também pode ser feito de outra forma,  
    ' sem mostrar a propriedade Value, que é padrão  
    MsgBox "Valor atual de elm1: " & obj  
End Sub
```

CAPÍTULO

4 Objetos em Tempo de Execução do Servidor

Esta seção contém informações sobre objetos que só podem ser utilizados em tempo de execução, o Servidor (Application) e as Pastas de Aplicativo.

4.1 Servidor

Esta seção contém informações sobre métodos específicos do objeto Server (Application). Este objeto possui os métodos gerais **Item** e **Save**, além das propriedades **Name** e **Count**, descritos na seção **Eventos, Métodos e Propriedades Gerais dos Objetos**. O objeto não possui eventos associados.

4.1.1 Métodos

Esta seção contém informações sobre os métodos específicos do objeto Servidor.

NOTA: Os métodos aqui descritos só podem ser utilizados em tempo de execução, não estando disponíveis no Studio.

4.1.1.1 ClearFailure

ClearFailure(*FailureName*)

Este método deve ser chamado para indicar que uma falha reportada por **ReportFailure** não está mais ativa. O parâmetro *FailureName* contém o nome da falha (definido pelo usuário) e deve ser passado para o método **ReportFailure**, quando de sua chamada.

4.1.1.2 ReportFailure

ReportFailure(*FailureName*, *FailureDescription*, *FailureWeight*)

Este método permite que o aplicativo reporte falhas para o E3 Server, que podem ser visualizadas clicando com o botão direito sobre o ícone do E3 na Área de Notificação. Quando o método é usado, o ícone do E3 na Área de Notificação começa a piscar em vermelho, indicando a ocorrência da falha. As falhas reportadas são normalmente condições que impedem que o aplicativo funcione parcialmente (problema de comunicação em uma porta serial, *link* de rede inativo, banco de dados sem conexão), e que normalmente não podem ser detectadas pelo E3 Server. Este método tem duas finalidades:

- Avisar o operador do sistema a respeito de problemas em um servidor.
- Auxiliar o gerenciador automático de *hot-standby* a decidir qual servidor está mais apto para executar o aplicativo.

O método **ReportFailure** possui os seguintes parâmetros:

Parâmetros do método ReportFailure

NOME	DESCRIÇÃO
FailureName	Contém o nome da falha (definido pelo usuário). Este mesmo parâmetro deve ser passado para o método ClearFailure quando o aplicativo detectar que a falha não está mais ativa. Por exemplo, "Falha COM1".
FailureDescription	Descrição textual da falha (definida pelo usuário). Por exemplo, "Falha de comunicação na COM1".
FailureWeight	Define a severidade (ou peso) da falha. O valor 0 indica um aviso (sem gravidade). Valores maiores que zero indicam falhas. Quanto maior o valor, mais grave é a falha.

Exemplo:

```

Sub TagEstadoSerial_OnValueChanged()
  If Value then
    ' O valor do tag é TRUE, indica que está em falha
    Application.ReportFailure "FALHA_COM1", "Falha de "_
      & "comunicação na COM1", 100
  Else
    ' O valor do tag é FALSE, limpa a falha
    ' (indica que está OK)
    Application.ClearFailure "FALHA_COM1"
  End If
End Sub

```

4.1.1.3 Trace

Trace(MessageText [, LogTimeStamp [, BreakLine]])

Este método permite registrar mensagens em um arquivo texto. As mensagens são gravadas em um arquivo com o mesmo nome e caminho do arquivo de Domínio, com a extensão .txt. Por exemplo, se a aplicação de Domínio for c:\teste.dom, as mensagens serão registradas no arquivo c:\teste.txt. Cada nova mensagem é adicionada sempre no final do arquivo. Caso haja falha no registro no arquivo (como por exemplo, direito de acesso negado, falta de espaço em disco, etc.), ocorrerá um erro de script.

Este método pode ser usado, por exemplo, para registrar mensagens de depuração de scripts que não sejam executados no Viewer (haja vista que neste caso não é possível utilizar o método **MsgBox**).

Parâmetros do método Trace

NOME	DESCRIÇÃO
MessageText	Mensagem de texto definida pelo usuário.
LogTimeStamp	(Opcional) Booleano que indica se cada registro deve ter a data e a hora (<i>timestamp</i>). Se omitido, assume True.
BreakLine	(Opcional) Booleano que indica a presença ou não de uma quebra de linha ao final de cada mensagem. Em False, faz com que o arquivo tenha todos os registros em uma só linha. Se omitido, assume True.

4.2 Pastas de Aplicativo

O objeto Pasta de Aplicativo é semelhante ao objeto **Pasta de Dados**, do objeto **Servidor de Dados**, permitindo agrupar objetos do Servidor dentro de pastas. Contudo, existem diferenças importantes:

- O objeto Pasta de Aplicativo só está disponível em tempo de execução.

- O objeto possui as propriedades gerais **Application**, **Count**, **Name**, **Parent** e **PathName**, além dos métodos gerais **Item** e **Save**.
- O objeto permite utilizar o comando **For Each** do VBScript para enumerar seus objetos filhos, mas este acesso leva em conta apenas os objetos do Servidor, não listando, por exemplo, telas e recursos que estejam dentro das pastas.
- A pasta raiz a partir da qual é possível enumerar os objetos é sempre o objeto **Servidor**.

Esta seção contém informações sobre eventos, métodos e propriedades dos seguintes objetos: Pasta de Dados, Tag Contador, Tag Demo, Tag Interno e Tag Timer.

5.1 Pasta de Dados

O objeto Pasta de Dados não contém eventos, métodos ou propriedades específicos, somente gerais. Estes podem ser visualizados na seção **Eventos, Métodos e Propriedades Gerais dos Objetos**.

5.2 Tag Contador

Esta seção contém informações sobre eventos e propriedades do objeto Tag Contador. Este objeto não possui métodos associados.

5.2.1 Eventos

Esta seção contém informações sobre os eventos do objeto Tag Contador.

5.2.1.1 OnPreset

OnPreset()

Ocorre toda vez que o valor da propriedade **Preset** é alcançado.

5.2.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Tag Contador.

5.2.2.1 AutoRestart

Indica que a contagem deve ser recomeçada do zero após ter sido atingido o valor estabelecido na propriedade **Preset**. Esta propriedade somente é válida quando a propriedade **CounterType** está configurada com o valor **0 - Preset**.

5.2.2.2 CounterType

 Define o comportamento do contador. Os valores disponíveis para esta propriedade são:

- **0 - Preset**: a contagem é interrompida quando o valor estabelecido na propriedade **Preset** é atingido.
- **1 - Infinite**: a contagem prossegue infinitamente.

5.2.2.3 Enabled

Inicia ou pára o contador. Em True, a contagem é iniciada; em False, ela é interrompida.

5.2.2.4 Increment

 Define o intervalo de atualização da propriedade **Value**. Caso esta propriedade tenha seu valor modificado enquanto a contagem estiver em curso, esta modificação só terá efeito quando o contador for parado e reiniciado.

5.2.2.5 Preset

 Limite a ser atingido pelo contador, em segundos. Caso o valor de **Preset** não seja um múltiplo de **Increment**, o tag irá atingir este valor antes do próximo incremento.

5.2.2.6 ResetCounterWhenEnabled

Habilita o reinício da contagem (do zero) cada vez que o valor da propriedade **Enabled** retorna a True. Quando desabilitada, o contador reinicia a contagem de onde ela foi interrompida anteriormente.

5.2.2.7 Value

9 Mostra a contagem do contador, em segundos. Propriedade somente de leitura. Essa propriedade recebe valores múltiplos de **Increment**, exceto quando a propriedade **Enabled** é ajustada para False. Nesse caso, o valor será o do momento em que o tag foi desabilitado. Ao reiniciar a contagem, o valor da propriedade será o próximo múltiplo de **Increment**. O valor máximo dessa propriedade é 2147483647 (0x7FFFFFFF).

5.3 Tag Demo

Esta seção contém informações sobre métodos e propriedades do objeto Tag Demo. Este objeto não possui eventos associados.

5.3.1 Métodos

Esta seção contém informações sobre os métodos do objeto Tag Demo.

5.3.1.1 Reset

Reset()

Zera a fase (deslocamento em tempo) da forma de onda do tag. A fase da onda somente será zerada quando o tag estiver habilitado. Este método, quando o tag estiver habilitado, não tem efeito nos tags do tipo CurrentTime e Random, que não são periódicos. Quando o tag estiver desabilitado, seu valor é simplesmente zerado, independentemente do tipo de tag.

5.3.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Tag Demo.

5.3.2.1 Enabled

■ A propriedade **Enabled** habilita ou desabilita a variação do tag Demo. Se **Enabled** estiver configurado para False, a variação estará desabilitada. Caso contrário, o tag atualizará a propriedade **Value** conforme a configuração das propriedades **Period** e **Scan**. O valor padrão desta propriedade é True.

5.3.2.2 Maximum

9 A propriedade **Maximum** determina o valor máximo do tag. O valor padrão desta propriedade é 100. Exemplo:

```
Sub CommandButton2_Click()  
    ' Ao clicar no botão, abre uma caixa de mensagens,  
    ' indicando o valor da propriedade Maximum do TagDemo6  
    MsgBox Application.GetObject("Dados.TagDemo6").Maximum  
End Sub
```

5.3.2.3 Minimum

9 A propriedade **Minimum** determina o valor mínimo do tag. O valor padrão desta propriedade é 0. Exemplo:

```
Sub CommandButton2_Click()  
    ' Ao clicar no botão, abre um messagebox indicando o valor  
    ' da propriedade Minimum do TagDemo6  
    MsgBox Application.GetObject("Dados.TagDemo6").Minimum  
End Sub
```

5.3.2.4 Period

9 Define o comprimento da forma de onda, em milissegundos. Não se aplica quando a propriedade **Type** estiver configurada para **0 - Random** ou **3 - CurrentTime**. O valor padrão desta propriedade é 10000 ms. Exemplo:

```
Sub TagDemo1_OnStartRunning()  
    Period = 1000  
End Sub
```

5.3.2.5 Scan

9 A propriedade **Scan** define o intervalo de tempo, em milissegundos, entre duas variações da propriedade **Value**. Utilize esta propriedade se você desejar maior ou menor quantidade de dados gerados pelo tag Demo. O valor padrão desta propriedade é 1000. O valor do *scan* deve ser maior que 0. Exemplo:

```
Sub Linha1_Click()  
    Application.GetObject("Dados.TagDemo2").Scan = 200  
End Sub
```

5.3.2.6 TimeStamp

 A propriedade **TimeStamp** é atualizada sempre que ocorrer alguma mudança de valor ou estado nas propriedades **Value** ou **Quality**. Ela informa qual a data/hora associada tanto ao valor quanto à qualidade do Tag Demo. Esta é uma propriedade somente para leitura. O valor padrão desta propriedade é 00:00:00.

5.3.2.7 Type

 Determina o tipo de onda do tag. Modifique a propriedade conforme a tabela a seguir. Quando a propriedade **Type** estiver configurado para 3 (**CurrentTime**), a propriedade **Value** irá conter a data/hora atual do servidor.

Opções disponíveis para Type

VALOR	FORMA DE ONDA
0	Random (aleatória)
1	Sine (onda senóide)
2	Square (onda quadrada)
3	CurrentTime (data/hora)
4	RampUp (borda de subida)
5	RampDown (borda de descida)
6	RampUpDown (borda de subida e descida)

Exemplo:

```
Sub Linha1_Click()  
    Application.GetObject("Dados.TagDemo2").Type = 2  
End Sub
```

5.3.2.8 Value

▼ A propriedade **Value** varia de acordo com o tipo da forma de onda, estabelecida na propriedade **Type**. Esta é uma propriedade apenas de leitura. O valor padrão desta propriedade é 0. Exemplo:

```
Sub Figura1_Click()  
    MsgBox Application.GetObject("Dados.TagDemo2").value = 10  
End Sub
```

5.4 Tag Interno

Esta seção contém informações sobre métodos e propriedades do objeto Tag Interno. Este objeto não possui eventos associados.

5.4.1 Métodos

Esta seção contém informações sobre os métodos do objeto Tag Interno.

5.4.1.1 WriteEx

WriteEx(*NewValue*, *NewTimestamp*, *NewQuality*)

Permite modificar o valor, a data/hora e a qualidade de um Tag Interno em uma única operação. Este método retorna um booleano que indica o sucesso ou não da operação.

O parâmetro *NewValue* especifica o novo valor do tag; se omitido, o valor do tag não é modificado. O parâmetro *NewTimestamp* especifica a nova data/hora do tag; se omitido, utiliza-se a data/hora do momento da chamada do método. O parâmetro *NewQuality* especifica a nova qualidade do tag; se omitida, será assumida a qualidade Boa (192). Todos estes parâmetros podem ser omitidos. Exemplo:

```
Sub CommandButton12_Click()
    Dim Ret
    Ret = Application.GetObject("Dados.TagInterno1")._
        WriteEx(123.456, "1/1/2001", 193)
    If Ret Then
        MsgBox "Funcionou!"
    Else
        MsgBox "Falhou!"
    End If
End Sub
```

5.4.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Tag Interno.

5.4.2.1 Quality

9 Informa a qualidade do valor contido na propriedade **Value**. Esta propriedade é de leitura e escrita, mas sempre que o valor do Tag Interno for modificado, seja por Script ou por uma associação, ela será atualizada de acordo. Exemplo:

```
Sub CommandButton1_Click()
    MsgBox Application.GetObject("Dados.TagInterno1").Quality
End Sub
```

NOTA: Para maiores informações sobre qualidade, consulte o tópico **Propriedade Quality** ou o tópico **Qualidade** no **Manual do Usuário**.

5.4.2.2 Retentive

■ A propriedade **Retentive** determina se o valor do tag interno deve ser armazenado automaticamente, caso o domínio seja parado. Se esta propriedade estiver configurada para True, o valor do tag interno será armazenado automaticamente, no caso de troca de servidor do domínio ativo. Isso garante que o valor do tag será sincronizado com um servidor *standby*. Assim, quando o servidor for executado, o valor do tag será o mesmo do servidor que parou. Caso contrário, o valor do tag será ajustado para o valor inicial sempre que o domínio for executado ou ocorrer troca do servidor ativo. Esta propriedade não tem efeito se for alterada em tempo de execução. Exemplo:

```
Sub CommandButton1_Click()  
    Dim status  
    status = Application.GetObject("Dados.TagInterno1").Retentive  
    MsgBox status  
    Select Case status  
        Case True  
            MsgBox "O valor do tag interno será _  
                armazenado automaticamente."  
        Case False  
            MsgBox "O valor do tag será ajustado para o valor _  
                inicial sempre que o domínio for executado ou _  
                ocorrer troca do servidor ativo."  
    End Select  
End Sub
```

NOTA: Esta propriedade só é válida para tags internos contidos no Servidor. Tags internos contidos no Viewer não podem ser retentivos.

5.4.2.3 TimeStamp

■ A propriedade **TimeStamp** informa a data/hora associada ao valor contido na propriedade **Value**. Esta propriedade é de leitura e escrita, mas sempre que o valor do Tag Interno for modificado, seja por script ou por uma associação, ela será atualizada de acordo.

5.4.2.4 Value

▼ A propriedade **Value** é do tipo Variant, o que permite armazenar valores de todo e qualquer tipo, desde um inteiro até referências para objetos (veja o exemplo a seguir). Utilize-o para guardar valores dentro do Viewer ou Servidor e para trocar dados entre os diversos pontos da aplicação. O valor padrão desta propriedade é vazio. Esta propriedade é de leitura e escrita. Exemplo:

```

Sub Meses_OnStartRunning()
    ' Meses é um tag do tipo InternalTag.
    ' Utiliza-se o evento para inicializar o vetor.
    Value = Array("Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho", "Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro")
End Sub

```

5.5 Tag Timer

Esta seção contém informações sobre eventos e propriedades do objeto Tag Timer. Este objeto não possui eventos associados.

5.5.1 Eventos

Esta seção contém informações sobre os eventos do objeto Tag Timer.

5.5.1.1 OnPreset

OnPreset()

Ocorre toda vez que o valor da propriedade **NextExecTime** é alcançado.

5.5.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Tag Timer.

5.5.2.1 Enabled

 Habilita ou desabilita o Timer. O valor padrão desta propriedade é True.

5.5.2.2 NextExecTime

 Exibe o próximo horário de disparo. É uma propriedade somente de leitura.

5.5.2.3 RepeatInterval

 É utilizada sempre que o tag for do tipo contínuo. O valor padrão desta propriedade é 00:00:01.

5.5.2.4 StartTime

 Data/hora inicial do Timer. Para o tipo **Single**, é a própria data/hora de disparo. Para os demais, é o momento a partir do qual o timer começa a disparar. O valor padrão é o dia e a hora locais.

5.5.2.5 TriggerType

 Tipo de disparo do timer. As opções disponíveis são:

Opções disponíveis para TriggerType

OPÇÃO	DESCRIÇÃO
0 - ttSingle	Disparo único.
1 - ttContinuous	Disparo contínuo.
2 - ttDaily	Disparo diário.
3 - ttMonthly	Disparo mensal.

CAPÍTULO

6 Telas e Objetos de Tela

Esta seção contém informações sobre eventos, métodos e propriedades de Telas e Objetos de Tela.

6.1 Tela

Esta seção contém informações sobre eventos, métodos e propriedades do objeto Tela.

6.1.1 Eventos

Esta seção contém informações sobre os eventos do objeto Tela.

6.1.1.1 Click

Click()

Ocorre quando o botão esquerdo do mouse é pressionado sobre a tela. Este evento não ocorrerá se a tela não estiver visível ou sua propriedade **Enabled** estiver configurada para False. A visibilidade da tela depende de três fatores: propriedade **Visible** igual a True; objeto-pai visível; e propriedade **Layer** do objeto presente na camada da Tela. Exemplo:

```
Sub Tela_Click()  
    ' Mostra uma caixa de mensagem quando  
    ' o usuário clica na tela  
    MsgBox "Você clicou sobre a tela."  
End Sub
```

6.1.1.2 DbClick

DbClick()

Ocorre quando há um duplo-clique, ou seja, o botão esquerdo do mouse é pressionado rapidamente duas vezes sobre a tela. Este evento não ocorrerá se o objeto não estiver visível ou sua propriedade **Enabled** estiver configurada para False. A visibilidade do objeto depende de três fatores: propriedade **Visible** igual a True;

objeto-pai visível; e propriedade **Layer** do objeto presente na camada da Tela.
Exemplo:

```
Sub Tela_DbClick()  
    ' Mostra uma caixa de mensagem quando  
    ' o usuário faz um duplo-clique na tela.  
    MsgBox "Você clicou duas vezes na tela."  
End Sub
```

6.1.1.3 KeyDown

KeyDown(KeyCode, Shift)

Ocorre no momento em que uma tecla é pressionada, independentemente do foco na Tela.

Variáveis do evento KeyDown

NOME	DESCRIÇÃO
KeyCode	Número inteiro que identifica o caractere ASCII da tecla que foi pressionada.
Shift	Mostra a tecla pressionada juntamente com o mouse: <ul style="list-style-type: none">• 4: Tecla SHIFT• 8: Tecla CTRL• 12: Teclas CTRL + SHIFT

Exemplo:

```
Sub Tela1_KeyDown(KeyCode, Shift)  
    ' Mostra uma caixa de mensagem quando  
    ' o usuário pressiona uma tecla.  
    MsgBox "Código da tecla: " & KeyCode  
End Sub
```

6.1.1.4 KeyUp

KeyUp(KeyCode, Shift)

Ocorre no momento em que uma tecla é solta, independentemente do foco na Tela.

Variáveis do evento KeyUp

NOME	DESCRIÇÃO
KeyCode	Número inteiro que identifica o caractere ASCII da tecla que foi pressionada.
Shift	Mostra a tecla pressionada juntamente com o mouse: <ul style="list-style-type: none">• 4: Tecla SHIFT• 8: Tecla CTRL• 12: Teclas CTRL +SHIFT

Exemplo:

```
Sub Tela1_KeyUp(KeyCode, Shift)
    Mostra uma caixa de mensagem quando o usuário
    soltar uma tecla
    MsgBox "Código da tecla: " & KeyCode
End Sub
```

6.1.1.5 MouseDown

MouseDown(Button, ShiftState, MouseX, MouseY)

Ocorre quando se pressiona qualquer botão do mouse sobre a Tela. Utilize o evento **MouseDown** para determinar ações específicas quando a Tela for clicada pelo usuário.

Variáveis do evento MouseDown

NOME	DESCRIÇÃO
Button	Mostra o botão do mouse pressionado: <ul style="list-style-type: none">• 1: O botão do mouse pressionado é o esquerdo.

NOME	DESCRIÇÃO
	<ul style="list-style-type: none"> • 2: O botão do mouse pressionado é o direito.
ShiftState	Mostra a tecla pressionada juntamente com o mouse: <ul style="list-style-type: none"> • 4: Tecla SHIFT • 8: Tecla CTRL • 12: Teclas CTRL + SHIFT
MouseX	Mostra a coordenada X onde o mouse foi clicado na Tela.
MouseY	Mostra a coordenada Y onde o mouse foi clicado na Tela.

Exemplo:

```
Sub TelaInicial_MouseDown(Button, ShiftState, MouseX, MouseY)
    ' Encerra a aplicação quando houver um clique de mouse
    ' no objeto TelaInicial.
    Application.Exit()
End Sub
```

6.1.1.6 MouseUp

MouseUp(*Button, ShiftState, MouseX, MouseY*)

Ocorre quando se solta qualquer botão do mouse previamente clicado. Utilize o evento **MouseUp** para especificar ações que serão disparadas somente quando o botão do mouse for solto.

Variáveis do evento MouseUp

NOME	DESCRIÇÃO
Button	Mostra o botão do mouse pressionado: <ul style="list-style-type: none"> • 1: O botão do mouse pressionado é o esquerdo. • 2: O botão do mouse pressionado é o direito.
ShiftState	Mostra a tecla pressionada juntamente com o mouse: <ul style="list-style-type: none"> • 4: Tecla SHIFT

NOME	DESCRIÇÃO
	<ul style="list-style-type: none"> • 8: Tecla CTRL • 12: Teclas CTRL + SHIFT
MouseX	Mostra a coordenada X onde o mouse foi clicado na Tela.
MouseY	Mostra a coordenada Y onde o mouse foi clicado na Tela.

Exemplo:

```
Sub TelaInicial_MouseUp(Button, ShiftState, MouseX, MouseY)
    ' Encerra a aplicação apenas quando o usuário soltar o botão.
    Application.Exit()
End Sub
```

6.1.1.7 OnHide

OnHide()

Ocorre quando uma Tela está prestes a ser fechada. Utilize o evento **OnHide** quando for necessário efetuar alguma operação antes do objeto Tela ser fechado. Este evento pode ocorrer de diversas maneiras:

- Quando a Tela for substituída por outra através do método **OpenScreen**.
- Quando o usuário fechar a janela onde está a Tela.
- Quando for chamado o método **Close** do objeto Tela.
- Quando o Viewer for fechado ou encerrado.

Exemplo:

```
Sub TelaInicial_OnHide()
    Application.Exit()
End Sub
```

6.1.1.8 OnPreShow

OnPreShow(Arg)

Ocorre antes da Tela ser mostrada. A variável de evento **ARG** recebe o conteúdo do parâmetro *Arg* do método **OpenScreen**, que gera esse evento. Logo após, é gerado o evento **OnShow**. Exemplo:

```

Sub Screen1_OnPreShow(Arg)
    ' O título da Tela Screen1 a ser mostrada
    ' foi passado como parâmetro na chamada do método
    ' OpenScreen que gerou o evento.
    Caption = Arg
End Sub

```

6.1.1.9 OnShow

OnShow()

Ocorre no momento exato que uma Tela é mostrada. Utilize o evento **OnPreShow** para efetuar alguma operação antes que esta seja exibida. Exemplo:

```

Sub TelaPrincipal_OnShow()
    MsgBox "Seja bem-vindo ao sistema!"
End Sub

```

6.1.2 Métodos

Esta seção contém informações sobre os métodos do objeto Tela.

6.1.2.1 Close

Close(*Code*)

Utilize o método **Close** para fechar a Tela. Este método gera o evento **OnHide** antes de ser efetivamente executado. O parâmetro *Code* terá o valor de retorno para o método **DoModal**, se a Tela em questão tiver sido chamada por este método. Exemplo:

```

Sub BotaoFecha_Click()
    ' Quando BotaoFecha for clicado, fecha a janela
    Screen.Close()
End Sub

```

6.1.2.2 SetFocus

SetFocus()

Utilize o método **SetFocus** para mover o foco do mouse ou do teclado para um objeto determinado. Exemplo:

```

Sub Screen1_OnShow()
    ' Ao abrir a Tela, move o foco para o Botão1
    Item("Button1").SetFocus()
End Sub

```

6.1.2.3 ToPixelX

ToPixelX()

Converte a coordenada X da Tela de Himetric para pixels.

6.1.2.4 ToPixelY

ToPixelY()

Converte a coordenada Y da Tela de Himetric para pixels.

6.1.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto Tela.

NOTA: O E3 utiliza para definição de coordenadas e espessura o sistema HIMETRIC. Neste sistema, cada unidade lógica equivale a um milésimo de centímetro; ou seja, cada 1000 unidades equivalem a 1 centímetro. Assim, este é o padrão adotado na descrição das propriedades do E3, quando aplicável.

6.1.3.1 BackgroundColor

 Especifica a cor de preenchimento do fundo da Tela. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é cinza (**RGB(192, 192, 192)**).

6.1.3.2 Caption

A A propriedade **Caption** define o título da Tela a ser mostrado na barra de título no Viewer.

6.1.3.3 FillStyle

 Esta propriedade especifica o estilo do preenchimento da Tela. A seguinte tabela contém os valores válidos para a propriedade **FillStyle**:

Opções disponíveis para FillStyle

OPÇÃO	DESCRIÇÃO
0 - bkSolid	O preenchimento será sólido (padrão).
2 - bkHorizontal	O preenchimento será com listras horizontais.
3 - bkVertical	O preenchimento será com listras verticais.
4 - bkDownward	O preenchimento será com listras descendo da esquerda para a direita em ângulo de 45 graus.
5 - bkUpward	O preenchimento será com listras subindo da esquerda para a direita em ângulo de 45 graus.
6 - bkCross	O preenchimento será com listras horizontais e verticais.
7 - bkDiagonalCross	O preenchimento será com listras descendo e subindo da esquerda para a direita em ângulo de 45 graus.
8 - bkGradient	O preenchimento será em degradê utilizando tanto ForegroundColor quanto BackgroundColor . O efeito será definido pela propriedade de GradientStyle .
11 - bkBackground	Preenche a Tela com a cor de fundo.
12 - bkPicture	Preenche a Tela com a figura selecionada na propriedade PictureFile .

6.1.3.4 ForegroundColor

 Especifica a cor do preenchimento da frente da Tela. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão para esta propriedade é preto (**RGB(0, 0, 0)**). Aplicações anteriores à introdução desta propriedade terão ambas as propriedades **ForegroundColor** e **BackgroundColor** configuradas com a cor antes armazenada na propriedade **BackgroundColor**, e o estilo de preenchimento é configurado como **11 - bkBackground**, que pinta toda a Tela com a cor de fundo (comportamento antigo, anterior à criação dos estilos). Exemplo:

```

Sub Tela_Click()
    Troca a cor de fundo da Tela para azul
    ForegroundColor = RGB(0, 0, 255)
End Sub

```

6.1.3.5 Frame

 Retorna o quadro pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

6.1.3.6 GradientStyle

 Esta propriedade especifica o estilo de preenchimento degradê da Tela. Esta propriedade só será utilizada quando o valor da propriedade **FillStyle** estiver configurada para 8 (**Gradient**). Os degradês consideram a mudança da cor partindo de **ForegroundColor** e indo para **BackgroundColor**.

Opções disponíveis para GradientStyle

OPÇÃO	DESCRIÇÃO
0 - LeftToRight	Degradê vertical da esquerda para a direita.
1 - RightToLeft	Degradê vertical da direita para a esquerda.
2 - VerFromCenter	Degradê vertical do centro para as bordas.
3 - VerToCenter	Degradê vertical das bordas para o centro.
4 - BottonUp	Degradê horizontal de baixo para cima.
5 - TopDown	Degradê horizontal de cima para baixo.
6 - HorzFromCenter	Degradê do centro para as bordas.
7 - HorzToCenter	Degradê das bordas para o centro.
8 - DiagUpRight	Degradê diagonal para cima com a cor de frente na direita (padrão).
9 - DiagUpLeft	Degradê diagonal para cima com a cor de frente na esquerda.
10 - DiagUpFromCenter	Degradê diagonal para cima do centro para as bordas.
11 - DiagUpToCenter	Degradê diagonal para cima - das bordas para o centro.

OPÇÃO	DESCRIÇÃO
12 - DiagDownLeft	Degradê diagonal para baixo com cor de frente na esquerda.
13 - DiagDownRight	Degradê diagonal para baixo com cor de frente na direita.
14 - DiagDownFromCenter	Degradê diagonal para baixo do centro para as bordas.
15 - DiagDownToCenter	Degradê diagonal para baixo das bordas para o centro.
16 - SpotSouthEast	Degradê com cor de frente partindo do canto inferior direito.
17 - SpotSouthWest	Degradê com cor de frente partindo do canto inferior esquerdo.
18 - SpotNorthWest	Degradê com cor de frente partindo do canto superior esquerdo.
19 - SpotNorthEast	Degradê com cor de frente partindo do canto superior direito.
20 - SpotFromCenter	Degradê com cor de fundo do centro para as bordas.
21 - SpotToCenter	Degradê com cor de fundo das bordas para o centro.

6.1.3.7 HasFocus

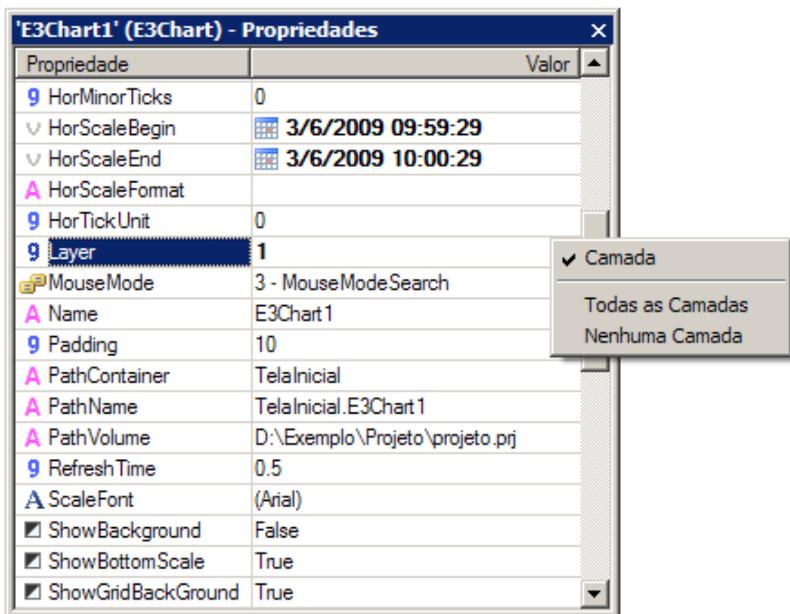
Esta propriedade determina que o objeto selecionado está com o foco. Esta propriedade é acessível apenas em tempo de execução.

6.1.3.8 Height

 A propriedade **Height** define a altura da Tela, em unidades Himetric.

6.1.3.9 Layer

9 Esta propriedade define em quais camadas o objeto deve aparecer. O valor representa uma máscara de 32 bits, um bit para cada camada. Portanto, podem ser definidas até 32 camadas individuais. Assim, objetos podem ser agrupados logicamente e mostrados ou escondidos apenas com a modificação da máscara da propriedade **Layer**.



Opções disponíveis na Lista de Propriedades

Exemplo:

```
Sub Tela1_Click()  
    Layer = 1  
End Sub
```

NOTA: A visibilidade do objeto depende de três fatores: a propriedade **Visible** tem que estar configurada para True; o objeto que é pai deste objeto tem que estar visível; a propriedade **Layer** deste objeto tem que estar habilitada para a Tela.

6.1.3.10 MouseOver

 A propriedade **MouseOver** informa se o ponteiro do mouse encontra-se em cima da tela. Caso afirmativo, a propriedade **MouseOver** é habilitada; do contrário, será False. Esta propriedade é somente de leitura e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

6.1.3.11 MouseOverChild

 A propriedade **MouseOverChild** informa se o ponteiro do mouse encontra-se em cima de um dos objetos inseridos na Tela. Caso afirmativo, a propriedade **MouseOverChild** é habilitada. Caso contrário, será False. Esta propriedade é somente para leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

6.1.3.12 PictureFile

 Contém o nome do arquivo da figura que será utilizada como fundo na Tela. Pode ser qualquer formato já suportado no E3 pelo objeto DrawPicture (*.bmp, *.gif, *.jpg, *.cur, *.ico, *.emf, *.wmf). O valor padrão desta propriedade é uma **String** em branco. Esta propriedade só será válida se a propriedade **FillStyle** estiver selecionada como **12 - bkPicture**.

6.1.3.13 PicturePosition

 Indica a posição da figura selecionada na propriedade **PictureFile** na Tela. Esta propriedade só será válida se a propriedade **FillStyle** estiver selecionada como **12 - bkPicture**. As opções válidas são:

Opções disponíveis para PicturePosition

OPÇÃO	DESCRIÇÃO
0 - Center	Figura no tamanho original, centralizada na Tela.
1 - Tile	Figura no tamanho original, repetida tantas vezes quantas forem necessárias para preencher a Tela.
2 - Stretch	Figura redimensionada para preencher a Tela.
3 - TopLeft	Figura no tamanho original, no canto esquerdo superior da Tela.

OPÇÃO	DESCRIÇÃO
4 - BottomLeft	Figura no tamanho original, no canto esquerdo inferior da Tela.
5 - BottomRight	Figura no tamanho original, no canto direito inferior da Tela.
6 - TopRight	Figura no tamanho original, no canto direito superior da Tela.

6.1.3.14 Screen

 Retorna a Tela pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

6.1.3.15 TabStop

Esta propriedade determina a utilização da tecla TAB no sistema. Se o valor do campo estiver em True, é possível utilizar a tecla; caso contrário, a tecla não pode ser utilizada.

6.1.3.16 Width

 A propriedade **Width** define a largura, em unidades Himeric, da Tela.

6.2 Objetos de Tela

Esta seção contém informações sobre eventos, métodos e propriedades dos Objetos de Tela.

6.2.1 Eventos Comuns

Esta seção contém informações sobre os eventos comuns a todos os Objetos de Tela.

6.2.1.1 Click

Click()

Ocorre quando o botão esquerdo do mouse é pressionado sobre o objeto. Este evento não ocorrerá se o objeto não estiver visível ou se sua propriedade **Enabled** estiver configurada para False. A visibilidade do objeto depende de três fatores: propriedade **Visible** igual a True; objeto-pai visível; e propriedade **Layer** do objeto presente na camada da Tela. Exemplo:

```
Sub Botao_Click()  
    ' Mostra uma caixa de mensagem quando  
    ' o usuário clica no objeto  
    MsgBox "Você clicou sobre o objeto."  
End Sub
```

6.2.1.2 DbClick

DbClick()

Ocorre quando há um duplo-clique, ou seja, o botão esquerdo do mouse é pressionado rapidamente duas vezes sobre o objeto. Este evento não ocorrerá se o objeto não estiver visível ou se sua propriedade **Enabled** estiver configurada para False. A visibilidade do objeto depende de três fatores: propriedade **Visible** igual a True; objeto-pai visível; e propriedade **Layer** do objeto presente na camada da Tela. Exemplo:

```
Sub Botao_DbClick()  
    ' Mostra uma caixa de mensagem quando  
    ' o usuário realiza um duplo-clique no objeto  
    MsgBox "Você clicou duas vezes sobre o objeto."  
End Sub
```

6.2.1.3 KeyDown

KeyDown(*KeyCode*, *Shift*)

Ocorre no momento em que uma tecla é pressionada e o objeto está com o foco do teclado. Note que este evento não será gerado se o objeto não estiver habilitado (propriedade **Enabled** igual a False) ou este objeto não possuir o foco do teclado.

Variáveis do evento KeyDown

NOME	DESCRIÇÃO
KeyCode	Número inteiro que identifica o caractere ASCII da tecla que foi pressionada.
Shift	Mostra a tecla pressionada juntamente com o mouse:

NOME	DESCRIÇÃO
	<ul style="list-style-type: none"> • 4: Tecla SHIFT • 8: Tecla CTRL • 12: Teclas CTRL + SHIFT

Exemplo:

```
Sub Botao_KeyDown(KeyCode, Shift)
    ' Mostra uma caixa de mensagem quando
    ' o usuário pressiona uma tecla
    MsgBox "Código da tecla: " & KeyCode
End Sub
```

6.2.1.4 KeyUp

KeyUp(KeyCode, Shift)

Ocorre no momento em que uma tecla é solta e o objeto está com o foco do teclado. Note que este evento não será gerado se o objeto não estiver habilitado (propriedade **Enabled** igual a False) ou este objeto não possuir o foco do teclado.

Variáveis do evento KeyUp

NOME	DESCRIÇÃO
KeyCode	Número inteiro que identifica o caractere ASCII da tecla que foi pressionada.
Shift	Mostra a tecla pressionada juntamente com o mouse: <ul style="list-style-type: none"> • 4: Tecla SHIFT • 8: Tecla CTRL • 12: Teclas CTRL + SHIFT

Exemplo:

```
Sub Botao_KeyUp(KeyCode, Shift)
    ' Mostra uma caixa de mensagem quando o usuário
    ' soltar uma tecla
    MsgBox "Código da tecla: " & KeyCode
End Sub
```

6.2.1.5 MouseDown

MouseDown(*Button, ShiftState, MouseX, MouseY*)

Ocorre quando se pressiona qualquer botão do mouse sobre o objeto.

Variáveis de evento MouseDown

NOME	DESCRIÇÃO
Button	Mostra o botão do mouse pressionado: <ul style="list-style-type: none">• 1: O botão do mouse pressionado é o esquerdo.• 2: O botão do mouse pressionado é o direito.
ShiftState	Mostra a tecla pressionada juntamente com o mouse: <ul style="list-style-type: none">• 4: Tecla SHIFT• 8: Tecla CTRL• 12: Teclas CTRL + SHIFT
MouseX	Mostra a coordenada X onde o mouse foi clicado na Tela.
MouseY	Mostra a coordenada Y onde o mouse foi clicado na Tela.

Exemplo:

```
' Encerra a aplicação quando houver um clique do mouse  
' no objeto TelaInicial.  
Sub TelaInicial_MouseDown(Button, ShiftState, MouseX, MouseY)  
Application.Exit()  
End Sub
```

6.2.1.6 MouseUp

MouseUp(*Button, ShiftState, MouseX, MouseY*)

Ocorre quando se solta qualquer botão do mouse previamente clicado sobre o objeto. Utilize o evento **MouseUp** para especificar ações que serão disparadas somente quando o botão do mouse for solto.

Variáveis do evento MouseUp

NOME	DESCRIÇÃO
Button	Mostra o botão do mouse pressionado: <ul style="list-style-type: none">• 1: O botão do mouse pressionado é o esquerdo.• 2: O botão do mouse pressionado é o direito.
ShiftState	Mostra a tecla pressionada juntamente com o mouse: <ul style="list-style-type: none">• 4: Tecla SHIFT• 8: Tecla CTRL• 12: Teclas CTRL + SHIFT
MouseX	Mostra a coordenada X onde o mouse foi clicado na Tela.
MouseY	Mostra a coordenada Y onde o mouse foi clicado na Tela.

Exemplo:

```
' Encerra a aplicação apenas quando o usuário soltar o botão  
' depois do clique sobre o objeto Retangulo1.  
Sub Retangulo1_MouseUp(Button, ShiftState, MouseX, MouseY)  
    Application.Exit()  
End Sub
```

6.2.2 Métodos Comuns

Esta seção contém informações sobre os métodos comuns a todos os Objetos de Tela.

6.2.2.1 BringToFront

BringToFront()

Faz com que o objeto se posicione à frente de todos os outros objetos da tela.

Exemplo:

```
Sub Botao1_Click()  
' Ao clicar em Botao1, o sistema  
' traz o objeto Retangulo1 para a frente  
Screen.Item("Retangulo1").BringToFront()  
End Sub
```

6.2.2.2 SendToBack

SendToBack()

Faz com que o objeto se posicione atrás de todos os outros objetos da tela. Exemplo:

```
Sub Botao2_Click()  
    ' Ao clicar em Botao2, o sistema  
    ' leva o objeto Retangulo1 para trás  
    Screen.Item("Retangulo1").SendToBack()  
End Sub
```

6.2.3 Propriedades Comuns

Neste tópicos, são listadas as propriedades comuns a todos os Objetos de Tela. As propriedades explicadas a seguir não se aplicam nos seguintes objetos: **ActiveX (MSForms)**, **E3Chart**, **E3Browser** e **E3Alarm**. Estes são citados posteriormente, em capítulos específicos para estes objetos.

NOTA 1: O E3 utiliza para definição de coordenadas e espessura o sistema HIMETRIC. Neste sistema, cada unidade lógica equivale a um milésimo de centímetro; ou seja, cada 1000 unidades equivalem a 1 centímetro. Assim, este é o padrão adotado na descrição das propriedades do E3, quando aplicável.

NOTA 2: As propriedades listadas a seguir são comuns a todos os objetos citados acima, inclusive o Grupo de objetos e movimentos de Rotação e Translação.

6.2.3.1 Enabled

Esta propriedade habilita ou desabilita o recebimento de eventos provenientes do teclado e do mouse. O valor padrão desta propriedade é True.

6.2.3.2 Frame

Retorna o quadro pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

6.2.3.3 HasFocus

☑ Esta propriedade determina se o objeto selecionado está com o foco. Esta propriedade é acessível apenas em tempo de execução.

6.2.3.4 Height

9 A propriedade **Height** define a altura do objeto, em unidades Himetric.

6.2.3.5 Layer

9 Esta propriedade define em quais camadas o objeto deve aparecer. O valor representa uma máscara de 32 bits, um bit para cada camada. Portanto, podem ser definidas até 32 camadas individuais. Assim, objetos podem ser agrupados logicamente e mostrados ou escondidos apenas com a modificação da máscara da propriedade **Layer**.

6.2.3.6 MouseOver

☑ A propriedade **MouseOver** informa se o ponteiro do mouse encontra-se em cima da Tela. Caso afirmativo, a propriedade **MouseOver** é habilitada; do contrário, será False. Esta propriedade é somente de leitura e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

6.2.3.7 MouseOverChild

☑ A propriedade **MouseOverChild** informa se o ponteiro do mouse encontra-se em cima de um dos objetos inseridos na Tela. Caso afirmativo, a propriedade **MouseOverChild** é habilitada. Caso contrário, é alterada para False. Esta propriedade é somente de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

6.2.3.8 Screen

☞ Retorna a Tela pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

6.2.3.9 TabStop

☑ Esta propriedade determina a utilização da tecla TAB no sistema. Se o campo estiver em True, é possível utilizar a tecla; caso contrário, a tecla não pode ser utilizada.

6.2.3.10 Tip

A A propriedade **Tip** mostra uma texto *popup* quando o mouse estiver por um breve momento sobre o objeto em execução. Exemplo:

```
Sub RetanguloArr1_MouseUp(nButton, nShiftState, x, y)
    Tip = "Este é um teste!"
End Sub
```

6.2.3.11 Visible

☑ Esta propriedade define se o objeto deve ou não estar visível. Se a propriedade estiver configurada para True, o objeto fica visível, desde que os seguintes itens sejam também obedecidos, obrigatoriamente: o objeto-pai deste objeto também deve estar visível, e a propriedade **Layer** do objeto deve estar presente na camada de Tela. Do contrário, o objeto fica invisível.

6.2.3.12 Width

9 A propriedade **Width** define a largura, em unidades Himetric, do objeto.

6.2.3.13 X

9 A propriedade **X** define a coordenada horizontal esquerda, em unidades Himetric, do objeto.

6.2.3.14 Y

9 Esta propriedade define a coordenada vertical superior, em unidades Himetric, do objeto.

6.2.4 Retângulo Arredondado

Esta seção contém informações sobre propriedades do objeto Retângulo Arredondado. Este objeto não possui eventos nem métodos associados.

6.2.4.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Retângulo Arredondado.

6.2.4.1.1 RoundAspectX

9 A propriedade **RoundAspectX** define o tamanho da dimensão das laterais do retângulo no eixo X. Sendo assim, de acordo com o valor definido na propriedade, os cantos do retângulo irão mudar a sua forma, desde um retângulo até uma elipse. Esta propriedade pode variar seu valor de 0.1 até 1.0. Exemplo:

```
Sub RetanguloArred_Click()  
    RoundAspectX = 0.5  
End Sub
```

6.2.4.1.2 RoundAspectY

9 A propriedade **RoundAspectY** define o tamanho da dimensão da altura do retângulo no eixo Y. Sendo assim, de acordo com o valor definido na propriedade, os cantos do retângulo irão mudar a sua forma, desde um retângulo até uma elipse. Esta propriedade pode variar seu valor de 0.1 até 1.0. Exemplo:

```
Sub RetanguloArred_Click()  
    RoundAspectY = 0.5  
End Sub
```

6.2.5 Arco de Elipse

Esta seção contém informações sobre propriedades do objeto Arco de Elipse. Este objeto não possui eventos nem métodos associados.

6.2.5.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Arco de Elipse.

6.2.5.1.1 ArcBeginAngle

9 Esta propriedade configura o ângulo inicial do arco do objeto, em graus. O intervalo aceito por essa propriedade vai de 0 até 359. O estilo e forma do arco dependerão também das configurações das propriedades **ArcEndAngle** e **ArcStyle**. O valor padrão desta propriedade é 0. Exemplo:

```
Sub CommandButton9_Click()  
    Screen.Item("Arco1").ArcBeginAngle = 12  
End Sub
```

6.2.5.1.2 ArcEndAngle

9 Esta propriedade configura o ângulo final do arco do objeto, em graus. O intervalo aceito por essa propriedade vai de 0 até 359. O estilo e forma do arco dependerão também das configurações das propriedades **ArcBeginAngle** e **ArcStyle**. O valor padrão desta propriedade é 270. Exemplo:

```
Sub CommandButton9_Click()  
    Screen.Item("Arco1").ArcEndAngle = 12  
End Sub
```

6.2.5.1.3 ArcStyle

👉 Esta propriedade especifica o estilo da borda ou linha do objeto. A borda do objeto será desenhada conforme o estilo definido, utilizando a cor especificada em **BorderColor** com a espessura **BorderWidth**. A seguinte tabela contém os valores válidos para a propriedade **ArcStyle**:

Opções disponíveis para ArcStyle

OPÇÃO	DESCRIÇÃO
0 - arc	O estilo do desenho será um arco.
1 - chord	O estilo do desenho será uma corda, unindo o ponto inicial e final.
2 - pie	O estilo do desenho será uma pizza (padrão).

Exemplo:

```
Sub CommandButton9_Click()  
    Screen.Item("Arco1").ArcStyle = 1  
End Sub
```

6.2.6 Figura

Esta seção contém informações sobre propriedades do objeto Figura. Este objeto não possui eventos nem métodos associados.

6.2.6.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Figura.

6.2.6.1.1 BackgroundColor

 Esta propriedade especifica a cor de preenchimento do fundo do objeto. Esta cor é utilizada quando a propriedade **BackgroundStyle** está configurada para 1 (opaco) e uma das propriedades **VerticalPercentFill** ou **HorizontalPercentFill** está com valores diferentes de 100. Outro uso desta cor é quando a propriedade **FillStyle** está configurada entre os valores 2 e 8. Isto faz com que a área restante utilize a cor de fundo para preenchimento. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é cinza (**RGB (192, 192, 192)**).

6.2.6.1.2 BackgroundStyle

 Esta propriedade especifica o modo do preenchimento do fundo do objeto. Esta propriedade habilita a utilização das propriedades **VerticalPercentFill** e **HorizontalPercentFill** com valores diferentes de 100 e também da propriedade **FillStyle** configurada entre os valores 2 e 8. Isto faz com que a área restante utilize a cor de fundo **BackgroundColor** para preenchimento. A seguinte tabela contém os valores válidos para a propriedade **BackgroundStyle**.

Opções disponíveis para BackgroundStyle

OPÇÃO	DESCRIÇÃO
0 - Transparente	Nenhum fundo deste objeto será desenhado.
1 - Opaco	Caso visível, o fundo será desenhado.

6.2.6.1.3 Convert

 Esta propriedade permite a conversão da figura. Se a opção do campo estiver em 0 é possível visualizar a conversão. Caso contrário, não será possível visualizar a conversão. O campo só aceita os valores 0 e 1. O valor padrão desta propriedade é 0.

6.2.6.1.4 EnableOverrideLineColor

Esta propriedade habilita ou desabilita o objeto a sobrescrever a cor original da linha da imagem pela cor definida na propriedade **OverrideLineColor**. Se a propriedade **EnableOverrideLineColor** estiver habilitada, faz com que sejam modificadas a cor original da linha do objeto pela cor de **OverrideLineColor**. Caso contrário, o objeto Figura exibirá a cor original.

6.2.6.1.5 Filename

 Define o nome do arquivo de imagem associado a esse objeto. O caminho do arquivo pode ser tanto o caminho completo do arquivo no disco quanto o caminho relativo à aplicação (quando inserido o arquivo de imagem como um recurso da aplicação). O valor padrão desta propriedade é vazio. Os seguintes tipos de arquivos de imagem são suportados:

PROPRIEDADE	FILTRO	DESCRIÇÃO	FILTRO
Bitmap file	BMP	Não	Sim
Graphics Interchange Format	GIF	Não	Sim
Joint Picture Expert Group	JPG	Não	Sim
Icon File	ICO	Não	Sim

6.2.6.1.6 HorizontalFillStyle

 Define como será o preenchimento horizontal do objeto. Esta propriedade trabalha em conjunto com a propriedade **HorizontalPercentFill**, que informa qual porcentagem do objeto deverá ser preenchida. Estas duas propriedades permitem uma simulação de um preenchimento de nível em um objeto, como ocorre em um nível de um tanque, por exemplo.

Opções disponíveis para HorizontalFillStyle

OPÇÃO	DESCRIÇÃO
0 - FillLeftToRight	A porcentagem de preenchimento será da esquerda para a direita (padrão).
1 - FillRightToLeft	A porcentagem de preenchimento será da direita para a esquerda.
2 - FillCenterToEdgesH	A porcentagem de preenchimento será do centro para as bordas.

Exemplo:

```
Sub Circulo1_OnStartRunning()  
    HorizontalFillStyle = 2  
End Sub
```

6.2.6.1.7 HorizontalPercentFill

9 Utilize a propriedade **HorizontalPercentFill** para especificar qual a porcentagem da área horizontal do objeto deverá ser preenchida. Os valores aceitos por esta propriedade variam de 0 a 100. Esta propriedade trabalha em conjunto com a propriedade **HorizontalFillStyle**, que informa como este preenchimento ocorrerá. O valor padrão desta propriedade é 100. Exemplo:

```
Sub Circulo1_OnStartRunning()  
    HorizontalPercentFill = 200  
End Sub
```

6.2.6.1.8 OverrideFillColor

 Quando a propriedade **OverrideFillMode** estiver especificada para os valores 2 ou 3, a propriedade **OverrideFillColor** será usada para definir a cor a ser usada no preenchimento da imagem, ao invés da cor original. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é vermelho (**RGB(255, 0, 0)**). Exemplo:

```
Sub DrawPicture1_Click()  
    ' Ao clicar no objeto seta o modo  
    ' Override para sólido e troca a cor de  
    ' preenchimento da imagem para azul  
    OverrideFillMode = 2  
    OverrideFillColor = RGB(0, 0, 255)  
End Sub
```

NOTA: Esta propriedade só funciona quando o objeto **Figura** está trabalhando com

arquivos metafile (WMF ou EMF).

6.2.6.1.9 OverrideFillMode

 A propriedade **OverrideFillMode** especifica o modo do preenchimento da imagem do objeto, quando este está mostrando um arquivo Windows Metafile. Ela altera o modo original de preenchimento da imagem sem alterar o arquivo definido pela propriedade **Filename**. A seguinte tabela contém os valores válidos para a propriedade **OverrideFillMode**.

Opções disponíveis para OverrideFillMode

OPÇÃO	DESCRIÇÃO
0 - NoOverride	A imagem manterá seu preenchimento original (padrão).
1 - WideFrame	A imagem não será preenchida.
2 - SolidFill	A imagem será preenchida com a cor especificada pela propriedade OverrideFillColor .
3 - ByBrightness	A imagem será preenchida com a cor especificada pela propriedade OverrideFillColor , porém levará em conta qual era a intensidade da cor original da imagem.

Exemplo:

```
Sub DrawPicture1_Click()  
    ' Ao clicar no objeto seta o modo  
    ' Override para sólido e troca a cor de  
    ' preenchimento da imagem para azul  
    OverrideFillMode = 2  
    OverrideFillColor = RGB(0, 0, 255)  
End Sub
```

NOTA: Esta propriedade só funciona quando o objeto Figura está trabalhando com arquivos metafile (WMF ou EMF).

6.2.6.1.10 OverrideLineColor

 Quando a propriedade **EnableOverrideLineColor** estiver configurada para True, a propriedade **OverrideLineColor** será usada para definir a cor a ser usada na cor da linha da figura, ao invés da cor original. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é vermelho (**RGB(255, 0, 0)**). Exemplo:

```
Sub DrawPicture1_Click()  
    OverrideLineColor = RGB(0, 0, 255)  
End Sub
```

NOTA: Esta propriedade só funciona quando o objeto Figura está trabalhando com arquivos metafile (WMF ou EMF).

6.2.6.1.11 Shadow

Indica a presença do efeito de sombra no objeto. Se True, o objeto possui uma sombra, cujas coordenadas são estabelecidas pelas propriedades **ShadowX** e **ShadowY**. Do contrário, o objeto não possui o efeito de sombra. O valor padrão desta propriedade é False.

6.2.6.1.12 ShadowColor

 Especifica a cor do preenchimento da sombra do objeto. Esta cor é utilizada quando a propriedade **Shadow** está configurada para True. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é cinza escuro (**RGB(128, 128, 128)**). Exemplo:

```
Sub Botao1_Click()  
    ' Troca a cor de fundo do botao para cinza claro  
    ' quando se clica no objeto  
    ShadowColor = RGB(192, 192, 192)  
End Sub
```

NOTA: Esta propriedade só funciona quando o objeto Figura está trabalhando com arquivos metafile (WMF ou EMF).

6.2.6.1.13 ShadowX

9 Define a coordenada vertical à esquerda da sombra deste objeto, em unidades Himetric. Esta sombra é sempre relativa à propriedade **X** do objeto. Valores positivos indicam que a sombra estará à direita do objeto, e negativos, à esquerda. O valor padrão desta propriedade é 200.

6.2.6.1.14 ShadowY

 Define a coordenada horizontal superior da sombra deste objeto, em unidades Himetric. Esta sombra é sempre relativa a propriedade **Y** do objeto. Valores positivos indicam que a sombra estará abaixo do objeto, e negativos, acima. O valor padrão desta propriedade é 200. Exemplo:

```
Sub Botao1_Click()  
    ShadowY = 250  
End Sub
```

NOTA: Esta propriedade só funciona quando o objeto Figura está trabalhando com arquivos metafile (WMF ou EMF).

6.2.6.1.15 TransparentColor

 Quando a propriedade **TransparentMode** estiver configurada para 1, esta propriedade definirá qual cor da imagem não será desenhada, e a imagem ficará transparente nestes pontos. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é branco (**RGB(255, 255, 255)**). Exemplo:

```
Sub DrawPicture1_Click()  
    ' Deixa a cor da imagem azul transparente  
    ' quando se clica no objeto Figura  
    TransparentMode = 1 ' ByColor  
    TransparentColor = RGB(0, 0, 255)  
End Sub
```

NOTA: Esta propriedade só funciona quando o objeto Figura está trabalhando com arquivos metafile (WMF ou EMF).

6.2.6.1.16 TransparentMode

 A propriedade **TransparentMode** especifica o modo como a imagem terá o efeito de transparência.

Opções disponíveis para TransparentMode

OPÇÃO	DESCRIÇÃO
0 - Disabled	Nenhuma transparência será efetuada.
1 - ByColor	A transparência será de acordo com a cor definida na propriedade TransparentColor .

OPÇÃO	DESCRIÇÃO
2 - ByPercent	A imagem ficará translúcida com porcentagem de transparência especificada na propriedade TransparentPercent .

Exemplo:

```
Sub DrawPicture1_Click()
    ' Deixa a cor da imagem azul transparente
    ' quando se clica no objeto Figura
    TransparentMode = 1 ' ByColor
    TransparentColor = RGB(0, 0, 255)
End Sub
```

NOTA: Esta propriedade só funciona quando o objeto Figura está trabalhando com arquivos metafile (WMF ou EMF).

6.2.6.1.17 TransparentPercent

9 Quando a propriedade **TransparentMode** estiver configurada para 2, esta propriedade define quão translúcida a imagem será exibida, variando de 0 (totalmente translúcido) a 100 (opaco ou sólido). Exemplo:

```
Sub DrawPicture1_Click()
    ' Deixa a imagem translúcida quando se clica no objeto
    TransparentMode = 1 ' ByPercent
    TransparentPercent = 50 ' 50% translúcido
End Sub
```

NOTA: Esta propriedade só funciona quando o objeto Figura está trabalhando com arquivos metafile (WMF ou EMF).

6.2.6.1.18 VerticalFillStyle

 Define como será o preenchimento vertical do objeto. Esta propriedade trabalha em conjunto com a propriedade **VerticalPercentFill**, que informa qual porcentagem do objeto deverá ser preenchida. Estas duas propriedades permitem uma simulação de um preenchimento de nível em um objeto.

Opções disponíveis para VerticalFillStyle

OPÇÃO	DESCRIÇÃO
0 - FillBottomToTop	A porcentagem de preenchimento será de baixo para cima.
1 - FillTopToBottom	A porcentagem de preenchimento será de cima para baixo.
2 - FillCenterToEdgesV	A porcentagem de preenchimento será do centro para as bordas.

Exemplo:

```
Sub Circulo1_OnStartRunning()  
    VerticalFillStyle = 2  
End Sub
```

6.2.6.1.19 VerticalPercentFill

9 Utilize a propriedade **VerticalPercentFill** para especificar qual a porcentagem da área vertical do objeto deverá ser preenchida. Os valores aceitos por esta propriedade variam de 0 a 100. Esta propriedade trabalha em conjunto com a propriedade **VerticalFillStyle**, que informa como este preenchimento ocorrerá. O valor padrão desta propriedade é 100. Exemplo:

```
Sub Circulo1_OnStartRunning()  
    VerticalPercentFill = 254  
End Sub
```

6.2.7 Texto, Display e SetPoint

Esta seção contém informações sobre eventos e propriedades dos objetos Texto, Display e SetPoint. Estes objetos não possuem métodos associados.

6.2.7.1 Eventos

Esta seção contém informações sobre os eventos dos objetos Texto, Display e SetPoint.

6.2.7.1.1 Validate

Validate(Cancel, NewValue)

Ocorre após os testes de limites do SetPoint (ver propriedades **MinLimit**, **MaxLimit**, **EnableLimits**) e antes do valor do SetPoint ser enviado para o tag. A finalidade deste evento é permitir que o usuário cancele o envio do valor do SetPoint para o tag.

O parâmetro *Cancel* é booleano e indica se a operação de atribuição do valor do SetPoint ao tag deve ser cancelada (*Cancel* igual a True). O padrão é False, ou seja, o valor do SetPoint será enviado ao tag. *NewValue* é o valor que está sendo avaliado. O valor antigo pode ser acessado pela propriedade **Value** do SetPoint. Exemplo:

```
Sub Texto1_validate(Cancel, NewValue)
    Mostra um MessageBox que pergunta ao usuário
    ' se ele deseja usar o novo valor digitado no SetPoint
    message = "valor atual: " & value & vbnewline & _
    "valor novo: " & NewValue & vbnewline & vbnewline & _
    "Aceita o novo valor?"
    If MsgBox (message, vbQuestion + vbYesNo, _
        Cancel = True
    End If
End Sub
```

6.2.7.2 Propriedades

Esta seção contém informações sobre as propriedades dos objetos Texto, Display e SetPoint.

6.2.7.2.1 EnableLimits

Indica se será feita a verificação de limites no texto. Quando **EnableLimits** for True, e o usuário inserir um valor não-numérico ou colocar um valor fora dos limites definidos em **MinLimit** e **MaxLimit**, uma mensagem de erro é mostrada (a propriedade **IsSetPoint** deve ser True). Exemplo:

```
Sub CommandButton1_Click()
    Screen.Item("Texto1").EnableLimits = _
    Not(Screen.Item("Texto1").EnableLimits)
End Sub
```

6.2.7.2.2 Format

A A propriedade **Format** especifica que tipo de formato será atribuído ao objeto. Permite mudar a forma como os dados são apresentados sem mudar o valor que existe por trás deles. Esta propriedade pode ser editada manualmente ou configurada através da janela de formatação. Seu uso é semelhante aos formataadores usados nas planilhas de dados, seguindo a mesma sintaxe. São suportados os seguintes tipos de dados:

Tipos de dados suportados por Format

DADOS	DESCRIÇÃO
Numérico	Saída decimal, científica, hexadecimal, binária e octal.
Texto	Textos em geral.
Booleano	Valores booleanos.
Data/Hora	Calendário Gregoriano.

6.2.7.2.3 IsSetPoint

A propriedade **IsSetPoint** é utilizada para determinar se o objeto tem comportamento de SetPoint, ou seja, permite edição da propriedade **Value**. A propriedade **Value** consiste em um **Variant**, que pode assumir valores de todo e qualquer tipo. A propriedade **IsSetPoint** é booleana e assume o seguinte comportamento: se a opção for True, será permitida a edição; já se a opção estiver False, a edição não será permitida. Isto pode ser visualizado quando o objeto Viewer está em execução. O valor padrão desta propriedade é False.

6.2.7.2.4 KeepFormatWhenEditing

 Permite que o valor do objeto seja editado com ou sem formatação. As opções disponíveis são:

- **0 - kfNever**: neste caso o valor é sempre editado sem formatação (padrão).
- **1 - kfAutomatic**: permite que o valor seja editado no valor formatado, caso o E3 detecte que o texto formatado é passível de ser interpretado como um valor.

Caso o formato seja considerado incompatível, o valor é editado sem formatação.

6.2.7.2.5 MaxLimit

9 Contém o valor máximo permitido no objeto (a propriedade **EnableLimits** deve ser True). Exemplo:

```
Sub CommandButton3_Click()  
    Screen.Item("Texto1").MaxLimit = Screen.Item("Texto6").Value  
End Sub
```

6.2.7.2.6 MinLimit

9 Contém o valor mínimo permitido no objeto (a propriedade **EnableLimits** deve ser True). Exemplo:

```
Sub CommandButton2_Click()  
    Screen.Item("Texto1").MinLimit = Screen.Item("Texto5").Value  
End Sub
```

6.2.7.2.7 Multiline

A propriedade **Multiline** indica se o texto terá múltiplas linhas (True) ou se será uma caixa de texto simples (False). Isto pode ser visualizado quando o objeto Viewer está em execução. O valor padrão desta propriedade é False. Exemplo:

```
Sub Tela1_OnStartRunning()  
    Screen.Item("TextBox1").Multiline = True  
End Sub
```

6.2.7.2.8 SetPointDataType

 Determina o tipo de valor que é enviado do SetPoint para o tag.

Opções disponíveis para SetPointDataType

OPÇÃO	DESCRIÇÃO
0 - stCurrentType	Mantém o tipo do valor atual no SetPoint (ver a seguir).
1 - stChar	Valor inteiro de 8 bits com sinal.
2 - stByte	Valor inteiro de 8 bits sem sinal.
3 - stWord	Valor inteiro de 16 bits sem sinal.
4 - stInteger	Valor inteiro de 16 bits com sinal.
5 - stLong	Valor inteiro de 32 bits com sinal.

OPÇÃO	DESCRIÇÃO
6 - stDWord	Valor inteiro de 16 bits sem sinal.
7 - stSingle	Valor de ponto flutuante de 32 bits.
8 - stDouble	Valor de ponto flutuante de 64 bits.
9 - stDateTime	Valor de data/hora.
10 - stString	Texto.

Quando o texto digitado é enviado pelo SetPoint, este primeiro tentará converter o valor para o tipo configurado (**Word, String, Double**, etc.). Se a conversão não for possível - ou seja, o valor digitado é inválido para o tipo escolhido - nenhum valor é enviado (por exemplo, se o usuário digitar -1 e o tipo for **Byte**). Já quando o valor da propriedade for **0 - stCurrentType**, o tipo do dado enviado pelo SetPoint será extraído do valor anterior presente no objeto. Caso o valor anterior seja Empty ou Null, nenhuma conversão será feita e o valor digitado será enviado como texto. Exemplo:

```
Sub Combobox1_Change()  
    Screen.Item("Texto1").SetPointDataType = CInt(Left(Value, 2))  
End Sub
```

6.2.7.2.9 StretchText

Redimensiona o tamanho do objeto. A partir do momento em que a propriedade StretchText for habilitada, o objeto automaticamente redimensiona o tamanho da fonte do texto para que a área ocupada por ele continue sempre a mesma. De outra forma, se a propriedade StretchText estiver configurada para False, nenhum redimensionamento será efetuado.

6.2.7.2.10 TextAlignment

 Esta propriedade especifica o alinhamento horizontal do texto mostrado no objeto.

Opções disponíveis para TextAlignment

OPÇÃO	DESCRIÇÃO
0 - LeftAlignment	O alinhamento horizontal do texto será à esquerda.
1 - CenterAlignment	O alinhamento horizontal do texto será centralizado.

OPÇÃO	DESCRIÇÃO
2 - RightAlignment	O alinhamento horizontal do texto será à direita.

6.2.7.2.11 TextColor

 Esta propriedade especifica a cor da fonte do texto a ser exibido. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

6.2.7.2.12 TextFont

A Define o tipo da fonte a ser utilizado pelo objeto. Esta propriedade não pode ser utilizada em scripts ou associações e é configurada somente via Studio.

6.2.7.2.13 Value

 A propriedade **Value** consiste em um **Variant**, que pode assumir valores de todo e qualquer tipo, e que será exibido pelo objeto. Geralmente esta propriedade conterà um texto, pois é automaticamente preenchida quando ocorre a criação de um novo objeto texto. A propriedade **IsSetPoint** é utilizada para determinar se o objeto texto vai ter comportamento de **IsSetPoint**, ou seja, permitir edição da propriedade **Value**. Exemplo:

```
Sub DrawString1_OnStartRunning()
    ' Lê o valor de um tag e mostra o Texto
    Dim obj
    Set obj = Application.GetObject("DataServer1.DemoTag1")
    Value = "valor de DemoTag1 = " & obj.Value
End Sub
```

6.2.7.2.14 VertTextAlignment

 Determina o alinhamento vertical do texto do objeto.

Opções disponíveis para VertTextAlignment

OPÇÃO	DESCRIÇÃO
0 - TopAlignment	Alinhamento vertical do texto no topo do objeto.
1 - MidAlignment	Alinhamento vertical do texto no centro do objeto.
2 - BottomAlignment	Alinhamento vertical do texto na base do objeto.

6.2.7.2.15 WordWrap

 Habilita ou desabilita uma quebra de linha no texto, caso a área disponível para o texto ultrapasse os limites determinados no objeto. Para que esta propriedade funcione, a propriedade **Multiline** deve ser obrigatoriamente igual a True.

6.2.8 Escala

Esta seção contém informações sobre propriedades do objeto Escala. Este objeto não possui eventos nem métodos associados.

6.2.8.1 Propriedades

Esta seção contém informações sobre propriedades do objeto Escala.

6.2.8.1.1 Angle

 A propriedade **Angle** define o ângulo de rotação em graus, no sentido anti-horário, que o objeto deve ser rotacionado. Isto também se aplica aos filhos que este objeto tiver, respeitando as limitações de rotação de cada objeto filho. O objeto rotacionará de acordo com o seu centro, que pode ser editado durante a operação de rotação. O valor padrão desta propriedade é 0 (sem rotação).

6.2.8.1.2 BackgroundColor

 Esta propriedade especifica a cor de preenchimento do fundo do objeto. Esta cor é utilizada quando a propriedade **BackgroundStyle** está configurada para 1 (opaco) e umas das propriedades **VerticalPercentFill** ou **HorizontalPercentFill** está com valores diferentes de 100. Outro uso desta cor é quando a propriedade **FillStyle** está configurada entre os valores 2 e 8. Isto faz com que a área restante utilize a cor de fundo para preenchimento. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é cinza (**RGB(192, 192, 192)**).

6.2.8.1.3 BorderColor

 Esta propriedade determina a cor da borda que será aplicada no objeto Escala. Com esta propriedade, é possível aplicar a cor padrão ou personalizá-la através da sua edição. O valor padrão desta propriedade é branco (**RGB(255, 255, 255)**). Exemplo:

```
Sub Escala1_Click()  
    BorderColor = RGB (255, 0, 0)  
End Sub
```

6.2.8.1.4 BorderStyle



A propriedade **BorderStyle** determina o estilo de borda que será aplicada ao objeto Escala.

Opções disponíveis para BorderStyle

OPÇÃO	DESCRIÇÃO
0 - Normal	Aplica uma borda sólida na grade vertical da escala (padrão).
1 - Dash	Aplica uma borda linha tracejada na escala.
2 - Dot	Aplica uma borda pontilhada na escala.
3 - Dashdot	Aplica uma borda traço e ponto na escala.
4 - Dashdotdot	Aplica uma borda traço ponto ponto na escala.
5 - Null	O objeto não terá borda.

Exemplo:

```
Sub Escala1_Click()  
    BorderStyle = 1  
End Sub
```

6.2.8.1.5 BorderWidth

9 Esta propriedade determina a largura (em pixels) da borda do objeto Escala. Através desta propriedade, é possível configurar a largura da borda, sem ter que mexer na sua estrutura. O valor padrão desta propriedade é 0. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("Escala1").Borderwidth = 120  
End Sub
```

6.2.8.1.6 FillStyle



Esta propriedade especifica o estilo do preenchimento do objeto. A seguinte tabela contém os valores válidos para a propriedade **FillStyle**:

Opções disponíveis para FillStyle

OPÇÃO	DESCRIÇÃO
0 - Solid	O preenchimento será sólido.
1 - Hollow	Não haverá preenchimento.
2 - Horizontal	O preenchimento será com listras horizontais.
3 - Vertical	O preenchimento será com listras verticais.
4 - Downward	O preenchimento será com listras descendo da esquerda para a direita em ângulo de 45 graus.
5 - Upward	O preenchimento será com listras subindo da esquerda para a direita em ângulo de 45 graus.
6 - Cross	O preenchimento será com listras horizontais e verticais.
7 - DiagonalCross	O preenchimento será com listras descendo e subindo da esquerda para a direita em ângulo de 45 graus.
8 - Gradient	O preenchimento será em degradê utilizando tanto ForegroundColor quanto BackgroundColor . O efeito será definido pela propriedade GradientStyle .
9 - SemiTransparent	Deixa o objeto translúcido.
10 - MouseArea	O preenchimento será vazio, mas o objeto continua sensível aos eventos (Padrão).

6.2.8.1.7 ForegroundColor

Esta propriedade especifica a cor do preenchimento do frente do objeto. Esta cor é utilizada quando a propriedade **FillStyle** está configurada para 0 (sólido) ou entre 2 e 9. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão para esta propriedade é azul (**RGB(0, 0, 255)**).

Exemplo:

```
Sub Botao1_Click()  
    ' Troca a cor de fundo do botao para verde  
    ' quando se clica no objeto  
    ForegroundColor = RGB(0, 255, 0)  
End Sub
```

6.2.8.1.8 Format

A propriedade **Format** especifica que tipo de formato será atribuído ao objeto. Permite mudar a forma como os dados são apresentados sem mudar o valor que existe por trás deles. Esta propriedade pode ser editada manualmente ou configurada através da janela de formatação. Seu uso é semelhante aos formatadores usados nas planilhas de dados, seguindo a mesma sintaxe. São suportados os seguintes tipos de dados:

Tipos de dados suportados por Format

DADOS	DESCRIÇÃO
Numérico	Saída decimal, científica, hexadecimal, binária e octal.
Texto	Textos em geral.
Booleano	Valores booleanos.
Data/Hora	Calendário Gregoriano.

6.2.8.1.9 GradientStyle

 Esta propriedade especifica o estilo de preenchimento degradê do objeto. Esta propriedade só será utilizada quando a valor da propriedade **FillStyle** estiver configurada para 8 (Gradient). Os degradês consideram a mudança da cor partindo do **ForegroundColor** e indo para **BackgroundColor**.

Opções disponíveis para GradientStyle

OPÇÃO	DESCRIÇÃO
0 - LeftToRight	Degradê vertical da esquerda para a direita.
1 - RightToLeft	Degradê vertical da direita para a esquerda.
2 - VerFromCenter	Degradê vertical do centro para as bordas.
3 - VerToCenter	Degradê vertical das bordas para o centro.
4 - BottomUp	Degradê horizontal de baixo para cima.
5 - TopDown	Degradê horizontal de cima para baixo.
6 - HorzFromCenter	Degradê do centro para as bordas.
7 - HorzToCenter	Degradê das bordas para o centro.
8 - DiagUpRight	Degradê diagonal para cima com a cor de frente na direita (padrão).
9 - DiagUpLeft	Degradê diagonal para cima com a cor de frente na esquerda.
10 - DiagUpFromCenter	Degradê diagonal para cima do centro para as bordas.
11 - DiagUpToCenter	Degradê diagonal para cima das bordas para o centro.
12 - DiagDownLeft	Degradê diagonal para baixo com cor de frente na esquerda.
13 - DiagDownRight	Degradê diagonal para baixo com cor de frente na direita.

OPÇÃO	DESCRIÇÃO
14 - DiagDownFromCenter	Degradê diagonal para baixo do centro para as bordas.
15 - DiagDownToCenter	Degradê diagonal para baixo das bordas para o centro.
16 - SpotSouthEast	Degradê com cor de frente partindo do canto inferior direito.
17 - SpotSouthWest	Degradê com cor de frente partindo do canto inferior esquerdo.
18 - SpotNorthWest	Degradê com cor de frente partindo do canto superior esquerdo.
19 - SpotNorthEast	Degradê com cor de frente partindo do canto superior direito.
20 - SpotFromCenter	Degradê com cor de fundo do centro para as bordas.
21 - SpotToCenter	Degradê com cor de fundo das bordas para o centro.

IMPORTANTE: Um grande número de objetos sendo mostrados com degradê simultaneamente acarreta uma perda de performance na atualização da Tela. O uso de figuras no lugar pode resolver o problema.

Exemplo:

```
Sub Botao1_Click()
    ' Objeto fica com degrade
    FillStyle = 8 ' GradientFill
    GradientStyle = 0 ' leftToRight
End Sub
```

6.2.8.1.10 LineColor

 Determina a cor da linha com as marcas divisórias e subdivisórias da escala. Para determinar a cor da legenda com números do objeto, use a propriedade **TextColor**. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

6.2.8.1.11 MaximumValue

9 Esta propriedade determina o valor máximo atingido pela escala. O valor padrão desta propriedade é 100. Exemplo:

```
Sub CommandButton1_Click()  
    ' Ao clicar no botão, abre um MessageBox indicando  
    ' o valor máximo da propriedade da escala  
    MsgBox CStr(Screen.Item("Escala1").MaximumValue)  
End Sub
```

6.2.8.1.12 MinimumValue

9 Esta propriedade determina o valor mínimo requerido pela escala. O valor padrão desta propriedade é 0. Exemplo:

```
Sub CommandButton1_Click()  
    ' Ao clicar no botão, abre um MessageBox indicando  
    ' o valor da propriedade Minimum do Escala1  
    MsgBox _  
        CStr(Application.GetObject("Dados.Escala1").MinimumValue)  
End Sub
```

6.2.8.1.13 MinorTicks

9 Esta propriedade determina a quantidade de marcas da escala. O valor padrão desta propriedade é 3. Exemplo:

```
Sub CommandButton1_Click()  
    ' Mostra o total de marcas da escala  
    MsgBox CStr(Screen.Item("Escala1").MinorTicks)  
End Sub
```

6.2.8.1.14 MinorTicksPercentSize

9 Esta propriedade determina o tamanho das marcas que subdividem cada medida do objeto Escala. O valor padrão desta propriedade é 10. Exemplo:

```
Sub CommandButton1_Click()  
    MsgBox CStr(Screen.Item("Escala1").MinorTicksPercentSize)  
End Sub
```

6.2.8.1.15 ScaleAlignment

 Esta propriedade determina o tipo de alinhamento que será atribuído à escala:

- **0 - RightSide**: à direita (valor padrão).
- **1 - LeftSide**: à esquerda.

Exemplo:

```
Sub CommandButton1_Click()
```

```
Screen.Item("Escala1").ScaleAlignment = 1
End Sub
```

6.2.8.1.16 ShowText

Esta propriedade determina a visibilidade do texto na legenda do objeto Escala. Se a opção estiver habilitada, o texto é mostrado. Caso contrário, o objeto mostra somente as linhas e subdivisões da escala. O valor padrão desta propriedade é True.

6.2.8.1.17 StretchText

Esta propriedade determina a aplicação de extensão (*stretch*) ao texto do objeto Escala (caso a largura ou altura da escala seja alterados, o texto acompanha esta variação). Se a propriedade estiver habilitada, o objeto acompanha as variações de altura e largura do objeto conforme for transformado. Caso contrário, o texto permanece com as configurações iniciais. O valor padrão desta propriedade é False.

6.2.8.1.18 TextAlignment

 A propriedade **TextAlignment** determina o alinhamento do texto no objeto.

Opções disponíveis para TextAlignment

OPÇÃO	DESCRIÇÃO
0 - leftAlignment	O alinhamento do texto será à esquerda (padrão).
1 - centerAlignment	O alinhamento do texto será centralizado.
2 - rightAlignment	O alinhamento do texto será à direita.

6.2.8.1.19 TextColor

 Determina a cor da fonte aplicada na legenda com números do objeto Escala. Para determinar a cor da linha com as marcas divisórias e subdivisórias da escala, utilize a propriedade **LineColor**. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

6.2.8.1.20 TextFont

A A propriedade **TextFont** determina a fonte que será aplicada na escala. A fonte escolhida será aplicada em todo o objeto Escala. Esta propriedade não pode ser utilizada em scripts ou associações e é configurada somente via Studio.

6.2.8.1.21 Ticks

9 Determina a quantidade de divisórias que a escala terá. O valor padrão desta propriedade é 5.

6.2.8.1.22 TickPercentSize

9 Determina o tamanho das linhas que dividem o objeto Escala. Conforme for o valor especificado nesta propriedade, maior ou menor ficará o tamanho da linha da divisória padrão da escala. O valor padrão desta propriedade é 20.

6.2.9 Animar com Translação

Esta seção contém informações sobre propriedades do objeto Animar com Translação. Este objeto não possui eventos nem métodos associados.

6.2.9.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Animar com Translação.

6.2.9.1.1 Angle

9 A propriedade **Angle** define o ângulo de rotação em graus, no sentido anti-horário, que o objeto deve ser rotacionado. Isto também se aplica aos filhos que este objeto tiver, respeitando as limitações de rotação de cada objeto filho. O objeto rotacionará de acordo com o seu centro, que pode ser editado durante a operação de rotação. O valor padrão desta propriedade é 0 (sem rotação). Exemplo:

```
Sub Arco1_Click()  
    Angle = 180  
End Sub
```

6.2.9.1.2 Detents

9 Através da propriedade **Detents** é determinado o número de passos do movimento no objeto. Exemplo:

```
Sub CommandButton1_Click()  
    MsgBox Screen.Item("DynamicRotate1").Detents  
End Sub
```

6.2.9.1.3 EnableOverrideLineColor

☑ Esta propriedade habilita ou desabilita o objeto a sobrescrever a cor original da linha da imagem pela cor definida na propriedade **OverrideLineColor**. Se a propriedade **EnableOverrideLineColor** estiver habilitada, faz com que seja modificada a cor original da linha do objeto pela cor de **OverrideLineColor**. Caso contrário, o objeto Figura exibirá a cor original. O valor padrão desta propriedade é False.

6.2.9.1.4 EnableSlider

☑ A propriedade **EnableSlider** habilita o slider no movimento do objeto. Se a propriedade estiver habilitada, o slider do movimento é habilitado. Caso contrário, isto não ocorre.

6.2.9.1.5 OverrideFillColor

🎨 Quando a propriedade **OverrideFillMode** estiver especificada para os valores 2 ou 3, a propriedade **OverrideFillColor** será usada para definir a cor a ser usada no preenchimento dos objetos no movimento, ao invés da cor original. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é vermelho (**RGB(255, 0, 0)**). Exemplo:

```
Sub DrawGroup1_Click()  
    ' Ao clicar no objeto seta o modo  
    ' Override para sólido e troca a cor de  
    ' preenchimento da imagem para azul  
    OverrideFillMode = 2  
    OverrideFillColor = RGB(0, 0, 255)  
End Sub
```

6.2.9.1.6 OverrideFillMode

📁 A propriedade **OverrideFillMode** especifica o modo do preenchimento dos objetos que estão em movimento. Ela altera o modo original de preenchimento da imagem sem alterar a configuração original de preenchimento dos objetos.

Opções disponíveis para OverrideFillMode

OPÇÃO	DESCRIÇÃO
0 - NoOverride	Preenchimento original do objeto.
1 - WireFrame	Os objetos não serão preenchidos, desenharão somente suas bordas.
2 - SolidFill	O preenchimento dos objetos contidos no grupo será sólido

OPÇÃO	DESCRIÇÃO
	com a cor especificada na propriedade OverrideFillColor .
3 - ByBrightness	O preenchimento dos objetos contidos no grupo será sólido com a cor especificada pela propriedade OverrideFillColor , porém levará em conta qual era a intensidade da cor original de preenchimento de cada objeto.

Exemplo:

```
Sub DrawGroup1_Click()
' Ao clicar no objeto seta o modo
' Override para sólido e troca a cor de
' preenchimento da imagem para azul
OverrideFillMode = 2
OverrideFillColor = RGB(0, 0, 255)
End Sub
```

6.2.9.1.7 OverrideLineColor

 Quando a propriedade **EnableOverrideLineColor** estiver configurada para True, a propriedade **OverrideLineColor** será usada para definir a cor a ser usada na cor da linha do objeto em movimento, ao invés da cor original. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é vermelho (**RGB(255, 0, 0)**). Exemplo:

```
Sub Grupo1_Click()
OverrideLineColor = RGB(255, 0, 0)
End Sub
```

6.2.9.1.8 RangeMax

 Através da propriedade **RangeMax** é determinado o alcance máximo do movimento linear ou rotacional no objeto. Exemplo:

```
Sub CommandButton1_Click()
MsgBox Screen.Item("DynamicMove1").RangeMax
End Sub
```

6.2.9.1.9 RangeMin

 Através da propriedade **RangeMin** é determinado o alcance mínimo do movimento linear ou rotacional no objeto. Exemplo:

```
Sub CommandButton1_Click()
MsgBox Screen.Item("DynamicMove1").RangeMin
End Sub
```

6.2.9.1.10 Shadow

Indica a presença do efeito de sombra no objeto. Se True, o objeto possui uma sombra, cujas coordenadas são estabelecidas pelas propriedades **ShadowX** e **ShadowY**. Do contrário, o objeto não possui o efeito de sombra. O valor padrão desta propriedade é False.

6.2.9.1.11 ShadowColor

 Especifica a cor do preenchimento da sombra do movimento. Esta cor é utilizada quando a propriedade **Shadow** está configurada para True. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é cinza escuro (**RGB(128, 128, 128)**). Exemplo:

```
Sub Botao1_Click()  
    ' Troca a cor de fundo do botao para cinza claro  
    ' quando clica no objeto  
    ShadowColor = RGB(192, 192, 192)  
End Sub
```

6.2.9.1.12 ShadowX

 Define a coordenada vertical à esquerda da sombra deste objeto, em unidades Himetric. Este sombra é sempre relativa a propriedade **X** do objeto. Valores positivos indicam que a sombra estará à direita do objeto, e negativos, à esquerda. O valor padrão desta propriedade é 200.

6.2.9.1.13 ShadowY

 Define a coordenada horizontal superior da sombra deste movimento, em unidades Himetric. Este sombra é sempre relativa a propriedade **Y** do objeto. Valores positivos indicam que a sombra estará abaixo do objeto, e negativos, acima. O valor padrão desta propriedade é 200.

6.2.9.1.14 Value

 É o valor inicial do movimento. Deve ser um valor entre os valores nas propriedades **RangeMax** e **RangeMin**.

6.2.10 Animar com Rotação

Esta seção contém informações sobre propriedades do objeto Animar com Rotação. Este objeto não possui eventos nem métodos associados.

6.2.10.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Animar com Rotação.

6.2.10.1.1 RotationAngle

 Esta propriedade determina o ângulo de rotação do movimento no objeto. Exemplo:

```
Sub CommandButton1_DbClick()  
    Screen.Item("DynamicMove1").RotationAngle = 180  
End Sub
```

6.2.10.1.2 RotationDirection

 Através da propriedade **RotationDirection** é determinada a direção do ângulo de rotação do movimento no objeto.

Opções disponíveis para RotationDirection

OPÇÃO	DESCRIÇÃO
0 - Clockwise	Direciona o ângulo de rotação a direita do objeto.
1 - CounterClockWise	Direciona o ângulo de rotação a esquerda do objeto.

Exemplo:

```
Sub CommandButton1_Click()  
    ' Direciona o ângulo de rotação à direita do objeto  
    Screen.Item("DynamicMove1").RotationDirection = 1  
End Sub
```

CAPÍTULO

7

ActiveX

Esta seção contém informações sobre eventos, métodos e propriedades de objetos ActiveX.

7.1 Métodos Comuns

Esta seção contém informações sobre os métodos comuns aos objetos ActiveX.

7.1.1 BringToFront

BringToFront()

Faz com que o objeto se posicione à frente de todos os outros objetos da tela.

Exemplo:

```
Sub Botao1_Click()  
    ' Ao clicar em Botao1, o sistema  
    ' traz o objeto Retangulo1 para a frente  
    Screen.Item("Retangulo1").BringToFront()  
End Sub
```

7.1.2 SendToBack

SendToBack()

Faz com que o objeto se posicione atrás de todos os outros objetos da tela. Exemplo:

```
Sub Botao2_Click()  
    ' Ao clicar em Botao2, o sistema  
    ' leva o objeto Retangulo1 para trás  
    Screen.Item("Retangulo1").SendToBack()  
End Sub
```

7.1.3 SetFocus

SetFocus()

Utilize o método **SetFocus** para mover o foco do mouse ou do teclado para um objeto determinado. Exemplo:

```
Sub Screen1_OnShow()  
    ' Ao abrir a tela, move o foco para o Botao1
```

```
Item("Botao1").SetFocus()  
End Sub
```

7.2 Propriedades Comuns

Esta seção contém informações sobre as propriedades comuns aos objetos ActiveX.

7.2.1 Enabled

Habilita ou desabilita o acesso via teclado ou mouse. Esta propriedade afeta apenas os objetos que permitem entrada via teclado ou mouse.

7.2.2 Frame

 Retorna o quadro-pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

7.2.3 HasFocus

Esta propriedade determina que o objeto selecionado está com o foco. Esta propriedade é acessível apenas em tempo de execução.

7.2.4 Height

 Determina a altura do objeto.

7.2.5 Layer

 Esta propriedade define em quais camadas o objeto deve aparecer. O valor representa uma máscara de 32 bits, um bit para cada camada. Portanto, podem ser definidas até 32 camadas individuais. Assim, objetos podem ser agrupados logicamente e mostrados ou escondidos apenas com a modificação da máscara da propriedade **Layer**. Para que o objeto apareça na tela, pelo menos uma das suas camadas precisa estar visível, ou seja, pelo menos 1 bit configurado na sua máscara também precisa estar configurado na propriedade **Layer** da Tela.

7.2.6 MouseOver

A propriedade **MouseOver** informa se o ponteiro do mouse encontra-se em cima da Tela. Caso afirmativo, a propriedade **MouseOver** é habilitada; do contrário, será False. Esta propriedade é apenas de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

7.2.7 MouseOverChild

☑ A propriedade **MouseOverChild** informa se o ponteiro do mouse encontra-se em cima de um dos objetos inseridos na Tela. Caso afirmativo, a propriedade **MouseOverChild** é habilitada; do contrário, será **False**. Esta propriedade é apenas de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é **False**.

7.2.8 Screen

☞ Retorna a Tela pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

7.2.9 TabStop

☑ Esta propriedade determina a utilização da tecla **TAB** no sistema. Se o campo estiver em **True**, é possível utilizar a tecla; caso contrário, a tecla não pode ser utilizada.

7.2.10 Tip

A A propriedade **Tip** mostra um texto *popup* quando o mouse estiver por um breve momento sobre o objeto em execução. Exemplo:

```
Sub RetanguloArr1_MouseUp(nButton, nShiftState, x, y)
    Tip = "Isto é um teste!"
End Sub
```

7.2.11 Visible

☑ Esta propriedade define a visibilidade do objeto. Se configurada como **True**, o objeto fica visível, desde que também sejam obedecidos os seguintes fatores: o objeto-pai deste objeto também deve estar visível, e a propriedade **Layer** deste objeto também deve estar presente na camada da Tela.

7.2.12 Width

9 Determina a largura do objeto.

7.2.13 X

9 A propriedade **X** define a coordenada horizontal esquerda, em unidades Himetric, do objeto.

7.2.14 Y

9 Esta propriedade define a coordenada vertical superior, em unidades Himetric, do objeto.

7.3 Microsoft Forms

Esta seção contém informações sobre eventos e propriedades comuns de objetos Microsoft Forms. Estes objetos não possuem métodos comuns associados.

7.3.1 Eventos Comuns

Esta seção contém informações sobre os eventos comuns aos objetos Microsoft Forms.

7.3.1.1 BeforeDragOver

BeforeDragOver(*Index, Cancel, Data, X, Y, DragState, Effect, Shift*)

Ocorre quando há uma ação de arrastar-e-soltar (drag-and-drop) sobre o objeto. Utiliza-se este evento para monitorar se o mouse entrou, deixou ou ficou sobre um objeto-alvo. O evento é disparado quando o usuário move o mouse ou pressiona ou solta algum botão do mouse. A posição do ponteiro do mouse vai indicar qual o objeto que irá gerar o evento. Pode-se determinar o status do ponteiro do mouse examinando a variável *DragState*.

Muitos objetos não suportam operações de arrastar-e-soltar enquanto a variável *Cancel* é *False*, que é o padrão. Isso significa que o objeto rejeita qualquer tentativa de arrastar ou soltar algum outro objeto sobre si e consequentemente, não dispara o evento **BeforeDropOrPaste**. A *TextBox* (caixa de texto) e a *ComboBox* são exceções; elas aceitam operações arrastar-e-soltar mesmo quando *Cancel* é *False*.

Variáveis do evento BeforeDragOver

NOME	DESCRIÇÃO
Index	Indica o índice da página, em um objeto multi-páginas, que

NOME	DESCRIÇÃO
	será afetada pela operação que gerou o evento. Para outros objetos é ignorada.
Cancel	Status do evento. O padrão é False e indica que será o objeto-alvo que tratará o evento, e não a aplicação principal.
Data	Dados sendo arrastados para o objeto-alvo, propriamente ditos.
X, Y	Posição do mouse dentro do objeto-alvo, em pontos. X é medido a partir do lado esquerdo do objeto; Y é medido a partir do topo.
DragState	<p>Indica a condição do mouse quando é gerado o evento:</p> <ul style="list-style-type: none"> • 0 - fmDragStateEnter: o mouse está dentro do alcance do objeto. • 1 - fmDragStateLeave: o mouse está fora do alcance do objeto. • 2 - fmDragStateOver: o mouse está em uma nova posição, mas ainda está dentro do alcance do objeto.
Effect	<p>Indica as ações que o objeto-alvo suporta, ou seja, o efeito de arrastar sobre o objeto em questão:</p> <ul style="list-style-type: none"> • 0 - fmDropEffectNone: objeto-alvo não aceita copiar ou mover de alguma origem. • 1 - fmDropEffectCopy: objeto-alvo permite copiar de alguma origem para si. • 2 - fmDropEffectMove: objeto-alvo permite mover de alguma origem para si. • 3 - fmDropEffectCopyOrMove: objeto-alvo permite copiar ou mover de alguma origem para si.
Shift	<p>Inteiro cuja soma dos fatores indica o status das teclas SHIFT, CTRL e ALT:</p> <ul style="list-style-type: none"> • 1: tecla SHIFT pressionada.

NOME	DESCRIÇÃO
	<ul style="list-style-type: none"> • 2: tecla CTRL pressionada. • 4: tecla ALT pressionada. <p>Por exemplo, um valor igual a 5 indica que as teclas SHIFT e ALT estavam pressionadas (1 + 4 = 5).</p>

7.3.1.2 BeforeDropOrPaste

BeforeDropOrPaste(*Index, Cancel, Ctrl, Action, Data, X, Y, Effect, Shift*)

Disparado no momento imediatamente anterior a uma operação de arrastar-e-soltar. Normalmente, isto ocorre logo após a ocorrência do evento **BeforeDragOver**.

Variáveis do evento BeforeDropOrPaste

NOME	DESCRIÇÃO
Index	Indica o índice da página, em um objeto multi-páginas, que será afetada pela operação que gerou o evento. Para outros objetos é ignorada.
Cancel	Status do evento. O padrão é False e indica que será o objeto-alvo que tratará o evento, e não a aplicação principal.
Ctrl	Objeto-alvo.
Data	Dados sendo arrastados para o objeto-alvo, propriamente ditos.
Action	Indica o resultado, baseado nas configurações do teclado, de uma operação arrastar-e-soltar pendente: <ul style="list-style-type: none"> • 2 - fmActionPaste: cola o objeto selecionado no objeto-alvo. • 3 - fmActionDragDrop: indica que o usuário arrastou o objeto selecionado de sua origem e soltou-o sobre o objeto-alvo.
X, Y	Posição do mouse dentro do objeto-alvo, em pontos. <i>X</i> é medido a partir do lado esquerdo do objeto; <i>Y</i> é medido a partir do topo.
DragState	Indica a condição do mouse quando é gerado o evento:

NOME	DESCRIÇÃO
	<ul style="list-style-type: none"> • 0 - fmDragStateEnter: o mouse está dentro do alcance do objeto. • 1 - fmDragStateLeave: o mouse está fora do alcance do objeto. • 2 - fmDragStateOver: o mouse está em uma nova posição, mas ainda está dentro do alcance do objeto.
Effect	<p>Indica as ações que o objeto-alvo suporta, ou seja, o efeito de arrastar sobre o objeto em questão:</p> <ul style="list-style-type: none"> • 0 - fmDropEffectNone: objeto-alvo não aceita copiar ou mover de alguma origem. • 1 - fmDropEffectCopy: objeto-alvo permite copiar de alguma origem para si. • 2 - fmDropEffectMove: objeto-alvo permite mover de alguma origem para si. • 3 - fmDropEffectCopyOrMove: objeto-alvo permite copiar ou mover de alguma origem para si
Shift	<p>Inteiro cuja a soma dos fatores indica o status das teclas SHIFT, CTRL e ALT:</p> <ul style="list-style-type: none"> • 1: tecla SHIFT pressionada. • 2: tecla CTRL pressionada. • 4: tecla ALT pressionada. <p>Por exemplo, um valor igual a 5 indica que as teclas SHIFT e ALT estavam pressionadas (1 + 4 = 5).</p>

7.3.1.3 Change

Change()

Ocorre quando o valor da propriedade **Value** do objeto é modificada. A seguir estão alguns exemplos de ações que disparam o evento **Change**:

- Clicar em uma caixa de seleção, em um botão de opções ou botão incremento-decremento.
- Clicar ou selecionar palavras em uma lista de seleção ou editor de texto.
- Selecionar diferentes abas em um diálogo.
- Mover a barra de rolagem em um objeto barra de rolagem.
- Clicar nas setas de um botão incremento-decremento.
- Selecionar diferentes páginas em um objeto multi-páginas.

7.3.1.4 OnError

OnError(Number, Description, SCode, Source, HelpFile, HelpContext, CancelDisplay)

Gerado por um erro interno no objeto. Se este evento não for tratado, o E3 irá mostrar uma mensagem de erro genérica.

Variáveis do evento OnError

NOME	DESCRIÇÃO
Number	Número inteiro identificador do erro.
Description	String com a descrição do erro.
SCode	Inteiro com o código de erro do subsistema OLE (não utilizado).
Source	String com o objeto que originou o erro.
HelpFile	String com o nome e o caminho do arquivo de ajuda.
HelpContext	Número de contexto do tópico de ajuda referente ao erro (inteiro).

NOME	DESCRIÇÃO
CancelDisplay	Booleano. Indica se o erro deve ser mostrado em uma MessageBox.

7.3.1.5 KeyPress

KeyPress(*KeyAscii*)

Ocorre quando o objeto tem o foco de teclado e o usuário pressiona uma tecla que corresponde a um caractere que pode ser mostrado em tela (uma tecla ANSI, de código indicado na variável *KeyAscii*). Ou seja, o evento ocorre quando alguma das teclas a seguir são pressionadas:

- Qualquer caractere do teclado que possa ser impresso.
- Tecla CTRL combinada com qualquer caractere do alfabeto padrão.
- Tecla CTRL combinada com qualquer caractere especial.
- Tecla BACKSPACE.
- Tecla ESC.

Este evento não ocorre nas seguintes condições:

- Pressionando a tecla TAB.
- Pressionando a tecla ENTER.
- Pressionando a tecla DEL (essa tecla não é ANSI).
- Pressionando as setas do teclado.
- Quando uma tecla faz com que o foco vá de um objeto a outro.

Enquanto um usuário pressiona uma tecla que produz um código ANSI, o objeto recebe os eventos **KeyDown** e **KeyPress** repetidamente. Quando o usuário libera a tecla, o evento **KeyUp** ocorre. Para monitorar o estado físico do teclado ou manipular teclas não reconhecidas pelo evento **KeyPress** (como teclas de função, navegação, etc.), utiliza-se os eventos **KeyDown** e **KeyUp**.

7.3.2 Propriedades Comuns

Esta seção contém informações sobre as propriedades comuns aos objetos Microsoft Forms.

NOTA: O E3 utiliza para definição de coordenadas e espessura o sistema HIMETRIC. Neste sistema, cada unidade lógica equivale a um milésimo de centímetro; ou seja, cada 1000 unidades equivalem a 1 centímetro. Assim, este é o padrão adotado na descrição das propriedades do E3, quando aplicável.

7.3.2.1 BackColor

 Determina a cor de fundo do objeto. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade para os objetos Lista de Seleção, Lista e Editor de Texto é a cor branca (**RGB(255, 255, 255)**), e para os demais objetos é a cor bege (**RGB(236, 233, 216)**).

7.3.2.2 Enabled

Habilita ou desabilita o objeto, isto é, o foco e a resposta à eventos gerados pelo usuário. Se a propriedade estiver configurada para True, o objeto pode receber o foco, responder a eventos gerados pelo usuário e estará acessível através de scripts (padrão). Caso contrário, o usuário não poderá interagir com o objeto usando o mouse, pressionando teclas ou teclas de atalho e o objeto aparece esmaecido. Além disso, se o objeto exibir um bitmap, o bitmap fica esmaecido sempre que este objeto estiver desabilitado.

As propriedades **Enabled** e **Locked** são interligadas. Quando as propriedades **Enabled** e **Locked** forem ambas True, o objeto pode receber foco e aparecer normalmente na Tela, e os dados podem ser copiados, mas não editados. Quando **Enabled** for True, mas **Locked** for False, os dados podem ser não só copiados, mas também editados. No entanto, quando **Enabled** for False, o objeto não recebe foco e fica esmaecido na Tela, independentemente do status de **Locked**. Além disso, os dados não podem ser copiados nem tampouco editados.

É possível combinar as configurações das propriedades **Enabled** e **TabStop** para impedir que o usuário selecione um botão de comando com a tecla TAB, embora ainda seja permitido ao usuário clicar no botão. Definir a propriedade **TabStop** como False significa que o botão de comando não aparecerá na ordem de tabulação. Entretanto, se **Enabled** for True, o usuário ainda poderá clicar no botão de comando, desde que a propriedade **TakeFocusOnClick** esteja definida como True.

7.3.2.3 ForeColor

 Especifica a cor do preenchimento do fundo do objeto. Nos scripts, utilize a função do VBScript **RGB** para formar a cor a ser associada a esta propriedade. O valor padrão desta propriedade para todos os objetos MSForms é a cor preta (**RGB(0, 0, 0)**).

7.3.2.4 MouseIcon

 A propriedade **MouseIcon** atribui uma imagem ao ponteiro do mouse, quando este é movido sobre o objeto. Esta propriedade é válida somente quando a propriedade **MousePointer** é definida como **99 - fmMousePointerCustom**.

Um arquivo com imagem pode ser selecionado para o ponteiro do mouse de duas formas: através da Lista de Propriedades (extensões .cur ou .ico); ou via scripts, utilizando-se a função **LoadPicture** para especificar o caminho e o nome do arquivo que contém o ícone personalizado (somente extensão .cur). Exemplo:

```
Sub CommandButton1_Click()  
    ' Atribuindo o ítem 99 - fmMousePointerCustom à propriedade  
    ' para que ela aceite a customização do ícone do mouse  
    Screen.Item("CheckBox1").MousePointer = 99  
    Screen.Item("CheckBox1").MouseIcon = LoadPicture("C:\a.cur")  
End Sub
```

7.3.2.5 MousePointer

 A propriedade **MousePointer** especifica o tipo de ponteiro do mouse que é mostrado quando o usuário posiciona o mesmo sobre um objeto. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para MousePointer

OPÇÃO	DESCRIÇÃO
0 - fmMousePointerDefault	Ponteiro padrão. A imagem é determinada pelo objeto.
1 - fmMousePointerArrow	Seta.
2 - fmMousePointerCross	Ponteiro em forma de cruz.
3 - fmMousePointerBeam	Em forma de I.
6 - fmMousePointerSizeNesw	Seta dupla apontando para nordeste e sudoeste.

OPÇÃO	DESCRIÇÃO
7 - fmMousePointerSizeNS	Seta dupla apontando para norte e sul.
8 - fmMousePointerNWse	Seta dupla apontando para noroeste e sudeste.
9 - fmMousePointerWE	Seta dupla apontando para oeste e leste.
10 - fmMousePointerUpArrow	Seta para cima.
11 - MousePointerStarHourGlassring	Ampulheta.
12 - fmMousePointerHelpNoDrop	Símbolo de Not (círculo com uma linha diagonal) na parte superior do objeto sendo arrastado. Indica um destino de soltar inválido.
13 - fmMousePointerAppStarting	Seta com uma ampulheta.
14 - fmMousePointerHelp	Seta com um ponto de interrogação.
15 - fmMousePointerSizeAll	Dimensiona todo o cursor (setas apontando para norte, sul, leste e oeste).
99 - fmMousePointerCustom	Usa o ícone especificado pela propriedade MouseIcon .

Use a propriedade **MousePointer** para indicar alterações de funcionalidade conforme o ponteiro do mouse passar sobre os objetos em uma Tela. Por exemplo, a configuração de ampulheta (opção 11) é útil para indicar que o usuário precisa esperar algum tempo um processo ou operação ser finalizado. Alguns ícones podem variar, dependendo das configurações do sistema, como os ícones associados aos temas da área de trabalho. O valor padrão desta propriedade é **0 - fmMousePointerDefault**.

7.3.3 Caixa de Seleção e Botão de Opção

Esta seção contém informações sobre propriedades do objeto Caixa de Seleção e do Botão de Opção. Estes objetos não possuem eventos nem métodos associados.

7.3.3.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Caixa de Seleção e do Botão de Opção.

7.3.3.1.1 Accelerator

A Define ou recupera a tecla aceleradora do objeto. Esta tecla aceleradora é uma tecla de atalho que, usada em conjunto com a tecla ALT, dá o foco ao objeto. O valor padrão desta propriedade é vazio.

7.3.3.1.2 Alignment

 A propriedade **Alignment** especifica a posição do objeto relativa a sua legenda. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmAlignmentLeft:** coloca a legenda à esquerda do objeto.
- **1 - fmAlignmentRight:** coloca a legenda à direita do objeto.

7.3.3.1.3 AutoSize

A propriedade **AutoSize** ajusta a largura do texto, caso a área disponível para o mesmo ultrapasse o tamanho do objeto. Quando a propriedade estiver configurada para True, o texto será redimensionado para coincidir com o tamanho atual do objeto.

7.3.3.1.4 BackStyle

 A propriedade **BackStyle** define o estilo do plano de fundo para os objetos. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmBackStyleTransparent:** define o objeto como transparente, isto é, nenhum fundo deste objeto será desenhado.
- **1 - fmBackStyleOpaque:** define o objeto como opaco, isto é, o fundo será desenhado (valor padrão).

NOTA: Esta propriedade não afeta a transparência de bitmaps. Deve-se usar um editor de imagens como o Paintbrush, por exemplo, para tornar um bitmap transparente. Nem todos os objetos ActiveX suportam bitmaps transparentes.

7.3.3.1.5 Caption

A Define o texto que será mostrado no objeto.

7.3.3.1.6 Font

A A propriedade **Font** é utilizada para determinar a fonte do objeto. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.3.3.1.7 GroupName

A A propriedade **GroupName** é utilizado para criar um grupo de objetos mutuamente exclusivos.

NOTA: Esta propriedade não é utilizada no E3 e foi mantida por motivos de compatibilidade com a especificação padrão dos objetos Microsoft Forms.

7.3.3.1.8 Locked

A propriedade **Locked** habilita ou desabilita a edição do objeto. Se esta propriedade estiver configurada para True, a edição não é permitida. Caso contrário, é possível editar o objeto. Os valores configurados na propriedade **Enabled** influenciam o comportamento de **Locked**. Para maiores detalhes, consulte a propriedade **Enabled**. O valor padrão desta propriedade é False.

7.3.3.1.9 Picture

 A propriedade **Picture** especifica a figura (bitmap) atribuída ao objeto. Um arquivo com imagem pode ser selecionado de duas maneiras: através da Lista de Propriedades ou via scripts, utilizando-se a função **LoadPicture** para especificar o caminho e o nome do arquivo que contém a figura. Para remover a figura, clique no valor da propriedade **Picture** e pressione a tecla DEL. A tecla BACKSPACE (tecla de espaço do teclado) não remove a figura. Exemplo:

```
Sub CommandButton1_Click()
```

```
Screen.Item("CheckBox1").Picture = LoadPicture("c:\aba.gif")
End Sub
```

7.3.3.1.10 PicturePosition

 A propriedade **PicturePosition** especifica a posição da figura atribuída ao objeto em relação a sua legenda. As opções disponíveis para esta propriedades são as seguintes:

Opções disponíveis para PicturePosition

OPÇÃO	DESCRIÇÃO
0 - fmPicturePositionLeftTop	A figura aparece à esquerda de sua legenda. A legenda é alinhada com a parte superior da figura.
1 - fmPicturePositionLeftCenter	A figura aparece à esquerda da legenda. A legenda é centralizada relativamente à imagem.
2 - fmPicturePositionLeftBotton	A figura aparece à esquerda da legenda. A legenda é alinhada com a parte inferior da figura.
3 - fmPicturePositionRightTop	A figura aparece à direita da legenda. A legenda é alinhada com a parte superior da figura.
4 - fmPicturePositionRightCenter	A figura aparece à direita da legenda. A legenda é centralizada relativamente à figura.
5 - fmPicturePositionRightBotton	A figura aparece à direita da legenda. A legenda é alinhada com a parte inferior da figura.
6 - fmPicturePositionAboveLeft	A figura aparece acima da legenda. A legenda é alinhada com a extremidade esquerda da figura.
7 - fmPicturePositionAboveCenter	A figura aparece acima da legenda. A legenda é centralizada abaixo da figura (padrão).
8 - fmPicturePositionAboveRight	A figura aparece acima da legenda. A legenda é alinhada com a extremidade

OPÇÃO	DESCRIÇÃO
	direita da figura.
9 - fmPicturePositionBelowLeft	A figura aparece abaixo da legenda. A legenda é alinhada com a extremidade esquerda da figura.
10 - fmPicturePositionBelowCenter	A figura aparece abaixo da legenda. A legenda é centralizada acima da figura.
11 - fmPicturePositionBelowRight	A figura aparece abaixo da legenda. A legenda é alinhada com a extremidade direita da figura.

7.3.3.1.11 SpecialEffect

 A propriedade **SpecialEffect** especifica a aparência de um objeto. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para SpecialEffect

OPÇÃO	DESCRIÇÃO
0 - fmButtonEffectFlat	O objeto aparece plano e possui um destaque na borda, uma alteração de cor ou ambos.
2 - fmButtonEffectSunken	O objeto possui uma sombra na parte superior esquerda e um destaque na parte inferior direita, como se estivesse afundado na Tela.

7.3.3.1.12 TextAlign

 Especifica como o texto é alinhado no objeto. As opções disponíveis são as seguintes:

- **1 - fmTextAlignLeft:** alinha o texto com a extremidade esquerda do objeto.
- **2 - fmTextAlignCenter:** centraliza o texto com a extremidade direita do objeto.
- **3 - fmTextAlignRight:** alinha o texto com a extremidade direita do objeto.

7.3.3.1.13 TripleState

☑ A propriedade **TripleState** determina até três estados de valores para o objeto. Se a propriedade estiver configurada para True, o usuário poderá escolher entre três opções de estados: Falso, verdadeiro ou Nulo. O valor Nulo é exibido como um botão sombreado. Caso contrário, o usuário pode escolher entre os valores False ou True. O valor padrão desta propriedade é False.

7.3.3.1.14 Value

▼ Indica o valor inicial do objeto. Tem comportamento booleano; se True, o objeto inicia marcado, do contrário, seu estado inicial é não marcado. O valor padrão desta propriedade é False.

7.3.3.1.15 WordWrap

☑ Habilita ou desabilita uma quebra de linha no texto, caso a área disponível para o texto ultrapasse os limites determinados no objeto. Para que esta propriedade funcione, a propriedade **Multiline** deve ser obrigatoriamente igual a True.

7.3.4 Lista de Seleção

Esta seção contém informações sobre eventos, métodos e propriedades do objeto Lista de Seleção.

7.3.4.1 Eventos

Esta seção contém informações sobre os eventos do objeto Lista de Seleção.

7.3.4.1.1 DropButtonClick

DropButtonClick()

Ocorre quando a lista de opções aparece ou desaparece ao se clicar no objeto.

7.3.4.2 Métodos

Esta seção contém informações sobre os métodos do objeto Lista de Seleção.

7.3.4.2.1 AddItem

AddItem([pvargItem], [pvargIndex])

O método **AddItem** é utilizado para se adicionar itens a uma Lista de Seleção. *pvargItem* é um string que contém o texto que será adicionado na lista; se for omitido, será acrescentado um string em branco. *pvargIndex* é o índice do texto na lista; se for omitido, *pvargItem* é adicionado como último item da lista. Exemplo:

```
Sub CommandButton1_Click()  
    EntryCount = EntryCount + 1  
    ComboBox1.AddItem(EntryCount & " - Selection")  
End Sub
```

7.3.4.2.2 Clear

Clear()

Limpa o texto do objeto. Exemplo:

```
Sub ClearTextButton_Click()  
    ComboBox1.Clear()  
End Sub
```

7.3.4.2.3 Copy

Copy()

Copia o texto previamente selecionado para a área de transferência. Utilize o método **Paste** para colar o texto no local indicado. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("ComboBox1").Copy()  
End Sub
```

7.3.4.2.4 Cut

Cut()

Recorta o texto previamente selecionado para a área de transferência. Utilize o método **Paste** para colar o texto no local indicado. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("ComboBox1").Cut()  
End Sub
```

7.3.4.2.5 DropDown

DropDown()

O método **DropDown** abre a lista de itens de uma Lista de Seleção. A chamada deste método tem o mesmo efeito que clicar, em tempo de execução, no botão com a seta ao lado direito do objeto. Exemplo:

```
Sub CommandButton1_Click()
    Dim ComboBox1
    ComboBox1.AddItem "Abacaxi"
    ComboBox1.AddItem "Morango"
    ComboBox1.AddItem "Uva"
    ComboBox1.AddItem "Laranja"
    ComboBox1.DropDown()
End Sub
```

7.3.4.2.6 Paste

Paste()

Insera no objeto o conteúdo da área de transferência. Exemplo:

```
Sub CommandButton1_Click()
    Screen.Item("ComboBox1").Paste()
End Sub
```

7.3.4.2.7 RemoveItem

RemoveItem(*pvargIndex*)

Remove itens de uma Lista ou de uma Lista de Seleção. Este método possui o parâmetro *pvargIndex*, que especifica a linha que será excluída, iniciando em zero. Ou seja, o primeiro elemento é 0, o segundo é 1, e assim por diante. Exemplo:

```
Sub CommandButton2_Click()
    ComboBox1.SetFocus
    ' Verifica se a lista contém dados selecionados
    If ComboBox1.ListCount >= 1 Then
        ' Se não há nenhuma seleção, escolher o último dado da lista.
        If ComboBox1.ListIndex = -1 Then
            ComboBox1.ListIndex = ComboBox1.ListCount - 1
        End If
        ComboBox1.RemoveItem(ComboBox1.ListIndex)
    End If
End Sub
```

7.3.4.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto Lista de Seleção.

7.3.4.3.1 AutoSize

☑ A propriedade **AutoSize** ajusta a largura do texto, caso a área disponível para o mesmo ultrapasse o tamanho do objeto. Para o objeto Lista de Seleção (Combo Box), quando a propriedade estiver configurada para True, a largura do texto é reajustada para a mesma do objeto.

7.3.4.3.2 AutoTab

☑ A propriedade **AutoTab** habilita ou desabilita a tabulação automática no objeto. Se a propriedade estiver configurada para True, a tabulação automática ocorre. Caso contrário, não é utilizada.

Após o usuário digitar o número máximo de caracteres em um objeto (utilizando a propriedade **MaxLength**), o foco se move automaticamente para o próximo objeto da ordem de tabulação, quando estes caracteres forem atingidos. Por exemplo, para que uma Lista de Seleção exiba dados de estoque que possuam sempre cinco caracteres, pode-se usar a propriedade **MaxLength** para especificar o número máximo de caracteres a serem introduzidos no objeto e a propriedade **AutoTab** para tabular automaticamente para o próximo objeto depois que o usuário digitar cinco caracteres.

7.3.4.3.3 AutoWordSelect

☑ Habilita ou desabilita a seleção automática de palavras no objeto. Se esta propriedade estiver configurada para True, a palavra indicada é selecionada no texto mais o espaço seguinte, caso se tenha selecionado parte dela. Caso contrário, somente o caractere indicado na palavra é selecionado.

7.3.4.3.4 BackStyle

👉 A propriedade **BackStyle** define o estilo do plano de fundo para os objetos. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmBackStyleTransparent:** define o objeto como transparente, isto é, nenhum fundo deste objeto será desenhado.
- **1 - fmBackStyleOpaque:** define o objeto como opaco, isto é, o fundo será desenhado (valor padrão).

NOTA: Esta propriedade não afeta a transparência de bitmaps. Deve-se usar um

editor de imagens como o Paintbrush, por exemplo, para tornar um bitmap transparente. Nem todos os objetos ActiveX suportam bitmaps transparentes.

7.3.4.3.5 BorderColor

 Esta propriedade determina a cor da borda que será aplicada no objeto. Com esta propriedade, é possível aplicar a cor padrão ou personalizá-la através da sua edição. Para que esta propriedade seja aplicável, é necessário que a propriedade **BorderStyle** esteja configurada para **1 - fmBorderStyleSingle**. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.3.4.3.6 BorderStyle

 A propriedade **BorderStyle** determina o estilo de borda que será aplicada ao objeto. As opções disponíveis são:

- **0 - fmBorderStyleNone**: sem borda.
- **1 - fmBorderStyleSingle**: com borda simples.

7.3.4.3.7 BoundColumn

 Determina a coluna da lista onde os dados serão armazenados. Por exemplo, se cada linha contém 8 itens e a propriedade **BoundColumn** é 3, o sistema armazena as informações na terceira coluna da linha atualmente selecionada. Se o valor for igual a 0, este valor é passado para a propriedade **ListIndex** do objeto. Se o valor for 1 ou superior, o dado indicado é atribuído à coluna referente ao valor especificado na propriedade. As colunas são numeradas a partir de 1.

NOTA: Esta propriedade não tem efeito no E3 e foi mantida por motivos de compatibilidade com a especificação padrão dos objetos Microsoft Forms.

7.3.4.3.8 CanPaste

 A propriedade **CanPaste** especifica se a área de transferência contém os dados aos quais o objeto dá suporte. Se a opção estiver configurada para True, o objeto pode receber informações coladas da área de transferência. Se os dados da área de transferência estiverem em um formato ao qual o objeto não dá suporte, a propriedade **CanPaste** será False. Por exemplo, ao tentar colar um bitmap em um objeto que só dê suporte a texto, **CanPaste** será False. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.9 Column

▼ Especifica a linha e a coluna no objeto. Se for especificado somente o valor de coluna, a propriedade **Column** lê ou escreve a coluna especificada na linha atual do objeto. Por exemplo, `MinhaListaDeSelecao.Column(3)` lê ou escreve a terceira coluna do objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.10 ColumnCount

9 A propriedade **ColumnCount** especifica o número de colunas do objeto. A configuração de **ColumnCount** como 0 não exibe nenhuma coluna, e a configuração desta propriedade como -1 exibe todas as colunas disponíveis. O valor padrão desta propriedade é 1.

7.3.4.3.11 ColumnHeads

☑ A propriedade **ColumnHeads** habilita ou desabilita a exibição do título das colunas no objeto. Se a propriedade estiver configurada para True, o título é mostrado. Caso contrário, o título das colunas não é mostrado. O valor padrão é False.

7.3.4.3.12 ColumnWidths

📏 A propriedade **ColumnWidths** é utilizada para especificar a largura da coluna do objeto, em pontos. Um valor igual a -1 ou em branco faz com que a largura seja calculada na coluna (a largura mínima de uma coluna calculada é 72 pontos, ou 1 polegada). Um valor igual a 0 oculta a coluna. Para produzir colunas mais estreitas, deve-se especificar a largura na propriedade ou utilizar alguma dos valores a seguir:

Opções disponíveis para ColumnWidths

OPÇÃO	DESCRIÇÃO
90;72;90	A primeira coluna possui 90 pontos (1,25 polegada); a segunda coluna possui 72 pontos (1 polegada); a terceira coluna possui 90 pontos.
6 cm;0;6 cm	A primeira coluna possui 6 centímetros; a segunda coluna está oculta; a terceira coluna possui 6 centímetros. Como parte da terceira coluna é visível, uma barra de rolagem horizontal aparece.
1,5 pol;0;2,5 pol	A primeira coluna possui 1,5 polegadas, a segunda coluna

OPÇÃO	DESCRIÇÃO
	está oculta, e a terceira coluna possui 2,5 polegadas.
2 pol;;2 pol	A primeira coluna possui 2 polegadas, a segunda coluna possui 1 polegada (padrão) e a terceira coluna possui 2 polegadas. Como somente metade da terceira coluna é visível, uma barra de rolagem horizontal aparece.
(Vazio)	Todas as três colunas possuem a mesma largura (1,33 polegadas). O valor padrão desta propriedade é vazio (o E3 irá utilizar o valor padrão do sistema).

7.3.4.3.13 CurTargetX

9 Retorna a posição horizontal de inserção de um texto no objeto. Esta posição é medida em unidades himétricas (um himetro é igual a 0,0001 de um metro). Pode-se usar **CurTargetX** e **CurX** para mover o ponto de inserção de um texto conforme o usuário percorre o conteúdo do objeto. Quando o usuário move o ponto de inserção para outra linha do texto, a propriedade **CurTargetX** especifica a posição mais indicada para o ponto de inserção do texto desejado. A propriedade **CurX** é definida neste valor, se a linha do texto for maior que o valor de **CurTargetX**. Caso contrário, a propriedade **CurX** é definida como o final da linha do texto. Esta propriedade é acessível apenas em tempo de execução.

NOTA: Esta propriedade não tem efeito no E3 e foi mantida por motivos de compatibilidade com a especificação padrão dos objetos Microsoft Forms.

7.3.4.3.14 CurX

9 A propriedade **CurX** especifica a posição atual horizontal do ponto de inserção no objeto. Esta propriedades é aplicada num objeto que possua várias linhas, isto é, a propriedade **Multiline** está habilitada. O valor de retorno é válido quando o objeto possui o foco. Pode-se usar a propriedade **Multiline** e a propriedade **CurX** para posicionar o ponto de inserção do texto conforme o usuário usa a barra de rolagem pelo conteúdo do objeto. Quando o usuário move o ponto de inserção para uma outra linha de texto rolando o conteúdo do objeto, a propriedade **CurTargetX** especifica a posição desejada para o ponto de inserção. A propriedade **CurX** é definida com este valor se a linha de texto for maior do que o valor de **CurTargetX**. Caso contrário, **CurX** é definido no fim da linha de texto. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.15 DragBehavior



Habilita ou desabilita o recurso de arrastar e soltar um texto no conteúdo do objeto. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmDragBehaviorDisabled:** não permite a ação de arrastar e soltar o texto no conteúdo do objeto.
- **1 - fmDragBehaviorEnabled:** permite a ação de arrastar e soltar o texto no conteúdo do objeto.

O valor padrão desta propriedade é **0 - fmDragBehaviorDisabled**.

NOTA: A propriedade **DragBehavior** não tem efeito se a propriedade **Style** estiver configurada para **2 - fmStyleDropDownList**.

7.3.4.3.16 DropButtonStyle



Esta propriedade especifica o símbolo exibido no botão da Lista de Seleção. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para DropButtonStyle

OPÇÃO	DESCRIÇÃO
0 - fmDropButtonStylePlain	Exibe um botão plano, sem símbolo.
1 - fmDropButtonStyleArrow	Exibe um seta para baixo (padrão).
2 - fmDropButtonStyleEllipsis	Exibe reticências.
3 - fmDropButtonStyleReduce	Exibe uma reta horizontal como um caractere sublinhado.

O valor padrão desta propriedade é **1 - fmDropButtonStyleArrow**.

7.3.4.3.17 EnterFieldBehavior

 Esta propriedade controla a forma como o conteúdo do texto é selecionado na área de edição quando se pressiona a tecla TAB no objeto, e não quando o objeto recebe o foco como um resultado do método **SetFocus**. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmEnterFieldBehaviorSelectAll**: seleciona todo o conteúdo do texto quando a tecla TAB é pressionada no objeto.
- **1 - fmEnterFieldBehaviorRecallSelection**: deixa a seleção inalterada.

O valor padrão desta propriedade é **0 - fmEnterFieldBehaviorSelectAll**.

7.3.4.3.18 Font

A A propriedade **Font** é utilizada para determinar a fonte do objeto. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.3.4.3.19 HideSelection

A propriedade **HideSelection** especifica se o texto selecionado continua em destaque quando um objeto não tiver mais o foco. Se a propriedade estiver configurada para True, o texto selecionado somente se mantém em destaque caso o objeto tenha o foco. Caso contrário, o texto selecionado sempre aparece em destaque, independente do foco no objeto. O valor padrão desta propriedade é True.

7.3.4.3.20 IMEMode

 A propriedade **IMEMode** especifica o modo IME (*Input Method Editor*) de um objeto.

NOTA: Essa propriedade aplica-se apenas em aplicativos escritos em idiomas asiáticos (chinês simplificado, chinês tradicional, coreano e japonês) e é ignorada em outros aplicativos. Foi mantida por motivos de compatibilidade com a especificação padrão dos objetos Microsoft Forms.

As opções disponíveis são as seguintes:

Opções disponíveis para IMEMode

OPÇÃO	DESCRIÇÃO
0 - fmIMEModeNoControl	Não controla IME (padrão).
1 - fmIMEModeOn	IME ativado.
2 - fmIMEModeOff	IME desativado. Modo inglês.
3 - fmIMEModeDisable	IME desativado. O usuário não pode ativar IME pelo teclado.
4 - fmIMEModeHiragana	IME ativado com modo Hiragana de largura total.
5 - fmIMEModeKatakanaFull	IME ativado com modo Katakana de largura total.
6 - fmIMEModeKatakana	IME ativado com modo Katakana de meia largura.
7 - fmIMEModeAlphaFull	IME ativado com modo Alfanumérico de largura total.
8 - fmIMEModeAlpha	IME ativado com modo Alfanumérico de meia largura.
9 - fmIMEModeHangulFull	IME ativado com modo Hangul de largura total.
10 - fmIMEModeHangul	IME ativado com modo Hangul de meia largura.
11 - fmIMEModeHanziFull	IME ativado com modo Hanzi de largura total.
12 - fmIMEModeHanzi	IME ativado com modo Hanzi de meia largura.

7.3.4.3.21 LineCount

9 A propriedade **LineCount** retorna o número de linhas do objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.22 List

▼ Retorna ou define as entradas da linha e da coluna na lista do objeto. A numeração de linha e de coluna começa com zero. Isto é, o número de linha da primeira linha na lista é zero; o número de coluna da primeira coluna é zero. O número da segunda linha ou coluna é 1 e assim por diante. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.23 ListCount

9 Retorna o número de itens contidos na lista do objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.24 ListIndex

9 Identifica o item atualmente selecionado na lista, chamado de índice. Os valores de **ListIndex** vão de -1 ao número total de linhas em uma lista menos um (isto é, **ListCount** $- 1$). Quando nenhuma linha é selecionada, **ListIndex** retorna -1 . Quando o usuário seleciona uma linha em uma Lista de seleção, o sistema define o valor da propriedade **ListIndex**. O valor da propriedade **ListIndex** da primeira linha de uma lista é 0, o valor da segunda linha é 1 e assim por diante. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.25 ListRows

9 A propriedade **ListRows** determina o número máximo de linhas na lista do objeto. O valor padrão desta propriedade é 8.

7.3.4.3.26 ListStyle

📄 A propriedade **ListStyle** determina o estilo da lista do objeto. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmListStylePlain**: listagem com os itens do plano de fundo em destaque.
- **1 - fmListStyleOption**: exibe botões de opção ou caixas de seleção para uma lista de várias opções. Quando o usuário seleciona um item do grupo, o botão de opção associado ao item é selecionado e os botões de opção para outros itens no grupo são desmarcados.

O valor padrão desta propriedade é **0 - fmListStylePlain**.

7.3.4.3.27 ListWidth

 A propriedade **ListWidth** determina a largura da lista do objeto. O valor padrão desta propriedade é 0.

7.3.4.3.28 Locked

 A propriedade **Locked** habilita ou desabilita a edição do objeto. Se esta propriedade estiver configurada para True, a edição não é permitida. Caso contrário, é possível editar o objeto. Os valores configurados na propriedade **Enabled** influenciam o comportamento de **Locked**. Para maiores detalhes, consulte a propriedade **Enabled**. O valor padrão desta propriedade é False.

7.3.4.3.29 MatchEntry

 Procura, através do texto digitado pelo usuário, por uma entrada de texto que coincida com os dados existentes no objeto. Ao encontrar uma incidência do texto, a linha contendo a mesma é selecionada e o conteúdo da coluna é exibido.

As opções disponíveis são as seguintes:

- **0 - fmMatchEntryFirstLetter**: procura pela entrada de texto que coincida com o primeiro caractere digitado no objeto. Se for digitada repetidamente a mesma letra, é passada para a próxima entrada de texto que comece com esta letra e assim por diante.
- **1 - fmMatchEntryComplete**: conforme cada caractere é digitado, o objeto procura por uma entrada de texto que coincida com os caracteres digitados.
- **2 - fmMatchEntryNone**: não faz a procura no objeto.

O valor padrão desta propriedade é **1 - fmMatchEntryComplete**.

7.3.4.3.30 MatchFound

 Indica se o texto que o usuário digitou no objeto coincide com qualquer entrada na lista. Se a propriedade estiver configurada para True, o conteúdo da propriedade **Value** coincide com um dos registros na lista. Caso contrário, o conteúdo da propriedade **Value** não coincide com nenhum dos registros na lista (padrão).

Esta propriedade é acessível apenas em tempo de execução, não sendo aplicável quando a propriedade **MatchEntry** é definida como 2. O valor padrão desta propriedade é False.

7.3.4.3.31 MatchRequired

☑ Especifica se o texto digitado deve ou não coincidir com os itens existente na Lista de Seleção. Se a propriedade estiver configurada para True, o usuário não pode sair da Lista de Seleção até que o texto inserido coincida com um item existente do objeto. Caso contrário, o texto inserido na Lista de Seleção pode ser diferente de todos os dados existente da mesma.

7.3.4.3.32 MaxLength

9 A propriedade **MaxLength** determina o número máximo de caracteres no objeto. Configurando esta propriedade como 0, não há limite de caractere no objeto.

7.3.4.3.33 SelectionMargin

☑ Habilita ou desabilita a margem de seleção do objeto. Se esta propriedade estiver configurada para True, o texto será selecionado ao se clicar na margem do objeto. Caso contrário, o texto não será selecionado ao se clicar na margem.

NOTA: Se a propriedade **SelectionMargin** estiver configurada para True quando o objeto for impresso, a margem de seleção também será impressa.

7.3.4.3.34 SelLength

9 Retorna o número de caracteres selecionados no objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.35 SelStart

9 Indica o ponto inicial do texto selecionado ou o ponto de inserção se nenhum texto for selecionado. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.36 SelText

A Retorna o texto selecionado no objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.37 ShowDropButtonWhen

 A propriedade **ShowDropButtonWhen** especifica quando mostrar o botão suspenso (tecla de navegação do objeto). As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmShowDropButtonWhenNever:** não mostra o botão suspenso em nenhuma circunstância.
- **1 - fmShowDropButtonWhenFocus:** mostra o botão suspenso quando o objeto possui o foco.
- **2 - fmShowDropButtonWhenAlways:** sempre mostra o botão suspenso.

7.3.4.3.38 SpecialEffect

 A propriedade **SpecialEffect** especifica a aparência de um objeto. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para SpecialEffect

OPÇÃO	DESCRIÇÃO
0 - fmSpecialEffectFlat	O objeto aparece plano e possui um destaque na borda, uma alteração de cor ou ambos.
1 - fmSpecialEffectRaised	O objeto possui um destaque na parte superior esquerda e uma sombra na parte inferior direita, como se estivesse em relevo.
2 - fmSpecialEffectSunken	O objeto possui uma sombra na parte superior esquerda e um destaque na parte inferior direita. O objeto e a sua borda aparecem afundados na Tela.
3 - fmSpecialEffectEtched	A borda parece estar esculpida ao redor da extremidade do objeto.
6 - fmSpecialEffectBump	O objeto possui um ressalto na parte inferior direita e parece plano na parte superior esquerda..

7.3.4.3.39 Style

 Determina o estilo do objeto. As opções disponíveis são as seguintes:

- **0 - fmStyleDropDownCombo:** a lista de seleção se comporta como uma caixa de combinação suspensa. O usuário pode digitar um valor na região de edição ou selecionar um valor da lista suspensa (padrão).
- **2 - fmStyleDropDownList:** a Lista de seleção se comporta como uma caixa de listagem. O usuário deve escolher um valor da lista.

7.3.4.3.40 Text

A Retorna o texto da opção selecionada. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.41 TextAlign

 Especifica como o texto é alinhado no objeto. As opções disponíveis são as seguintes:

- **1 - fmTextAlignLeft:** alinha o texto com a extremidade esquerda do objeto.
- **2 - fmTextAlignCenter:** centraliza o texto com o centro do objeto.
- **3 - fmTextAlignRight:** alinha o texto com a extremidade direita do objeto.

7.3.4.3.42 TextColumn

V A propriedade **TextColumn** identifica a coluna no objeto. Os valores para a propriedade **TextColumn** vão de -1 ao número de colunas na lista. O valor **TextColumn** para a primeira coluna é 1, o valor da segunda coluna é 2, e assim por diante. A configuração de **TextColumn** como 0 exibe os valores para a propriedade **ListIndex**. A configuração de **TextColumn** como -1 exibe a primeira coluna que possui o valor da propriedade **ColumnWidths** maior do que 0.

7.3.4.3.43 TextLength

9 Retorna o número de caracteres digitados no objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.4.3.44 TopIndex

9 A propriedade **TopIndex** define ou retorna o item da lista que aparece na posição mais próxima do topo na lista. Esta propriedade retorna o valor -1 se a lista está vazia ou não é exibida.

7.3.4.3.45 Value

√ É o valor da propriedade **BoundColumn** das linhas atualmente selecionadas. Uma mudança de valor na propriedade **Value** não implica em mudança de valor na propriedade **BoundColumn**. Para adicionar ou apagar entradas em uma lista de seleção, pode-se usar os métodos **AddItem** ou **RemoveItem**.

7.3.5 Botão de Comando

Esta seção contém informações sobre eventos e propriedades do objeto Botão de Comando. Este objeto não possui métodos associados.

7.3.5.1 Eventos

Esta seção contém informações sobre os eventos do objeto Botão de Comando.

7.3.5.1.1 MouseMove

MouseMove()

Ocorre quando o ponteiro do mouse é movimentado sobre o botão de comando.

7.3.5.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Botão de Comando.

7.3.5.2.1 Accelerator

A Define ou recupera a tecla aceleradora do objeto. Esta tecla aceleradora é uma tecla de atalho que usada em conjunto com a tecla ALT, dá o foco ao objeto. O valor padrão desta propriedade é vazio.

7.3.5.2.2 AutoSize

A propriedade **AutoSize** ajusta a largura do texto, caso a área disponível para o mesmo ultrapasse o tamanho do objeto. Para o objeto Botão de Comando (Command Button), quando a propriedade estiver configurada para True, a largura do texto é reajustada para a mesma do objeto. O conteúdo do texto é cortado quando ele excede a área do objeto.

7.3.5.2.3 BackStyle

 A propriedade **BackStyle** define o estilo do plano de fundo para os objetos. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmBackStyleTransparent:** define o objeto como transparente, isto é, nenhum fundo deste objeto será desenhado.
- **1 - fmBackStyleOpaque:** define o objeto como opaco, isto é, o fundo será desenhado (valor padrão).

NOTA: Esta propriedade não afeta a transparência de bitmaps. Deve-se usar um editor de imagens como o Paintbrush, por exemplo, para tornar um bitmap transparente. Nem todos os objetos ActiveX suportam bitmaps transparentes.

7.3.5.2.4 Caption

A Define o texto que será mostrado no objeto.

7.3.5.2.5 Font

A A propriedade **Font** é utilizada para determinar a fonte do objeto. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.3.5.2.6 Locked

 A propriedade **Locked** habilita ou desabilita a edição do objeto. Se esta propriedade estiver configurada para True, a edição não é permitida. Caso contrário, é possível editar o objeto. Os valores configurados na propriedade **Enabled** influenciam o comportamento de **Locked**. Para maiores detalhes, consulte a propriedade **Enabled**. O valor padrão desta propriedade é False.

7.3.5.2.7 Picture

 A propriedade **Picture** especifica a figura (bitmap) atribuída ao objeto. Um arquivo com imagem pode ser selecionado de duas maneiras: através da Lista de Propriedades ou via scripts, utilizando-se a função **LoadPicture** para especificar o caminho e o nome do arquivo que contém a figura. Para remover a figura, clique no valor da propriedade **Picture** e pressione a tecla DEL. A tecla BACKSPACE (tecla de espaço do teclado) não remove a figura.

7.3.5.2.8 PicturePosition

 A propriedade **PicturePosition** especifica a posição da figura atribuída ao objeto em relação a sua legenda. As opções disponíveis para esta propriedades são as seguintes:

Opções disponíveis para PicturePosition

OPÇÃO	DESCRIÇÃO
0 - fmPicturePositionLeftTop	A figura aparece à esquerda de sua legenda. A legenda é alinhada com a parte superior da figura.
1 - fmPicturePositionLeftCenter	A figura aparece à esquerda da legenda. A legenda é centralizada relativamente à imagem.
2 - fmPicturePositionLeftBottom	A figura aparece à esquerda da legenda. A legenda é alinhada com a parte inferior da figura.
3 - fmPicturePositionRightTop	A figura aparece à direita da legenda. A legenda é alinhada com a parte superior da figura.

OPÇÃO	DESCRIÇÃO
4 - fmPicturePositionRightCenter	A figura aparece à direita da legenda. A legenda é centralizada relativamente à figura.
5 - fmPicturePositionRightBottom	A figura aparece à direita da legenda. A legenda é alinhada com a parte inferior da figura.
6 - fmPicturePositionAboveLeft	A figura aparece acima da legenda. A legenda é alinhada com a extremidade esquerda da figura.
7 - fmPicturePositionAboveCenter	A figura aparece acima da legenda. A legenda é centralizada abaixo da figura (padrão).
8 - fmPicturePositionAboveRight	A figura aparece acima da legenda. A legenda é alinhada com a extremidade direita da figura.
9 - fmPicturePositionBelowLeft	A figura aparece abaixo da legenda. A legenda é alinhada com a extremidade esquerda da figura.
10 - fmPicturePositionBelowCenter	A figura aparece abaixo da legenda. A legenda é centralizada acima da figura.
11 - fmPicturePositionBelowRight	A figura aparece abaixo da legenda. A legenda é alinhada com a extremidade direita da figura.

7.3.5.2.9 TakeFocusOnClick

Especifica se o objeto recebe o foco quando clicado. Se a propriedade estiver configurada para True, o objeto recebe o foco quando clicado. Caso contrário, o objeto não recebe o foco quando clicado.

7.3.5.2.10 WordWrap

Habilita ou desabilita uma quebra de linha no texto, caso a área disponível para o texto ultrapasse os limites determinados no objeto. Para que esta propriedade funcione, a propriedade **Multiline** deve ser obrigatoriamente igual a True.

7.3.6 Texto

Esta seção contém informações sobre propriedades do objeto Texto. Este objeto não possui eventos nem métodos associados.

7.3.6.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Texto.

7.3.6.1.1 Accelerator

A Define ou recupera a tecla aceleradora do objeto. Esta tecla aceleradora é uma tecla de atalho que usada em conjunto com a tecla ALT, dá o foco ao objeto. O valor padrão desta propriedade é vazio.

7.3.6.1.2 AutoSize

A propriedade **AutoSize** ajusta a largura do texto, caso a área disponível para o mesmo ultrapasse o tamanho do objeto. Para o objeto Texto (Label), quando a propriedade estiver configurada para True, o texto é redimensionado para coincidir com o tamanho atual do objeto, permitindo assim a sua exibição completa.

7.3.6.1.3 BackStyle

 A propriedade **BackStyle** define o estilo do plano de fundo para os objetos. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmBackStyleTransparent:** define o objeto como transparente, isto é, nenhum fundo deste objeto será desenhado.
- **1 - fmBackStyleOpaque (valor padrão):** define o objeto como opaco, isto é, o fundo será desenhado.

NOTA: Esta propriedade não afeta a transparência de bitmaps. Deve-se usar um editor de imagens como o Paintbrush, por exemplo, para tornar um bitmap transparente. Nem todos os objetos ActiveX suportam bitmaps transparentes.

7.3.6.1.4 BorderColor

 Esta propriedade determina a cor da borda que será aplicada no objeto. Com esta propriedade, é possível aplicar a cor padrão ou personalizá-la através da sua edição. Para que esta propriedade seja aplicável, é necessário que a propriedade **BorderStyle** esteja configurada para **1 - fmBorderStyleSingle**. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.3.6.1.5 BorderStyle

 A propriedade **BorderStyle** determina o estilo de borda que será aplicada ao objeto. As opções disponíveis são:

- **0 - fmBorderStyleNone:** sem borda.
- **1 - fmBorderStyleSingle:** com borda simples.

7.3.6.1.6 Caption

A Define o texto que será mostrado no objeto.

7.3.6.1.7 Font

A A propriedade **Font** é utilizada para determinar a fonte do objeto. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.3.6.1.8 Picture

 A propriedade **Picture** especifica a figura (bitmap) atribuída ao objeto. Um arquivo com imagem pode ser selecionado de duas maneiras: através da Lista de Propriedades ou via scripts, utilizando-se a função **LoadPicture** para especificar o caminho e o nome do arquivo que contém a figura. Para remover a figura, clique o valor da propriedade **Picture** e pressione a tecla DEL. A tecla BACKSPACE (tecla de espaço do teclado) não remove a figura. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("CheckBox1").Picture = LoadPicture("C:\aba.gif")  
End Sub
```

7.3.6.1.9 PicturePosition



A propriedade **PicturePosition** especifica a posição da figura atribuída ao objeto em relação a sua legenda. As opções disponíveis para esta propriedades são as seguintes:

Opções disponíveis para PicturePosition

OPÇÃO	DESCRIÇÃO
0 - fmPicturePositionLeftTop	A figura aparece à esquerda de sua legenda. A legenda é alinhada com a parte superior da figura.
1 - fmPicturePositionLeftCenter	A figura aparece à esquerda da legenda. A legenda é centralizada relativamente à imagem.
2 - fmPicturePositionLeftBottom	A figura aparece à esquerda da legenda. A legenda é alinhada com a parte inferior da figura.
3 - fmPicturePositionRightTop	A figura aparece à direita da legenda. A legenda é alinhada com a parte superior da figura.
4 - fmPicturePositionRightCenter	A figura aparece à direita da legenda. A legenda é centralizada relativamente à figura.
5 - fmPicturePositionRightBottom	A figura aparece à direita da legenda. A legenda é alinhada com a parte inferior da figura.
6 - fmPicturePositionAboveLeft	A figura aparece acima da legenda. A legenda é alinhada com a extremidade esquerda da figura.
7 - fmPicturePositionAboveCenter	A figura aparece acima da legenda. A legenda é centralizada abaixo da figura (padrão).
8 - fmPicturePositionAboveRight	A figura aparece acima da legenda. A legenda é alinhada com a extremidade direita da figura.

OPÇÃO	DESCRIÇÃO
9 - fmPicturePositionBelowLeft	A figura aparece abaixo da legenda. A legenda é alinhada com a extremidade esquerda da figura.
10 - fmPicturePositionBelowCenter	A figura aparece abaixo da legenda. A legenda é centralizada acima da figura.
11 - fmPicturePositionBelowRight	A figura aparece abaixo da legenda. A legenda é alinhada com a extremidade direita da figura.

7.3.6.1.10 SpecialEffect

 A propriedade **SpecialEffect** especifica a aparência de um objeto. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para SpecialEffect

OPÇÃO	DESCRIÇÃO
0 - fmSpecialEffectFlat	O objeto aparece plano e possui um destaque na borda, uma alteração de cor ou ambos.
1 - fmSpecialEffectRaised	O objeto possui um destaque na parte superior esquerda e uma sombra na parte inferior direita. O objeto parece estar em relevo.
2 - fmSpecialEffectSunken	O objeto possui uma sombra na parte superior esquerda e um destaque na parte inferior direita. O objeto e a sua borda parecem estar afundados na Tela.
3 - fmSpecialEffectEtched	A borda parece estar esculpida ao redor da extremidade do objeto.
6 - fmSpecialEffectBump	O objeto possui um resalto na parte inferior direita e parece plano na parte superior esquerda.

7.3.6.1.11 TextAlign

 Especifica como o texto é alinhado no objeto. As opções disponíveis são as seguintes:

- **1 - fmTextAlignLeft:** alinha o texto com a extremidade esquerda do objeto.
- **2 - fmTextAlignCenter:** centraliza o texto com a extremidade direita do objeto.
- **3 - fmTextAlignRight:** alinha o texto com a extremidade direita do objeto.

7.3.6.1.12 WordWrap

Habilita ou desabilita uma quebra de linha no texto, caso a área disponível para o texto ultrapasse os limites determinados no objeto. Para que esta propriedade funcione, a propriedade **Multiline** deve ser obrigatoriamente igual a True.

7.3.7 Lista

Esta seção contém informações sobre métodos e propriedades do objeto Lista. Este objeto não possui eventos associados.

7.3.7.1 Métodos

Esta seção contém informações sobre os métodos do objeto Lista.

7.3.7.1.1 AddItem

AddItem([pvargItem], [pvargIndex])

O método **AddItem** é utilizado para se adicionar itens a uma Lista. *pvargItem* é um string que contém o texto que será adicionado à lista; se for omitido, será acrescentado um string em branco. *pvargIndex* é o índice do texto na lista; se for omitido, *pvargItem* é adicionado como último item da lista. Exemplo:

```
Sub CommandButton1_Click()  
    EntryCount = EntryCount + 1  
    ListBox1.AddItem(EntryCount & " - Selection")  
End Sub
```

7.3.7.1.2 Clear

Clear()

Limpa o texto do objeto. Exemplo:

```
Sub ClearTextButton_Click()  
    ListBox1.Clear()  
End Sub
```

7.3.7.1.3 RemoveItem

RemoveItem(*pvargIndex*)

Remove itens da Lista. Este método possui o parâmetro *pvargIndex*, que especifica a linha que será excluída, iniciando em zero. Ou seja, o primeiro elemento é 0, o segundo é 1, e assim por diante. Exemplo:

```
Sub CommandButton2_Click()  
    List1.SetFocus  
    ' Verifica se a lista contém dados selecionados  
    If List1.ListCount >= 1 Then  
        ' Se não há nenhuma seleção, escolher o último dado da lista.  
        If List1.ListIndex = -1 Then  
            List1.ListIndex = List1.ListCount - 1  
        End If  
        List1.RemoveItem(List1.ListIndex)  
    End If  
End Sub
```

7.3.7.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Lista.

7.3.7.2.1 BorderColor

 Esta propriedade determina a cor da borda que será aplicada no objeto. Com esta propriedade, é possível aplicar a cor padrão ou personalizá-la através da sua edição. Para que esta propriedade seja aplicável, é necessário que a propriedade **BorderStyle** esteja configurada para **1 - fmBorderStyleSingle**. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.3.7.2.2 BorderStyle

 A propriedade **BorderStyle** determina o estilo de borda que será aplicada ao objeto. As opções disponíveis são:

- **0 - fmBorderStyleNone**: sem borda.
- **1 - fmBorderStyleSingle**: com borda simples.

7.3.7.2.3 BoundColumn

▼ Determina a coluna da lista onde os dados serão armazenados. Por exemplo, se cada linha contém 8 itens e a propriedade **BoundColumn** é 3, o sistema armazena as informações na terceira coluna da linha atualmente selecionada. Se o valor for igual a 0, este valor é passado para a propriedade **ListIndex** do objeto. Se o valor for 1 ou superior, o dado indicado é atribuído à coluna referente ao valor especificado na propriedade. As colunas são numeradas a partir de 1.

NOTA: Esta propriedade não tem efeito no E3 e foi mantida por motivos de compatibilidade com a especificação padrão dos objetos Microsoft Forms.

7.3.7.2.4 Column

▼ Especifica a linha e coluna no objeto. Se for especificado somente o valor de coluna, a propriedade **Column** lê ou escreve a coluna especificada na linha atual do objeto. Por exemplo, `MinhaListaDeSeleção.Column(3)` lê ou escreve a terceira coluna do objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.7.2.5 ColumnCount

 A propriedade **ColumnCount** especifica o número de colunas do objeto. A configuração de **ColumnCount** como 0 não exibe nenhuma coluna, e a configuração desta propriedade como -1 exibe todas as colunas disponíveis. O valor padrão desta propriedade é 1.

7.3.7.2.6 ColumnHeads

A propriedade **ColumnHeads** habilita ou desabilita a exibição do título das colunas no objeto. Se a propriedade estiver configurada para True, o título é mostrado. Caso contrário, o título das colunas não é mostrado. O valor padrão é False.

7.3.7.2.7 ColumnWidths

9 A propriedade **ColumnWidths** é utilizada para especificar a largura da coluna do objeto, em pontos. Um valor igual a -1 ou em branco faz com que a largura seja calculada na coluna (a largura mínima de uma coluna calculada é 72 pontos, ou 1 polegada). Um valor igual a 0 oculta a coluna. Para produzir colunas mais estreitas, deve-se especificar a largura na propriedade ou utilizar alguma dos valores a seguir:

Opções disponíveis para ColumnWidths

OPÇÃO	DESCRIÇÃO
90;72;90	A primeira coluna possui 90 pontos (1,25 polegada); a segunda coluna possui 72 pontos (1 polegada); a terceira coluna possui 90 pontos.
6 cm;0;6 cm	A primeira coluna possui 6 centímetros; a segunda coluna está oculta; a terceira coluna possui 6 centímetros. Como parte da terceira coluna é visível, uma barra de rolagem horizontal aparece.
1,5 pol;0;2,5 pol	A primeira coluna possui 1,5 polegadas, a segunda coluna está oculta, e a terceira coluna possui 2,5 polegadas.
2 pol;;2 pol	A primeira coluna possui 2 polegadas, a segunda coluna possui 1 polegada (padrão) e a terceira coluna possui 2 polegadas. Como somente metade da terceira coluna é visível, uma barra de rolagem horizontal aparece.
(Vazio)	Todas as três colunas possuem a mesma largura (1,33 polegadas). O valor padrão desta propriedade é vazio (o E3 irá utilizar o valor padrão do sistema).

7.3.7.2.8 Font

A propriedade **Font** é utilizada para determinar a fonte do objeto. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.3.7.2.9 IMEMode

A propriedade **IMEMode** especifica o modo IME (*Input Method Editor*) de um objeto. Essa propriedade aplica-se apenas em aplicativos escritos em idiomas asiáticos (chinês simplificado, chinês tradicional, coreano e japonês) e é ignorada em outros aplicativos. Ela não tem efeito no E3 e foi mantida por motivos de compatibilidade com a especificação padrão dos objetos Microsoft Forms. As opções disponíveis são as seguintes:

Opções disponíveis para IMEMode

OPÇÃO	DESCRIÇÃO
0 - fmIMEModeNoControl	Não controla IME (padrão).
1 - fmIMEModeOn	IME ativado.
2 - fmIMEModeOff	IME desativado. Modo inglês.
3 - fmIMEModeDisable	IME desativado. O usuário não pode ativar IME pelo teclado.
4 - fmIMEModeHiragana	IME ativado com modo Hiragana de largura total.
5 - fmIMEModeKatakanaFull	IME ativado com modo Katakana de largura total.
6 - fmIMEModeKatakana	IME ativado com modo Katakana de meia largura.
7 - fmIMEModeAlphaFull	IME ativado com modo Alfanumérico de largura total.
8 - fmIMEModeAlpha	IME ativado com modo Alfanumérico de meia largura.
9 - fmIMEModeHangulFull	IME ativado com modo Hangul de largura total.
10 - fmIMEModeHangul	IME ativado com modo Hangul de meia largura.

OPÇÃO	DESCRIÇÃO
11 - fmIMEModeHanziFull	IME ativado com modo Hanzi de largura total.
12 - fmIMEModeHanzi	IME ativado com modo Hanzi de meia largura.

7.3.7.2.10 IntegralHeight

■ A propriedade **IntegralHeight** ajusta a altura da área de edição do texto, caso a área disponível para o mesmo ultrapasse o tamanho do objeto. Se esta propriedade estiver configurada para True, a altura da área de edição do texto é reajustada para coincidir com o tamanho atual do objeto, permitindo assim a exibição completa do conteúdo do texto. Caso contrário, a área de edição do texto continua com seu tamanho original. Se os textos são maiores que o espaço disponível, este não é exibido no objeto.

7.3.7.2.11 List

▼ Retorna ou define as entradas da linha e da coluna na lista do objeto. A numeração de linha e de coluna começa com zero. Isto é, o número de linha da primeira linha na lista é zero; o número de coluna da primeira coluna é zero. O número da segunda linha ou coluna é 1 e assim por diante. Esta propriedade é acessível apenas em tempo de execução.

7.3.7.2.12 ListCount

9 Retorna o número de itens contidos na lista do objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.7.2.13 ListIndex

▼ Identifica o item atualmente selecionado na lista, chamado de índice. Os valores de **ListIndex** vão de -1 ao número total de linhas em uma lista menos um (isto é, **ListCount** - 1). Quando nenhuma linha é selecionada, **ListIndex** retorna -1. Quando o usuário seleciona uma linha em uma Lista de Seleção, o sistema define o valor da propriedade **ListIndex**. O valor da propriedade **ListIndex** da primeira linha de uma lista é 0, o valor da segunda linha é 1 e assim por diante. Esta propriedade é acessível apenas em tempo de execução.

7.3.7.2.14 ListStyle

 A propriedade **ListStyle** determina o estilo da lista do objeto. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmListStylePlain**: listagem com os itens do plano de fundo em destaque.
- **1 - fmListStyleOption**: exibe botões de opção ou caixas de seleção para uma lista de várias opções. Quando o usuário seleciona um item do grupo, o botão de opção associado ao item é selecionado e os botões de opção para outros itens no grupo são desmarcados.

O valor padrão desta propriedade é **0 - fmListStylePlain**.

NOTA: A opção **1 - fmListStyleOption** só pode ser habilitada se a propriedade **MultiSelect** estiver configurada como **1 - fmMultiselectMulti**.

7.3.7.2.15 Locked

 A propriedade **Locked** habilita ou desabilita a edição do objeto. Se esta propriedade estiver configurada para True, a edição não é permitida. Caso contrário, é possível editar o objeto. Os valores configurados na propriedade **Enabled** influenciam o comportamento de **Locked**. Para maiores detalhes, consulte a propriedade **Enabled**. O valor padrão desta propriedade é False.

7.3.7.2.16 MatchEntry

 Procura, através do texto digitado pelo usuário, por uma entrada de texto que coincida com os dados existentes no objeto. Ao encontrar uma incidência do texto, a linha contendo a mesma é selecionada e o conteúdo da coluna é exibido.

As opções disponíveis são as seguintes:

- **0 - fmMatchEntryFirstLetter**: procura pela entrada de texto que coincida com o primeiro caractere digitado no objeto. Se for digitada repetidamente a mesma letra, é passada para a próxima entrada de texto que comece com esta letra e assim por diante.
- **1 - fmMatchEntryComplete**: conforme cada caractere é digitado, o objeto procura por uma entrada de texto que coincida com os caracteres digitados.
- **2 - fmMatchEntryNone**: não faz a procura no objeto.

O valor padrão desta propriedade é **1 - fmMatchEntryComplete**.

7.3.7.2.17 MultiSelect

 A propriedade **MultiSelect** indica se o objeto permite múltiplas seleções. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmMultiSelectSingle**: somente um item pode ser selecionado.
- **1 - fmMultiSelectMulti**: permite selecionar um item através da barra de espaço ou clique com o mouse, fazendo com que marque ou desmarque um item na lista.
- **2 - fmMultiSelectExtended**: permite selecionar um item através da tecla SHIFT, clicando com o mouse ou pressionando SHIFT e uma das teclas de seta, estendendo a seleção para o item atual. Pressionando a tecla CTRL e clicando com o mouse, marca ou desmarca um item.

O valor padrão desta propriedade é **0 - fmMultiSelectSingle**.

7.3.7.2.18 Selected

Seleciona ou desseleciona um item, e verifica se um item está selecionado, quando a propriedade **Multiline** está configurada como True. Para saber se um determinado item está selecionado, o índice do item a ser consultado deve ser passado, e a propriedade retorna se o item está selecionado ou não. Dessa forma, é possível ver quais itens estão selecionados quando o usuário selecionar mais de um. Esta propriedade é acessível apenas em tempo de execução. Quando o usuário não estiver utilizando múltiplas seleções, recomenda-se que sejam usadas as propriedades **Value** ou **ListIndex**.

7.3.7.2.19 SpecialEffect

 A propriedade **SpecialEffect** especifica a aparência de um objeto. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para SpecialEffect

OPÇÃO	DESCRIÇÃO
0 - fmSpecialEffectFlat	O objeto aparece plano e possui um destaque na borda, uma alteração de cor ou ambos.
1 - fmSpecialEffectRaised	O objeto possui um destaque na parte

OPÇÃO	DESCRIÇÃO
	superior esquerda e uma sombra na parte inferior direita. O objeto parece estar em relevo.
2 - fmSpecialEffectSunken	O objeto possui uma sombra na parte superior esquerda e um destaque na parte inferior direita. O objeto e a sua borda parecem estar afundados na Tela.
3 - fmSpecialEffectEtched	A borda parece estar esculpida ao redor da extremidade do objeto.
6 - fmSpecialEffectBump	O objeto possui um resalto na parte inferior direita e parece plano na parte superior esquerda.

7.3.7.2.20 Text

A Retorna o texto da opção selecionada. Esta propriedade é acessível apenas em tempo de execução.

7.3.7.2.21 TextAlign

 Especifica como o texto é alinhado no objeto. As opções disponíveis são as seguintes:

- **1 - fmTextAlignLeft:** alinha o texto com a extremidade esquerda do objeto.
- **2 - fmTextAlignCenter:** centraliza o texto com o centro do objeto.
- **3 - fmTextAlignRight:** alinha o texto com a extremidade direita do objeto.

7.3.7.2.22 TextColumn

V A propriedade **TextColumn** identifica a coluna no objeto. Os valores para a propriedade **TextColumn** vão de -1 ao número de colunas na lista. O valor **TextColumn** para a primeira coluna é 1, o valor da segunda coluna é 2, e assim por diante. A configuração de **TextColumn** como 0 exibe os valores para a propriedade **ListIndex**. A configuração de **TextColumn** como -1 exibe a primeira coluna que possui o valor da propriedade **ColumnWidths** maior do que zero.

7.3.7.2.23 TopIndex

9 A propriedade **TopIndex** define ou retorna o item da lista que aparece na posição mais próxima do topo na lista. Esta propriedade retorna o valor -1 se a lista está vazia ou não é exibida.

7.3.7.2.24 Value

V É o valor da propriedade **BoundColumn** das linhas atualmente selecionadas. Esta propriedade não tem efeito no E3 e foi mantida por motivos de compatibilidade com a especificação padrão dos objetos Microsoft Forms.

7.3.8 Botão Liga-Desliga

Esta seção contém informações sobre eventos e propriedades do objeto Botão Liga-Desliga. Este objeto não possui métodos associados.

7.3.8.1 Eventos

Esta seção contém informações sobre os eventos do objeto Botão Liga-Desliga.

7.3.8.1.1 MouseMove

MouseMove()

Ocorre quando o ponteiro do mouse é movimentado sobre o botão Liga-Desliga.

7.3.8.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Botão Liga-Desliga.

7.3.8.2.1 Accelerator

A Define ou recupera a tecla aceleradora do objeto. Esta tecla aceleradora é uma tecla de atalho que, usada em conjunto com a tecla ALT, dá o foco ao objeto. O valor padrão desta propriedade é vazio.

7.3.8.2.2 Alignment

 A propriedade **Alignment** especifica a posição do objeto relativa a sua legenda. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmAlignmentLeft**: coloca a legenda à esquerda do objeto.
- **1 - fmAlignmentRight**: coloca a legenda à direita do objeto.

Esta propriedade é acessível apenas em tempo de execução.

7.3.8.2.3 AutoSize

 A propriedade **AutoSize** ajusta a largura do texto, caso a área disponível para o mesmo ultrapasse o tamanho do objeto. Para o objeto Botão Liga-Desliga (Toggle Button), quando a propriedade estiver configurada para True, o texto é redimensionado para coincidir com o tamanho atual o objeto, permitindo assim a sua exibição completa.

7.3.8.2.4 BackStyle

 A propriedade **BackStyle** define o estilo do plano de fundo para os objetos. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmBackStyleTransparent**: define o objeto como transparente, isto é, nenhum fundo deste objeto será desenhado.
- **1 - fmBackStyleOpaque**: define o objeto como opaco, isto é, o fundo será desenhado (valor padrão).

NOTA: Esta propriedade não afeta a transparência de bitmaps. Deve-se usar um editor de imagens como o Paintbrush, por exemplo, para tornar um bitmap transparente. Nem todos os objetos ActiveX suportam bitmaps transparentes.

7.3.8.2.5 Caption

 Define o texto que será mostrado no objeto.

7.3.8.2.6 Font

A A propriedade **Font** é utilizada para determinar a fonte do objeto. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.3.8.2.7 GroupName

A A propriedade **GroupName** é utilizado para criar um grupo de objetos mutuamente exclusivos. Esta propriedade é acessível apenas em tempo de execução.

NOTA: Esta propriedade não é utilizada no E3 e foi mantida por motivos de compatibilidade com a especificação padrão dos objetos Microsoft Forms.

7.3.8.2.8 Locked

A propriedade **Locked** habilita ou desabilita a edição do objeto. Se esta propriedade estiver configurada para True, a edição não é permitida. Caso contrário, é possível editar o objeto. Os valores configurados na propriedade **Enabled** influenciam o comportamento de **Locked**. Para maiores detalhes, consulte a propriedade **Enabled**. O valor padrão desta propriedade é False.

7.3.8.2.9 Picture

 A propriedade **Picture** especifica a figura (bitmap) atribuída ao objeto. Um arquivo com imagem pode ser selecionado de duas maneiras: através da Lista de Propriedades ou via scripts, utilizando-se a função **LoadPicture** para especificar o caminho e o nome do arquivo que contém a figura. Para remover a figura, clique no valor da propriedade **Picture** e pressione a tecla DEL. A tecla BACKSPACE (tecla de espaço do teclado) não remove a figura. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("CheckBox1").Picture = LoadPicture("C:\aba.gif")  
End Sub
```

7.3.8.2.10 PicturePosition



A propriedade **PicturePosition** especifica a posição da figura atribuída ao objeto em relação a sua legenda. As opções disponíveis para esta propriedades são as seguintes:

Opções disponíveis para PicturePosition

OPÇÃO	DESCRIÇÃO
0 - fmPicturePositionLeftTop	A figura aparece à esquerda de sua legenda. A legenda é alinhada com a parte superior da figura.
1 - fmPicturePositionLeftCenter	A figura aparece à esquerda da legenda. A legenda é centralizada relativamente à imagem.
2 - fmPicturePositionLeftBottom	A figura aparece à esquerda da legenda. A legenda é alinhada com a parte inferior da figura.
3 - fmPicturePositionRightTop	A figura aparece à direita da legenda. A legenda é alinhada com a parte superior da figura.
4 - fmPicturePositionRightCenter	A figura aparece à direita da legenda. A legenda é centralizada relativamente à figura.
5 - fmPicturePositionRightBottom	A figura aparece à direita da legenda. A legenda é alinhada com a parte inferior da figura.
6 - fmPicturePositionAboveLeft	A figura aparece acima da legenda. A legenda é alinhada com a extremidade esquerda da figura.
7 - fmPicturePositionAboveCenter	A figura aparece acima da legenda. A legenda é centralizada abaixo da figura (padrão).
8 - fmPicturePositionAboveRight	A figura aparece acima da legenda. A legenda é alinhada com a extremidade direita da figura.

OPÇÃO	DESCRIÇÃO
9 - fmPicturePositionBelowLeft	A figura aparece abaixo da legenda. A legenda é alinhada com a extremidade esquerda da figura.
10 - fmPicturePositionBelowCenter	A figura aparece abaixo da legenda. A legenda é centralizada acima da figura.
11 - fmPicturePositionBelowRight	A figura aparece abaixo da legenda. A legenda é alinhada com a extremidade direita da figura.

7.3.8.2.11 SpecialEffect

 A propriedade **SpecialEffect** especifica a aparência de um objeto. Esta propriedade é acessível apenas em tempo de execução. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para SpecialEffect

OPÇÃO	DESCRIÇÃO
0 - fmSpecialEffectFlat	O objeto aparece plano e possui um destaque na borda, uma alteração de cor ou ambos.
1 - fmSpecialEffectRaised	O objeto possui um destaque na parte superior esquerda e uma sombra na parte inferior direita. O objeto parece estar em relevo.
2 - fmSpecialEffectSunken	O objeto possui uma sombra na parte superior esquerda e um destaque na parte inferior direita. O objeto e a sua borda parecem estar afundados na Tela.
3 - fmSpecialEffectEtched	A borda parece estar esculpida ao redor da extremidade do objeto.
6 - fmSpecialEffectBump	O objeto possui um resalto na parte inferior direita e parece plano na parte superior esquerda.

7.3.8.2.12 TextAlign

 Especifica como o texto é alinhado no objeto. As opções disponíveis são as seguintes:

- **1 - fmTextAlignLeft:** alinha o texto com a extremidade esquerda do objeto.
- **2 - fmTextAlignCenter:** centraliza o texto ao centro do objeto.
- **3 - fmTextAlignRight:** alinha o texto com a extremidade direita do objeto.

7.3.8.2.13 TripleState

A propriedade **TripleState** determina até três estados de valores para o objeto. Se a propriedade estiver configurada para True, o usuário poderá escolher entre três opções de estados: Falso, Verdadeiro ou Nulo. O valor Nulo é exibido como um botão sombreado. Caso contrário, o usuário pode escolher entre os valores False ou True. O valor padrão desta propriedade é False.

7.3.8.2.14 Value

Indica o valor inicial do objeto. Tem comportamento booleano; se True, o objeto inicia marcado, do contrário, seu estado inicial é não marcado. O valor padrão desta propriedade é False.

7.3.8.2.15 WordWrap

Habilita ou desabilita uma quebra de linha no texto, caso a área disponível para o texto ultrapasse os limites determinados no objeto. Para que esta propriedade funcione, a propriedade **Multiline** deve ser obrigatoriamente igual a True.

7.3.9 Editor de Texto

Esta seção contém informações sobre eventos, métodos e propriedades do objeto Editor de Texto.

7.3.9.1 Eventos

Esta seção contém informações sobre os eventos do objeto Editor de Texto.

7.3.9.1.1 DropButtonClick

DropButtonClick()

Ocorre quando a lista de opções aparece ou desaparece ao se clicar no objeto.

7.3.9.2 Métodos

Esta seção contém informações sobre os métodos do objeto Editor de Texto.

7.3.9.2.1 Copy

Copy()

Copia para a área de transferência um texto previamente selecionado. Utilize o método **Paste** para colar o texto em outro local. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("TextBox1").Copy()  
End Sub
```

7.3.9.2.2 Cut

Cut()

Recorta para a área de transferência um texto previamente selecionado. Utilize o método **Paste** para colar o texto em outro local. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("TextBox1").Cut()  
End Sub
```

7.3.9.2.3 Paste

Paste()

Insere no objeto de texto o conteúdo da área de transferência. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("TextBox1").Paste()  
End Sub
```

7.3.9.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto Editor de Texto.

7.3.9.3.1 AutoSize

☑ A propriedade **AutoSize** ajusta a largura do texto, caso a área disponível para o mesmo ultrapasse o tamanho do objeto. Para o objeto Editor de Texto (Text Box), quando a propriedade estiver configurada para True, a largura do texto é reajustada para a mesma do objeto. O valor padrão desta propriedade é False.

NOTA: Recomenda-se evitar o uso da propriedade **AutoSize** com um Editor de Texto vazio que também utilize as propriedades **Multiline** e **WordWrap**. Quando o usuário digita em um Editor de Texto com essas propriedades ajustadas para True, ele automaticamente se redimensionará como uma caixa longa e estreita, como um caractere de largura e uma linha de texto de comprimento.

7.3.9.3.2 AutoTab

☑ A propriedade **AutoTab** habilita ou desabilita a tabulação automática no objeto. Se a propriedade estiver configurada para True, a tabulação automática ocorre. Caso contrário, não é utilizada.

Após o usuário digitar o número máximo de caracteres em um objeto (utilizando a propriedade **MaxLength**), o foco se move automaticamente para o próximo objeto da ordem de tabulação, quando estes caracteres forem atingidos. Por exemplo, quando se quer que um Editor de Texto exiba dados de estoque que possuam sempre cinco caracteres, pode-se usar a propriedade **MaxLength** para especificar o número máximo de caracteres a serem introduzidos no objeto e a propriedade **AutoTab** para tabular automaticamente para o próximo objeto depois que o usuário digitar cinco caracteres.

7.3.9.3.3 AutoWordSelect

☑ Habilita ou desabilita a seleção automática de palavras no objeto. Se esta propriedade estiver configurada para True, a palavra indicada é selecionada no texto mais o espaço seguinte, caso se tenha selecionado parte dela. Caso contrário, somente o caractere indicado na palavra é selecionado.

7.3.9.3.4 BackStyle

 A propriedade **BackStyle** define o estilo do plano de fundo para os objetos. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmBackStyleTransparent:** define o objeto como transparente, isto é, nenhum fundo deste objeto será desenhado.
- **1 - fmBackStyleOpaque:** define o objeto como opaco, isto é, o fundo será desenhado (valor padrão).

NOTA: Esta propriedade não afeta a transparência de bitmaps. Deve-se usar um editor de imagens como o Paintbrush, por exemplo, para tornar um bitmap transparente. Nem todos os objetos ActiveX suportam bitmaps transparentes.

7.3.9.3.5 BorderColor

 Esta propriedade determina a cor da borda que será aplicada no objeto. Com esta propriedade, é possível aplicar a cor padrão ou personalizá-la através da sua edição. Para que esta propriedade seja aplicável, é necessário que a propriedade **BorderStyle** esteja configurada para **1 - fmBorderStyleSingle**. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.3.9.3.6 BorderStyle

 A propriedade **BorderStyle** determina o estilo de borda que será aplicada ao objeto. As opções disponíveis são:

- **0 - fmBorderStyleNone:** sem borda.
- **1 - fmBorderStyleSingle:** com borda simples.

7.3.9.3.7 CanPaste

A propriedade **CanPaste** especifica se a área de transferência contém os dados aos quais o objeto dá suporte. Se a opção estiver configurada para True, o objeto pode receber informações coladas da área de transferência. Se os dados da área de transferência estiverem em um formato ao qual o objeto não dá suporte, a propriedade **CanPaste** será False. Por exemplo, ao se tentar colar um bitmap em um objeto que só dê suporte a texto, **CanPaste** será False.

Esta propriedade é acessível apenas em tempo de execução.

7.3.9.3.8 CurLine

9 Especifica a linha corrente do objeto, ou seja, a linha que contém o ponto de inserção de texto. O número da primeira linha é 0. O valor padrão desta propriedade é 0.

7.3.9.3.9 CurTargetX

9 Retorna a posição horizontal de inserção de um texto no objeto. Esta posição é medida em unidades himétricas (um himetro é 0,0001 de um metro).

Pode-se usar **CurTargetX** e **CurX** para mover o ponto de inserção de um texto conforme o usuário percorre o conteúdo do objeto. Quando o usuário move o ponto de inserção para outra linha do texto, a propriedade **CurTargetX** especifica a posição mais indicada para o ponto de inserção do texto desejado. A propriedade **CurX** é definida neste valor, se a linha do texto for maior que o valor de **CurTargetX**. Caso contrário, a propriedade **CurX** é definida como o final da linha do texto. Esta propriedade é acessível apenas em tempo de execução.

7.3.9.3.10 CurX

9 A propriedade **CurX** especifica a posição atual horizontal do ponto de inserção no objeto. Esta propriedade é aplicada num objeto que possua várias linhas, isto é, a propriedade **Multiline** está habilitada. O valor de retorno é válido quando o objeto possui o foco. Pode-se usar a propriedade **Multiline** e a propriedade **CurX** para posicionar o ponto de inserção do texto conforme o usuário usa a barra de rolagem pelo conteúdo no objeto. Quando o usuário move o ponto de inserção para uma outra linha de texto rolando o conteúdo do objeto, a propriedade **CurTargetX** especifica a posição desejada para o ponto de inserção. A propriedade **CurX** é definida com este valor se a linha de texto for maior do que o valor de **CurTargetX**. Caso contrário, **CurX** é definido no fim da linha de texto. Esta propriedade é acessível apenas em tempo de execução.

7.3.9.3.11 DragBehavior

Habilita ou desabilita o recurso de arrastar e soltar um texto no conteúdo do objeto. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmDragBehaviorDisabled**: não permite a ação de arrastar e soltar o texto no conteúdo do objeto.

- **1 - fmDragBehaviorEnabled:** permite a ação de arrastar e soltar o texto no conteúdo do objeto.

O valor padrão desta propriedade é **0 - fmDragBehaviorDisabled**.

NOTA: A propriedade **DragBehavior** não tem efeito se a propriedade **Style** estiver configurada para 2.

7.3.9.3.12 EnterFieldBehavior

 Esta propriedade controla a forma como o conteúdo do texto é selecionado na área de edição, quando se pressiona a tecla TAB no objeto e não quando o objeto recebe o foco como um resultado do método **SetFocus**. As opções disponíveis para esta propriedade são as seguintes:

- **0 - fmEnterFieldBehaviorSelectAll (valor padrão):** seleciona todo o conteúdo do texto quando a tecla TAB é pressionada no objeto.
- **1 - fmEnterFieldBehaviorRecallSelection:** deixa a seleção inalterada.

7.3.9.3.13 EnterKeyBehavior

Define o efeito da tecla ENTER no objeto. Se esta propriedade estiver configurada para True, ao pressionar a tecla ENTER uma nova linha é criada na área de edição do texto no objeto. Caso contrário, ao pressionar a tecla ENTER o foco é passado para o próximo objeto da ordem de tabulação. Isto também ocorre se a propriedade **Multiline** estiver configurada para False, independente do valor da propriedade **EnterKeyBehavior**.

A combinação das teclas CTRL+ENTER também depende do valor da propriedade **Multiline**. Se esta propriedade estiver configurada para True, ao pressionar estas teclas uma nova linha é criada na área de edição do texto do objeto, independente do valor da propriedade **EnterKeyBehavior**. Se a propriedade for False, as teclas não terão efeito sobre o texto.

7.3.9.3.14 Font

A A propriedade **Font** é utilizada para determinar a fonte do objeto. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.3.9.3.15 HideSelection

☑ A propriedade **HideSelection** especifica se o texto selecionado continua em destaque quando um objeto não tem mais o foco. Se a propriedade estiver configurada para True, o texto selecionado não está em destaque, a menos que o objeto tenha o foco. Caso contrário, o texto selecionado sempre aparece em destaque. O valor padrão desta propriedade é True.

7.3.9.3.16 IMEMode

📄 A propriedade **IMEMode** especifica o modo IME (*Input Method Editor*) de um objeto. Essa propriedade aplica-se apenas a programas escritos em idiomas asiáticos (chinês simplificado, chinês tradicional, coreano e japonês) e é ignorada em outros aplicativos. Foi mantida por motivos de compatibilidade com a especificação padrão dos objetos Microsoft Forms. As opções disponíveis são as seguintes:

Opções disponíveis para IMEMode

OPÇÃO	DESCRIÇÃO
0 - fmIMEModeNoControl	Não controla IME (padrão).
1 - fmIMEModeOn	IME ativado.
2 - fmIMEModeOff	IME desativado. Modo inglês.
3 - fmIMEModeDisable	IME desativado. O usuário não pode ativar IME pelo teclado.
4 - fmIMEModeHiragana	IME ativado com modo Hiragana de largura total.
5 - fmIMEModeKatakanaFull	IME ativado com modo Katakana de largura total.
6 - fmIMEModeKatakana	IME ativado com modo Katakana de meia largura.
7 - fmIMEModeAlphaFull	IME ativado com modo Alfanumérico de largura total.
8 - fmIMEModeAlpha	IME ativado com modo Alfanumérico de meia largura.

OPÇÃO	DESCRIÇÃO
9 - fmIMEModeHangulFull	IME ativado com modo Hangul de largura total.
10 - fmIMEModeHangul	IME ativado com modo Hangul de meia largura.
11 - fmIMEModeHanziFull	IME ativado com modo Hanzi de largura total.
12 - fmIMEModeHanzi	IME ativado com modo Hanzi de meia largura.

7.3.9.3.17 IntegralHeight

A propriedade **IntegralHeight** ajusta a altura da área de edição do texto, caso a área disponível para o mesmo ultrapasse o tamanho do objeto. Se esta propriedade estiver configurada para True, a altura da área de edição do texto é reajustada para coincidir com o tamanho atual do objeto, permitindo assim a exibição completa do conteúdo do texto. Caso contrário, a área de edição do texto continua com seu tamanho original. Se os textos são maiores que o espaço disponível, estes não são exibidos no objeto.

7.3.9.3.18 LineCount

 A propriedade **LineCount** retorna o número de linhas do objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.9.3.19 Locked

A propriedade **Locked** habilita ou desabilita a edição do objeto. Se esta propriedade estiver configurada para True, a edição não é permitida. Caso contrário, é possível editar o objeto. Os valores configurados na propriedade **Enabled** influenciam o comportamento de **Locked**. Para maiores detalhes, consulte a propriedade **Enabled**. O valor padrão desta propriedade é False.

7.3.9.3.20 MaxLength

 A propriedade **MaxLength** determina o número máximo de caracteres no objeto. Configurando esta propriedade como 0, não há limite de caracteres no objeto.

7.3.9.3.21 Multiline

A propriedade **Multiline** indica se o texto terá múltiplas linhas (True) ou será uma caixa de texto simples (False). Isto pode ser visualizado quando o objeto Viewer está em execução. O valor padrão desta propriedade é False.

7.3.9.3.22 PasswordChar

A Converte o texto do objeto para um caractere especial configurado na propriedade. Utilize esta propriedade para proteger informações sensíveis, como senhas ou códigos de segurança. O valor de **PasswordChar** é o caractere (normalmente um asterisco) que aparece em um objeto, ao invés dos caracteres reais que o usuário digita. Se não for especificado um caractere, o controle exibe os caracteres que o usuário digita.

7.3.9.3.23 ScrollBars

 Especifica se o objeto possui barras de rolagem verticais, horizontais ou ambas. As opções disponíveis são as seguintes:

- **0 - fmScrollBarNone**: não exibe barras de rolagem.
- **1 - fmScrollBarHorizontal**: exibe uma barra de rolagem horizontal.
- **2 - fmScrollBarVertical**: exibe uma barra de rolagem vertical.

O valor padrão desta propriedade é **0 - fmScrollBarNone**.

7.3.9.3.24 SelectionMargin

Habilita ou desabilita a margem de seleção do objeto. Se esta propriedade estiver configurada para True, ao clicar na margem do objeto, o texto será selecionado. Caso contrário, ao clicar na margem o texto não será selecionado.

NOTA: Se a propriedade **SelectionMargin** estiver configurada para True quando o objeto for impresso, a margem de seleção também será impressa.

7.3.9.3.25 SelLength

 Retorna o número de caracteres selecionados no objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.9.3.26 SelStart

 Indica o ponto inicial do texto selecionado ou o ponto de inserção se nenhum texto for selecionado. Esta propriedade é acessível apenas em tempo de execução.

7.3.9.3.27 SelText

 Retorna o texto selecionado no objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.9.3.28 SpecialEffect

 A propriedade **SpecialEffect** especifica a aparência de um objeto. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para SpecialEffect

OPÇÃO	DESCRIÇÃO
0 - fmSpecialEffectFlat	O objeto aparece plano e possui um destaque na borda, uma alteração de cor ou ambos.
1 - fmSpecialEffectRaised	O objeto possui um destaque na parte superior esquerda e uma sombra na parte inferior direita. O objeto parece estar em relevo.
2 - fmSpecialEffectSunken	O objeto possui uma sombra na parte superior esquerda e um destaque na parte inferior direita. O objeto e a sua borda parecem estar afundados na Tela.
3 - fmSpecialEffectEtched	A borda parece estar esculpida ao redor da extremidade do objeto.
6 - fmSpecialEffectBump	O objeto possui um ressalto na parte inferior direita e parece plano na parte superior

OPÇÃO	DESCRIÇÃO
	esquerda.

7.3.9.3.29 TabKeyBehavior

 Determina se as tabulações são permitidas na região de edição. Se a propriedade estiver configurada para True, ao pressionar a tecla TAB é inserido um caractere de espaçamento na região de edição. Caso contrário, ao pressionar TAB, o foco é passado para o próximo objeto na ordem de tabulação.

7.3.9.3.30 Text

 Retorna o texto sendo digitado na opção selecionada. Esta propriedade é acessível apenas em tempo de execução.

7.3.9.3.31 TextAlign

 Especifica como o texto é alinhado no objeto. As opções disponíveis são as seguintes:

- **1 - fmTextAlignLeft:** alinha o texto com a extremidade esquerda do objeto.
- **2 - fmTextAlignCenter:** centraliza o texto com o centro do objeto.
- **3 - fmTextAlignRight:** alinha o texto com a extremidade direita do objeto.

7.3.9.3.32 TextLength

 Retorna o número de caracteres digitados no objeto. Esta propriedade é acessível apenas em tempo de execução.

7.3.9.3.33 Value

 É o texto na área de edição.

7.3.9.3.34 WordWrap

Habilita ou desabilita uma quebra de linha no texto, caso a área disponível para o texto ultrapasse os limites determinados no objeto. Para que esta propriedade funcione, a propriedade **Multiline** deve ser obrigatoriamente igual a True.

7.3.10 Botão Incremento-Decremento

Esta seção contém informações sobre eventos e propriedades do objeto Botão Incremento-Decremento. Este objeto não possui métodos associados.

7.3.10.1 Eventos

Esta seção contém informações sobre os eventos do objeto Botão Incremento-Decremento.

7.3.10.1.1 SpinUp

SpinUp()

Ocorre quando o usuário pressiona a seta para cima. Este evento incrementa a propriedade **Value** do objeto.

7.3.10.1.2 SpinDown

SpinDown()

Ocorre quando o usuário pressiona a seta para baixo. Este evento decrementa a propriedade **Value** do objeto.

7.3.10.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Botão Incremento-Decremento.

7.3.10.2.1 Delay

9 Especifica um tempo de espera para o objeto. A propriedade **Delay** afeta a duração de tempo entre os eventos **SpinUp**, **SpinDown** e **Change** consecutivos, gerados quando o usuário clica e mantém pressionado o botão Incremento-Decremento. O primeiro evento ocorre imediatamente. O tempo de espera até a segunda ocorrência do evento é de cinco vezes o valor especificado na propriedade **Delay**. Após o tempo inicial, o intervalo entre os eventos é o valor especificado para **Delay**.

O valor padrão de **Delay** é 50ms. Isto significa que o objeto inicia o primeiro evento depois de 250ms (cinco vezes o valor especificado) e inicia cada evento subsequente depois de 50ms.

7.3.10.2.2 Max

9 A propriedade **Max** é utilizada para determinar o limite máximo do objeto.

7.3.10.2.3 Min

9 A propriedade **Min** é utilizada para determinar o limite mínimo do objeto.

7.3.10.2.4 Orientation

 A propriedade **Orientation** é utilizada para determinar a orientação do objeto na Tela. As opções disponíveis para esta propriedade são as seguintes:

- **-1 - fmOrientationAuto**: determina automaticamente a orientação com base nas dimensões do objeto, isto é, conforme ele foi criado.
- **0 - fmOrientationVertical**: o objeto é disposto verticalmente.
- **1 - fmOrientationHorizontal**: o objeto é disposto horizontalmente.

O valor padrão desta propriedade é **-1 - fmOrientationAuto**.

7.3.10.2.5 SmallChange

9 A propriedade **SmallChange** especifica a quantidade de movimento que ocorre quando o usuário clica em uma seta de rolagem no objeto. O valor padrão desta propriedade é 1.

7.3.10.2.6 Value

▼ Número inteiro entre os valores definidos pelas propriedades **Min** e **Max**. Indica a posição inicial do incremento ou decremento. Não aceita valores menores que o valor de **Min**, nem maiores que o valor de **Max**.

7.3.11 Barra de Rolagem

Esta seção contém informações sobre eventos e propriedades do objeto Barra de Rolagem. Este objeto não possui métodos associados.

7.3.11.1 Eventos

Esta seção contém informações sobre eventos do objeto Barra de Rolagem.

7.3.11.1.1 Scroll

Scroll()

Gerado quando o ponteiro da barra de rolagem é movido para alguma direção.

7.3.11.2 Propriedades

Esta seção contém informações sobre propriedades do objeto Barra de Rolagem.

7.3.11.2.1 Delay

9 Especifica um tempo de espera para o objeto. A propriedade **Delay** afeta a duração de tempo entre os eventos **SpinUp**, **SpinDown** e **Change** consecutivos, gerados quando o usuário clica e mantém pressionada a barra de rolagem. O primeiro evento ocorre imediatamente. O tempo de espera até a segunda ocorrência do evento é de cinco vezes o valor especificado na propriedade **Delay**. Após o tempo inicial, o intervalo entre os eventos é o valor especificado para **Delay**.

O valor padrão de **Delay** é 50 ms. Isto significa que o objeto inicia o primeiro evento depois de 250 ms (cinco vezes o valor especificado) e inicia cada evento subsequente depois de 50 ms.

7.3.11.2.2 LargeChange

 Especifica a quantidade de passos do cursor da Barra de Rolagem. O valor da propriedade **LargeChange** é a quantidade pela qual a propriedade **Value** é alterada, quando o usuário clica na área entre a caixa de rolagem e o cursor da barra de rolagem. Qualquer valor inteiro é permitido para a propriedade **LargeChange**, mas o intervalo recomendado é de -32,767 a +32,767, sendo que este valor deve estar entre os valores determinados nas propriedades **Max** e **Min** da barra de rolagem.

7.3.11.2.3 Max

 A propriedade **Max** é utilizada para determinar o limite máximo do objeto.

7.3.11.2.4 Min

 A propriedade **Min** é utilizada para determinar o limite mínimo do objeto.

7.3.11.2.5 Orientation

 A propriedade **Orientation** é utilizada para determinar a orientação do objeto na Tela. As opções disponíveis para esta propriedade são as seguintes:

- **-1 - fmOrientationAuto**: determina automaticamente a orientação com base nas dimensões do objeto, isto é, conforme ele foi criado.
- **0 - fmOrientationVertical**: o objeto é disposto verticalmente.
- **1 - fmOrientationHorizontal**: o objeto é disposto horizontalmente.

O valor padrão desta propriedade é **-1 - fmOrientationAuto**.

7.3.11.2.6 ProportionalThumb

 A propriedade **ProportionalThumb** especifica se o tamanho da caixa de rolagem é igual a dimensão do objeto. Se a propriedade estiver configurada para **True**, a caixa da Barra de Rolagem tem a mesma dimensão do objeto. Caso contrário, se for dimensionado o objeto, a caixa de rolagem permanece com o tamanho original. O valor padrão desta propriedade é **True**.

7.3.11.2.7 SmallChange

 A propriedade **SmallChange** especifica a quantidade de movimento que ocorre quando o usuário clica em uma seta de rolagem no objeto. O valor padrão desta propriedade é 1.

7.3.11.2.8 Value

 Número inteiro entre os valores definidos pelas propriedades **Min** e **Max**. Indica a posição inicial da barra de rolagem. Não aceita valores menores que o valor de **Min**, nem maiores que o valor de **Max**.

7.4 E2Controls

Esta seção contém informações sobre eventos, métodos e propriedades de objetos E2Controls.

7.4.1 Propriedades Comuns

Esta seção contém informações sobre propriedades comuns aos objetos E2Controls.

7.4.1.1 Frame_BorderColor

 Define uma cor para a moldura do objeto.

7.4.1.2 Frame_BorderEnabled

Habilita ou desabilita a moldura do objeto.

7.4.1.3 Frame_BorderThickness

 Define a espessura da moldura do objeto, em pixels.

7.4.1.4 Frame_Color

 Define a cor de fundo da área de título do objeto. O valor padrão desta propriedade é cinza (**RGB(192, 192, 192)**).

7.4.1.5 Frame_Enable

Habilita ou desabilita a exibição da moldura do objeto. O valor padrão desta propriedade é verdadeiro, exceto para o objeto **E2Button**.

7.4.1.6 Frame_Enable3D

Habilita ou desabilita o efeito 3D para a moldura do objeto.

7.4.1.7 Frame_Separator

Habilita ou desabilita a exibição de uma linha separadora entre o título e o objeto.

7.4.1.8 Frame_Set3DInset

Se esta propriedade estiver ajustada para verdadeiro, a borda do objeto aparecerá rebaixada. Se estiver em falso (valor padrão), a borda do objeto aparecerá ressaltada.

7.4.1.9 Frame_Thickness3D

9 Define a espessura da borda 3D da moldura do objeto, em pixels.

7.4.1.10 Frame_Title

A Esta propriedade define o título da moldura do objeto.

7.4.1.11 Frame_TitleColor

 Define a cor da fonte do título da moldura. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.4.1.12 Frame_TitleEnabled

Habilita ou desabilita a exibição do título da moldura. O valor padrão desta propriedade é verdadeiro.

7.4.1.13 Frame_TitleFont

 A propriedade **Frame_TitleFont** é utilizada para determinar a fonte do título da moldura. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.4.2 E2Animation

Esta seção contém informações sobre propriedades do objeto E2Animation. Este objeto não possui eventos nem métodos associados.

7.4.2.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto E2Animation.

7.4.2.1.1 BackgroundColor

 Esta propriedade define a cor de fundo do objeto E2Animation. O valor padrão desta propriedade é branco (**RGB(255, 255, 255)**).

7.4.2.1.2 BlinkTime

 Define o intervalo de tempo, em milissegundos, do efeito de pisca do objeto.

7.4.2.1.3 Border

Habilita ou desabilita a exibição de uma borda ao redor do objeto.

7.4.2.1.4 DefaultZone

 Define a zona padrão que será mostrada quando o tag associado estiver fora dos limites das outras zonas definidas para o objeto.

7.4.2.1.5 IsTransparent

Se o valor desta propriedade for verdadeiro, define que o fundo do objeto será transparente, permitindo que o fundo da tela apareça. Caso contrário, a cor de fundo será sólida, definida em **BackgroundColor**.

7.4.2.1.6 Value

9 Essa propriedade define o valor que irá determinar qual a zona ativa. O valor padrão dessa propriedade é 0.

7.4.2.2 Coleção de Zonas

Esta seção contém informações sobre os métodos do objeto Coleção de Zonas. Este objeto não contém eventos nem propriedades associadas.

7.4.2.2.1 Métodos

Esta seção contém informação sobre os métodos do objeto Coleção de Zonas.

7.4.2.2.1.1 Add

Add([AxisName])

Adiciona uma nova zona na coleção de zonas. O parâmetro *AxisName* é opcional e não tem efeito, sendo mantido por questões de compatibilidade com versões anteriores.

7.4.2.2.1.2 Remove

Remove(Index)

Remove uma zona. O parâmetro *Index* indica o índice da zona que deve ser removida.

7.4.2.3 Zonas

Define um conjunto de imagens de bitmap que serão usadas para criar o efeito de animação no objeto. As zonas podem ser configuradas acessando a janela **Propriedades** do objeto, na aba **E2Animation**. As opções dessa janela são as seguintes.

Opções disponíveis na aba E2Animation

OPÇÃO	DESCRIÇÃO
Zonas	Listagem com todas as zonas definidas no objeto.
Botão Adicionar	Adiciona uma nova zona.

Botão Remover	Apaga a zona selecionada.
Zona Padrão	Define a zona selecionada como a zona padrão do objeto.
Pisca	Define se o bitmap irá piscar quando o valor do objeto estiver dentro do intervalo da zona.
Dica	Mostra um texto de ajuda sobre a zona.
Mínimo	Valor mínimo para a variação da zona.
Máximo	Valor máximo para a variação da zona.
Arquivo de imagem	Nome do arquivo de bitmap que será mostrado quando o valor do objeto estiver dentro do intervalo da zona.
Exemplo	Mostra uma prévia do arquivo de bitmap da zona selecionada.

7.4.2.3.1 Propriedades

Esta seção contém informações sobre as propriedades das Zonas do objeto E2Animation.

7.4.2.3.1.1 Blink

Indica que esta zona participa do efeito de piscar. O valor padrão dessa propriedade é falso.

7.4.2.3.1.2 Filename

A Indica qual o nome do arquivo de imagem utilizado na zona.

7.4.2.3.1.3 Maximum

9 Define o valor máximo para a zona. O valor padrão dessa propriedade é 20000.

7.4.2.3.1.4 Minimum

9 Define o valor mínimo para a zona. O valor padrão dessa propriedade é 0.

7.4.2.3.1.5 TipEnable

Habilita ou desabilita a dica para a zona. O valor padrão dessa propriedade é falso.

7.4.2.3.1.6 TipText

A Define a dica para a zona. O valor padrão dessa propriedade é em branco.

7.4.3 E2Bitmap

Esta seção contém informações sobre propriedades do objeto E2Bitmap. Este objeto não possui eventos nem métodos associados.

7.4.3.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto E2Bitmap.

7.4.3.1.1 Filename

 Define o nome do arquivo de imagem associado ao E2Bitmap. O caminho do arquivo pode ser tanto o caminho completo do arquivo no disco quanto o caminho relativo à aplicação (quando o arquivo de imagem é inserido como um recurso da aplicação). O valor padrão desta propriedade é vazio.

7.4.3.1.2 IsTransparent

Esta propriedade habilita ou desabilita a transparência do objeto, baseada na cor definida na propriedade **TransparentColor**.

7.4.3.1.3 TransparentColor

 Define qual cor será considerada pela propriedade **IsTransparent** como sendo transparente. O valor padrão desta propriedade é branco (**RGB(255, 255, 255)**).

7.4.4 E2Button

Esta seção contém informações sobre eventos e propriedades do objeto E2Button. Este objeto não possui métodos associados.

7.4.4.1 Eventos

Esta seção contém informações sobre os eventos do objeto E2Button.

7.4.4.1.1 OnRelease

OnRelease()

Este evento é gerado quando o botão do mouse é solto.

7.4.4.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto E2Button.

7.4.4.2.1 Action

 Esta propriedade define o comportamento do objeto E2Button quando é clicado. Os valores possíveis para esta propriedade são:

- **0 - Momentary:** Comportamento normal do botão, aparecendo rebaixado somente enquanto o mouse está pressionado.
- **1 - Toggle:** Possui dois estados, ligado e desligado.
- **2 - Jog:** Alterna entre dois valores, um quando o botão é pressionado e outro quando o botão é solto.

O valor padrão desta propriedade é **0 - Momentary**.

7.4.4.2.2 Alignment



Determina o alinhamento do texto do botão. Os valores possíveis desta propriedade são:

- **0 - HorizontalAlignmentLeft:** Alinha o texto à esquerda.
- **1 - HorizontalAlignmentCenter:** Alinha o texto ao centro.
- **2 - HorizontalAlignmentRight:** Alinha o texto à direita.

O valor padrão desta propriedade é **1 - HorizontalAlignmentCenter**.

7.4.4.2.3 BackgroundColor0



Define a cor de fundo do botão quando não está pressionado. O valor padrão desta propriedade é cinza (**RGB(192, 192, 192)**).

7.4.4.2.4 BackgroundColor1



Define a cor de fundo do botão quando está pressionado. O valor padrão desta propriedade é cinza (**RGB(192, 192, 192)**).

7.4.4.2.5 Bitmap0



Define a imagem do botão quando não está pressionado. O valor padrão desta propriedade é vazio.

7.4.4.2.6 Bitmap1



Define a imagem do botão quando está pressionado. O valor padrão desta propriedade é vazio.

7.4.4.2.7 Text0



Define o texto do botão quando não está pressionado. O valor padrão desta propriedade é "DESLIGADO".

7.4.4.2.8 Text1

 Define o texto do botão quando está pressionado. O valor padrão desta propriedade é "LIGADO".

7.4.4.2.9 TextColor0

 Define a cor do texto do botão quando não está pressionado. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.4.4.2.10 TextColor1

 Define a cor do texto do botão quando está pressionado. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.4.4.2.11 TextFont0

 A propriedade **TextFont0** é utilizada para determinar a fonte do botão quando não está pressionado. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.4.4.2.12 TextFont1

 A propriedade **TextFont1** é utilizada para determinar a fonte do botão quando está pressionado. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.4.4.2.13 Type

 Define o tipo do botão. Os valores possíveis para esta propriedade são os seguintes:

- **0 - ButtonTypeKey**: comportamento normal de um botão.
- **1 - ButtonTypeSwitchH**: o comportamento do botão é o de uma chave dividida horizontalmente.
- **2 - ButtonTypeSwitchV**: o comportamento do botão é o de uma chave dividida verticalmente.

- **3 - ButtonTypeLeverH**: o comportamento do botão é o de uma alavanca que se movimenta da esquerda para a direita e vice-versa.
- **4 - ButtonTypeLeverV**: o comportamento do botão é o de uma alavanca que se movimenta de cima para baixo e vice-versa.
- **5 - ButtonTypeTransparent**: o botão é transparente.
- **6 - ButtonTypeUserBitmap**: o botão alterna a exibição das imagens definidas nas propriedades **Bitmap0** e **Bitmap1**.
- **7 - ButtonTypeCheckbox**: o comportamento do botão é o mesmo de uma Caixa de Seleção.
- **8 - ButtonTypeRadio**: o comportamento do botão é o mesmo de um Botão de Opções.

O valor padrão desta propriedade é **0 - ButtonTypeKey**.

7.4.4.2.14 Value

▼ A propriedade **Value** é um **Variant** que assume o valor contido na propriedade **Value0** se o botão não está pressionado e o valor contido na propriedade **Value1** se o botão está pressionado.

7.4.4.2.15 Value0

▼ Define o valor da propriedade **Value** quando o botão não está pressionado.

7.4.4.2.16 Value1

▼ Define o valor da propriedade **Value** quando o botão está pressionado.

7.4.5 E2Display

Esta seção contém informações sobre propriedades do objeto E2Display. Este objeto não possui eventos nem métodos associados.

7.4.5.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto E2Display.

7.4.5.1.1 BackgroundColor

 Esta propriedade define a cor de fundo do objeto. O valor padrão desta propriedade é cinza (**RGB(192, 192, 192)**).

7.4.5.1.2 BackgroundStyle

 Define o estilo de fundo do objeto. Os valores desta propriedade são os seguintes:

- **0 - bsTransparent:** o fundo é transparente.
- **1 - bsOpaque:** a cor definida em **BackgroundColor** é visível.

O valor padrão desta propriedade é **1 - bsOpaque**.

7.4.5.1.3 Format

 Contém um texto que representa uma máscara dentro da qual os valores do objeto serão mostrados. Esta máscara pode representar vários tipos de valores:

- **Geral:** Não possui formatação específica, adaptando-se automaticamente ao valor especificado.
- **Número:** Apresenta números com parte inteira e fracionária. O usuário pode optar por até 15 casas decimais, por usar ou não um separador de milhares, e por apresentar números negativos com sinal ou entre parênteses. Para números muito grandes ou muito pequenos, recomenda-se utilizar o formato Científico.
- **Data:** Apresenta valores numéricos de data e hora (quando válidos). Para representar apenas a hora, use o formato equivalente.
- **Hora:** Apresenta valores numéricos de hora e data (quando válidos). Para representar apenas a data, use o formato equivalente.
- **Porcentagem:** Multiplica o número por 100 e adiciona o símbolo de porcentagem. Admite até 15 casas decimais.
- **Científico:** Apresenta o número em notação de mantissa e expoente. Ideal para

números de magnitude variada. Admite até 15 casas decimais.

- **Especial:** Permite formatar números inteiros em bases não-decimais (hexadecimal, octal ou binária, por exemplo).
- **Outro:** Permite editar diretamente o código de formatação desejado, ou selecionar um formato criado anteriormente.

7.4.5.1.4 HorizontalAlignment

 Define o alinhamento horizontal do texto do E2Display. Os valores desta propriedade são os seguintes:

- **0 - HorizontalAlignmentLeft:** alinha horizontalmente à esquerda.
- **1 - HorizontalAlignmentCenter:** alinha horizontalmente ao centro.
- **2 - HorizontalAlignmentRight:** alinha horizontalmente à direita.

O valor padrão desta propriedade é **1 - HorizontalAlignmentCenter**.

7.4.5.1.5 MultiLine

Define se o objeto possui múltiplas linhas ou não. Esta propriedade só terá efeito se a propriedade **Value** for do tipo **String**.

7.4.5.1.6 TextColor

 Define a cor do texto do objeto. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.4.5.1.7 TextFont

 A propriedade **TextFont** é utilizada para determinar a fonte do objeto. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.4.5.1.8 Value

▼ Esta propriedade contém um **Variant** que pode assumir valores de todo e qualquer tipo de dados, e a forma de apresentação desses valores será definida pela propriedade **Format**.

7.4.5.1.9 VerticalAlignment

📄 Define o alinhamento vertical do texto do E2Display. Os valores desta propriedade são os seguintes:

- **0 - VerticalAlignmentTop**: alinha verticalmente com a parte superior do objeto.
- **1 - VerticalAlignmentMiddle**: alinha verticalmente com o centro do objeto.
- **2 - VerticalAlignmentBottom**: alinha verticalmente com a parte inferior do objeto.

O valor padrão desta propriedade é **1 - VerticalAlignmentMiddle**.

7.4.6 E2Gauge

Esta seção contém informações sobre propriedades do objeto E2Gauge. Este objeto não possui eventos nem métodos associados.

7.4.6.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto E2Gauge.

7.4.6.1.1 BackgroundColor

🎨 Esta propriedade define a cor de fundo do objeto. O valor padrão desta propriedade é cinza (**RGB(128, 128, 128)**).

7.4.6.1.2 BulletsVisible

☑ Exibe ou esconde as marcas da escala em forma de *bullet*.

7.4.6.1.3 DecimalPlaces

 Esta propriedade define o número de casas decimais para o valor nominal do E2Gauge.

7.4.6.1.4 HiColorLegend

 Define a cor da legenda para o limite Alto. O valor padrão desta propriedade é amarelo (**RGB(255, 255, 0)**).

7.4.6.1.5 HiDiv

 Marca o início da escala para o limite Alto. O valor padrão dessa propriedade é 13300.

7.4.6.1.6 HiHiColorLegend

 Define a cor da legenda para o limite Muito Alto. O valor padrão desta propriedade é vermelho (**RGB(255, 0, 0)**).

7.4.6.1.7 HiHiDiv

 Marca o início da escala para o limite Muito Alto. O valor padrão dessa propriedade é 16600.

7.4.6.1.8 HiHiLimitVisible

Habilita ou desabilita a exibição do limite Muito Alto.

7.4.6.1.9 HiLimit

 O valor máximo dessa propriedade é 1 e o mínimo é limitado pela propriedade **LowLimit**. O valor padrão dessa propriedade é 0.7.

7.4.6.1.10 HiLimitVisible

Habilita ou desabilita a exibição do limite Alto.

7.4.6.1.11 LegendVisible

Exibe uma barra ao longo do objeto E2Gauge onde podem ser configuradas diferentes cores, dependendo da faixa de valores. O valor padrão desta propriedade é True.

7.4.6.1.12 LimitVisible

Define se os valores mínimos e máximos da escala aparecem ou não no gráfico.

7.4.6.1.13 LowColorLegend

 Define a cor da legenda para o limite Baixo. O valor padrão desta propriedade é verde escuro (**RGB(0, 128, 0)**).

7.4.6.1.14 LowDiv

 Marca o início da escala para o limite Baixo. O valor padrão dessa propriedade é 6600.

7.4.6.1.15 LowLimit

 O valor mínimo dessa propriedade é 0.1, e o máximo é limitado pela propriedade **HiLimit**. O valor padrão dessa propriedade é 0.62.

7.4.6.1.16 LowLimitVisible

Habilita ou desabilita a exibição do limite Baixo.

7.4.6.1.17 LowLowColorLegend

 Define a cor da legenda para o limite Muito Baixo. O valor padrão desta propriedade é verde (**RGB(0, 255, 0)**).

7.4.6.1.18 LowLowDiv

 Marca o início da escala para o limite Muito Baixo. O valor padrão dessa propriedade é 3300.

7.4.6.1.19 LowLowLimitVisible

Habilita ou desabilita a exibição do limite Muito Baixo.

7.4.6.1.20 Maximum

 Define o valor máximo da escala do E2Gauge.

7.4.6.1.21 Minimum

 Define o valor mínimo da escala do E2Gauge.

7.4.6.1.22 NeedleColor

 Define a cor do ponteiro do E2Gauge. O valor padrão desta propriedade é branco (**RGB(255, 255, 255)**).

7.4.6.1.23 NeedleThickness

 Define a espessura do ponteiro do E2Gauge, em pixels. O valor padrão dessa propriedade é 2, e só são aceitos os valores 1 ou 2.

7.4.6.1.24 NormalColor

 Define a cor da legenda para o limite Normal. O valor padrão desta propriedade é oliva (**RGB(128, 128, 0)**).

7.4.6.1.25 NumberOfPoints

 Define o número de subdivisões que estarão visíveis na escala do objeto.

7.4.6.1.26 Orientation

 Define a orientação do objeto E2Gauge. Os valores possíveis desta propriedade são os seguintes:

- **0 - Left:** a parte inferior do objeto está alinhada com o lado esquerdo da moldura.
- **1 - Up:** a parte inferior do objeto está alinhada com a parte de cima da moldura.
- **2 - Down:** a parte inferior do objeto está alinhada com a parte de baixo da moldura.
- **3 - Right:** a parte inferior do objeto está alinhada com o lado direito da moldura.

O valor padrão desta propriedade é **2 - Down**.

7.4.6.1.27 Reverted

Habilita ou desabilita a reversão da escala do objeto.

7.4.6.1.28 ShowFrame

Habilita ou desabilita a exibição do fundo ao longo do percurso do ponteiro.

7.4.6.1.29 StartAngle

 Define o ângulo inicial de exibição do ponteiro do objeto E2Gauge.

7.4.6.1.30 SubTickColor

 Define a cor das subdivisões da escala. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.4.6.1.31 SubTicksVisible

Habilita ou desabilita a exibição das subdivisões da escala.

7.4.6.1.32 TextColor

 Define a cor do texto da escala. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.4.6.1.33 TextFont

 A propriedade **TextFont** é utilizada para determinar a fonte do texto da escala. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.4.6.1.34 ThickTicks

Habilita ou desabilita a exibição de divisores da escala mais espessos. O valor padrão dessa propriedade é falso.

7.4.6.1.35 TickColor

 Define a cor dos divisores da escala. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.4.6.1.36 TicksVisible

Habilita ou desabilita a exibição das divisões da escala.

7.4.6.1.37 TickValues

Habilita ou desabilita a exibição dos valores das divisões da escala.

7.4.6.1.38 TotalNumberOfSubTicks

 Define o número total de subdivisões que serão mostradas na escala.

7.4.6.1.39 Value

 Esta propriedade define um valor entre as propriedades **Maximum** e **Minimum** da escala do objeto.

7.4.6.1.40 ValueVisible

Habilita ou desabilita a exibição do valor contido na propriedade **Value**. O valor padrão dessa propriedade é falso.

7.4.7 E2Setpoint

Esta seção contém informações sobre propriedades do objeto E2Setpoint. Este objeto não possui eventos nem métodos associados.

7.4.7.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto E2Setpoint.

7.4.7.1.1 AutoSend

Se esta propriedade estiver em True (valor padrão), o valor definido na propriedade **Value** é atualizado nas associações assim que o objeto perder o foco. Caso contrário, as associações só recebem o valor quando for usada a tecla ENTER.

7.4.7.1.2 BackgroundColor

 Esta propriedade define a cor de fundo do objeto. O valor padrão desta propriedade é cinza (**RGB(192, 192, 192)**).

7.4.7.1.3 EnableMaxLimit

Habilita ou desabilita a definição de um limite máximo para o valor do E2Setpoint.

7.4.7.1.4 EnableMinLimit

Habilita ou desabilita a definição de um limite mínimo para o valor do E2Setpoint.

7.4.7.1.5 Format

A Contém um texto que representa uma máscara dentro da qual os valores do objeto serão mostrados. Esta máscara pode representar vários tipos de valores:

- **Geral:** Não possui formatação específica, adaptando-se automaticamente ao valor especificado.
- **Número:** Apresenta números com parte inteira e fracionária. O usuário pode optar por até 15 casas decimais, por usar ou não um separador de milhares, e por apresentar números negativos com sinal ou entre parênteses. Para números muito grandes ou muito pequenos, recomenda-se utilizar o formato Científico.
- **Data:** Apresenta valores numéricos de data e hora (quando válidos). Para representar apenas a hora, use o formato equivalente.
- **Hora:** Apresenta valores numéricos de hora e data (quando válidos). Para representar apenas a data, use o formato equivalente.
- **Porcentagem:** Multiplica o número por 100 e adiciona o símbolo de porcentagem. Admite até 15 casas decimais.
- **Científico:** Apresenta o número em notação de mantissa e expoente. Ideal para números de magnitude variada. Admite até 15 casas decimais.
- **Especial:** Permite formatar números inteiros em bases não-decimais (hexadecimal, octal ou binária, por exemplo).
- **Outro:** Permite editar diretamente o código de formatação desejado, ou selecionar um formato criado anteriormente.

7.4.7.1.6 HorizontalAlignment

 Define o alinhamento horizontal do texto do E2Setpoint. Os valores desta propriedade são os seguintes:

- **0 - HorizontalAlignmentLeft:** alinha horizontalmente à esquerda.
- **1 - HorizontalAlignmentCenter:** alinha horizontalmente ao centro.
- **2 - HorizontalAlignmentRight:** alinha horizontalmente à direita.

O valor padrão desta propriedade é **1 - HorizontalAlignmentCenter**.

7.4.7.1.7 HScroll

Habilita ou desabilita a exibição de uma barra de rolagem horizontal no texto, caso a propriedade **Multiline** esteja em True.

7.4.7.1.8 MaxLimit

9 Limite máximo que pode ser atingido pela propriedade **Value** do objeto. O valor padrão dessa propriedade é 200. O limite só é verificado se a propriedade **EnableMaxLimit** estiver habilitada.

7.4.7.1.9 MinLimit

9 Limite mínimo que pode ser atingido pela propriedade **Value** do objeto. O valor padrão dessa propriedade é 0. O limite só é verificado se a propriedade **EnableMinLimit** estiver habilitada.

7.4.7.1.10 MultiLine

Define se o objeto possui múltiplas linhas ou não. Esta propriedade só estará disponível se a propriedade **Value** for do tipo **String**.

7.4.7.1.11 ReadOnly

Indica se o objeto pode ser editado ou não em tempo de execução. O valor padrão dessa propriedade é falso.

7.4.7.1.12 Refresh

Indica se o valor do E2Setpoint será atualizado ou não sempre que o valor do tag mudar. O valor padrão desta propriedade é verdadeiro.

7.4.7.1.13 SelectAllOnFocus

Habilita ou desabilita a seleção de todos os caracteres do E2Setpoint quando o objeto recebe o foco. O valor padrão dessa propriedade é verdadeiro.

7.4.7.1.14 TextColor

 Define a cor do texto do objeto. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

7.4.7.1.15 TextFont

 A propriedade **TextFont** é utilizada para determinar a fonte do objeto. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada unicamente via Studio.

7.4.7.1.16 Type

 Define o tipo de Setpoint. Os valores possíveis desta propriedade são os seguintes:

- **0 - setpointString**: aceita quaisquer caracteres alfanúmericos.
- **1 - setpointNumeric**: aceita somente caracteres numéricos e o separador decimal (ponto ou vírgula, dependendo das configurações regionais).
- **2 - setpointDateTime**: aceita somente valores de data/hora, que serão convertidos para o formato definido nas configurações regionais).

O valor padrão desta propriedade é **1 - setpointNumeric**.

7.4.7.1.17 Value

 Esta propriedade define um valor para o E2Setpoint. A forma como esse valor será visualizado é definida na propriedade **Format**.

7.4.7.1.18 VerticalAlignment

 Define o alinhamento vertical do texto do E2Setpoint. Os valores desta propriedade são os seguintes:

- **0 - VerticalAlignmentTop**: alinha verticalmente com a parte superior do objeto.
- **1 - VerticalAlignmentMiddle**: alinha verticalmente com o centro do objeto.
- **2 - VerticalAlignmentBottom**: alinha verticalmente com a parte inferior do objeto.

O valor padrão desta propriedade é **1 - VerticalAlignmentMiddle**.

7.4.7.1.19 VScroll

Habilita ou desabilita a exibição de uma barra de rolagem vertical no texto, caso a propriedade **Multiline** esteja em True.

7.4.8 E2Text

Esta seção contém informações sobre propriedades do objeto E2Text. Este objeto não possui eventos nem métodos associados.

7.4.8.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto E2Text.

7.4.8.1.1 BlinkTime

 Define o intervalo de tempo, em milissegundos, do efeito de pisca do objeto.

7.4.8.1.2 DefaultZone

 Define a zona padrão do objeto.

7.4.8.1.3 Value

 Esta propriedade contém um **Variant** que pode assumir valores de todo e qualquer tipo de dados (**Integer**, **Boolean**, **String**, etc.).

7.4.8.2 Coleção de Zonas

Esta seção contém informações sobre os métodos do objeto Coleção de Zonas do E2Text. Este objeto não possui eventos nem propriedades associadas.

7.4.8.2.1 Métodos

Esta seção contém informações sobre os métodos do objeto Coleção de Zonas do E2Text.

7.4.8.2.1.1 Add

Add([AxisName])

Adiciona uma nova zona na coleção de zonas. O parâmetro *AxisName* é opcional e não tem efeito, sendo mantido por questões de compatibilidade com versões anteriores.

7.4.8.2.1.2 Remove

Remove(Index)

Remove uma zona. O parâmetro *Index* indica o índice da zona que deve ser removida.

7.4.8.3 Zonas

Define um conjunto de zonas para o objeto E2Text. Estas zonas podem ser configuradas acessando a janela de propriedades do objeto, na aba **Zonas**. As opções disponíveis nesta aba são as seguintes:

Opções disponíveis na aba Zonas

OPÇÃO	DESCRIÇÃO
Zonas	Listagem com todas as zonas definidas no objeto.
Botão Adicionar	Adiciona uma nova zona.
Botão Remover	Remove a zona selecionada.
Pisca a cada (ms)	Define se o texto e o fundo da zona irão piscar quando o valor do objeto estiver dentro do intervalo da zona.
Mensagem	Texto da mensagem exibida quando o valor do objeto estiver dentro do intervalo da zona.
Alinhamento	Define o alinhamento do texto.
Fonte	Define a fonte do texto.

Cor de fundo	Define a cor de fundo da zona.
Transparente	Define se o fundo do objeto será transparente quando a zona estiver ativa.
Zona padrão	Define a zona selecionada como a zona padrão do objeto.
Pisca	Define se a zona irá piscar quando o valor do objeto estiver dentro do intervalo da zona.
Mínimo	Valor mínimo para a zona.
Máximo	Valor máximo para a zona.
Dica	Mostra um texto de ajuda para a zona.
Exemplo	Mostra um exemplo do comportamento da zona em tempo de execução.

7.4.8.3.1 Propriedades

Esta seção contém informações sobre as propriedades das Zonas do objeto E2Text.

7.4.8.3.1.1 BackgroundColor

 Define a cor de fundo do texto da zona. O valor padrão dessa propriedade é branco (**RGB(255, 255, 255)**).

7.4.8.3.1.2 Blink

Indica que esta zona participa do efeito de piscar. O valor padrão dessa propriedade é falso. Se estiver habilitada, esta zona alterna com a zona padrão conforme o tempo definido na propriedade **BlinkTime** do objeto E2Text.

7.4.8.3.1.3 HorizontalAlignment

 Define o alinhamento do texto. Os valores possíveis são:

- **0 - HorizontalAlignmentLeft**: alinha horizontalmente à esquerda.
- **1 - HorizontalAlignmentCenter**: alinha horizontalmente ao centro.

- **2 - HorizontalAlignmentRight**: alinha horizontalmente à direita.

O valor padrão desta propriedade é **1 - HorizontalAlignmentCenter**.

7.4.8.3.1.4 Maximum

9 Define o valor máximo para a zona. O valor padrão dessa propriedade é 20000.

7.4.8.3.1.5 Message

A Define o texto associado à zona. Esta mensagem será mostrada quando o tag associado estiver dentro dos limites da zona.

7.4.8.3.1.6 Minimum

9 Define o valor mínimo para a zona. O valor padrão dessa propriedade é 0.

7.4.8.3.1.7 TextColor

 Define a cor do texto da zona. O valor padrão dessa propriedade é preto (**RGB(0, 0, 0)**).

7.4.8.3.1.8 TextFont

A Define o estilo, a cor e o tamanho da fonte usada para mostrar o texto da mensagem.

7.4.8.3.1.9 TipEnable

Habilita ou desabilita a dica para a zona. O valor padrão dessa propriedade é falso.

7.4.8.3.1.10 TipText

A Define a dica para a zona. O valor padrão dessa propriedade é em branco.

7.4.8.3.1.11 Transparent

- Define que o fundo do objeto será transparente quando esta zona estiver ativa.

CAPÍTULO

8 Quadros e Divisores

Esta seção contém informações sobre métodos e propriedades de Quadros e Divisores. O objeto `Divisor` não possui eventos associados e o objeto `Quadro` não possui eventos nem métodos associados.

8.1 Métodos do Divisor

Esta seção contém informações sobre os métodos do objeto `Divisor`.

8.1.1 BringToFront

`BringToFront()`

Traz para frente um divisor que esteja escondido ou abaixo de outro. Exemplo:

```
Sub Botao1_Click()  
    Application.GetFrame("Teste").BringToFront()  
End Sub
```

8.1.2 CaptureScreen

`CaptureScreen(Filename)`

Captura o conteúdo de um divisor, gravando-o no arquivo de nome e caminho *Filename*, no formato BMP. Exemplo:

```
Sub CommandButton1_Click()  
    ' Quando o botão é clicado, copia o conteúdo  
    ' do divisor para o arquivo Frame.bmp.  
    Screen.Frame.CaptureScreen ("c:\temp\Framе.bmp")  
End Sub
```

8.1.3 Close

`Close(Code)`

Utilize o método `Close` para fechar a janela do quadro. O parâmetro *Code* terá o valor de retorno para o método `DoModal`, se a janela em questão tiver sido chamada por este método. Exemplo:

```
Sub BotaoFecha_Click()
    Quando BotaoFecha for clicado, fecha a janela.
    Screen.Close()
End Sub
```

8.1.4 FlashWindow

FlashWindow(*Number*, *Time*)

Este método faz com que ícone do Viewer comece a piscar na barra de tarefas do Windows. O parâmetro *Number* determina o número de vezes que a barra de tarefas deve piscar e *Time* determina o tempo (em milissegundos) entre um piscar e outro.

Exemplo:

```
Sub Texto1_Click()
    Set quadro = Application.GetFrame("_top")
    quadro.FlashWindow 50, 500
End Sub
```

8.1.5 MaximizeFrame

MaximizeFrame()

Maximiza um quadro ou Tela modal. Exemplo:

```
Sub CommandButton4_Click()
    Application.GetFrame("Outro").MaximizeFrame()
End Sub
```

8.1.6 MinimizeFrame

MinimizeFrame()

Minimiza um quadro ou Tela modal. Exemplo:

```
Sub CommandButton4_Click()
    Application.GetFrame("Outro").MinimizeFrame()
End Sub
```

8.1.7 MoveFrame

MoveFrame(*[PosX]*, *[PosY]*, *[SizeX]*, *[SizeY]*)

Move e redimensiona um Divisor (Frame) para uma coordenada e um tamanho específicos. Os parâmetros *PosX* e *PosY* informam a nova posição, em pixels, com relação à esquerda e ao topo, respectivamente. Os parâmetros *SizeX* e *SizeY* informam o novo comprimento e a nova altura, respectivamente, em pixels ou em HIMETRIC. Todos os parâmetros são opcionais. Exemplo:

```
Sub Tela2_OnPreShow(vArg)
    ' Quando a Tela2 é aberta no frame Teste, altera a posição
    ' e o tamanho do frame
    Application.GetFrame("Teste").MoveFrame 100, 100, 200, 200
End Sub
```

NOTA: Os valores de tamanho neste método podem ser informados como números ou strings. No caso de números, são considerados em pixels. No caso de strings, se vierem acompanhados da unidade "hm", serão interpretados como HIMETRIC. Qualquer outro caso é considerado em pixels.

8.1.8 OpenScreen

OpenScreen(ScreenName, Arg)

O método **OpenScreen** abre uma Tela dentro de um divisor. O parâmetro *ScreenName* determina o nome da Tela a ser aberta. É possível também especificar o percentual de zoom da Tela e a habilitar a barra de rolagem através da chave "?" como o modelo a seguir:

```
<nome-da-tela?<zoom?<habilitar-rolagem>
```

onde <nome-da-tela> é o nome da Tela a ser aberta; <ZOOM> é o percentual de zoom e <habilitar-rolagem> habilita ou desabilita a rolagem.

O percentual de zoom da Tela pode assumir os seguintes valores:

- **1:** toda a página.
- **2:** a largura da Tela ocupa 100% da largura do divisor, com altura proporcional.
- **3:** a altura da Tela ocupa 100% da altura do divisor, com largura proporcional.
- **4:** a Tela preenche totalmente o divisor.
- **5 a 100:** equivale ao percentual de zoom da Tela propriamente dito.

A habilitação de rolagem pode assumir os seguintes valores:

- **0:** desabilita a rolagem.
- **1:** habilita a rolagem.

O parâmetro *Arg* permite repassar o valor especificado à Tela através do evento **OnPreShow**. Exemplo:

```
Sub Botao1_Click()  
    ' Quando clica no botao abre a Tela2 no frame Teste  
    ' e passa o valor 1 que será usado no evento OnPreShow  
    Application.GetFrame("Teste")._.  
        OpenScreen "Tela2?100?0", "Isso é um teste."  
End Sub  
  
Sub Tela2_OnPreShow(vArg)  
    ' A caixa de mensagem vai mostrar a  
    MsgBox vArg  
End Sub
```

8.1.9 Refresh

Refresh(*Force*)

O método **Refresh** permite forçar o redesenho do conteúdo de uma Tela ou divisor. Ele deve ser usado em scripts de Viewer com processamento massivo (por exemplo, laços), ou em chamadas de métodos que demandem muito tempo e exijam indicações visuais ao usuário do progresso do processo.

Devido ao redesenho em geral ser uma operação pesada, a versão padrão do método **Refresh** (sem parâmetros) é otimizada de forma a ignorar pedidos de redesenho muito próximos, incluindo redesenho do próprio E3. Este comportamento padrão é ideal para indicações de progresso em laços, onde muitos redeseños são feitos em seqüência. O parâmetro *Force* desabilita esta otimização, garantindo que para cada chamada do método **Refresh** seja feito um redesenho. Porém, ao utilizar-se esta opção, o método **Refresh** não pode ser chamado repetidamente, como dentro de um laço, por exemplo. Exemplo:

```
Sub CommandButton1_Click()  
    ' Desenha uma barra de progresso de uma operação  
    while i < 31  
        Screen.Item("Retangulo2").  
            .HorizontalPercentFill = (i / 30) * 100  
        Frame.Refresh True  
        ' <-- alguma operação demorada -->  
    wend  
End Sub
```

8.1.10 RestoreFrame

RestoreFrame()

Permite restaurar a janela do quadro para o seu tamanho original. Exemplo:

```
Sub CommandButton1_Click()  
    Application.GetFrame("Outro").RestoreFrame()  
End Sub
```

8.1.11 SetForegroundWnd

SetForegroundWnd()

O método **SetForegroundWnd** ativa e dá o foco para a janela do Viewer. Este método é útil quando se deseja chamar a atenção do operador para algum evento ocorrido, quando a janela do Viewer encontra-se escondida ou minimizada. Exemplo:

```
Sub CommandButton1_Click()  
    Application.GetFrame("Outro").SetForegroundWnd()  
End Sub
```

8.1.12 SetDisableTaskSwitching

SetDisableTaskSwitching(Disable)

Habilita ou desabilita a troca de janelas. O parâmetro *Disable* é um valor booleano que indica se a troca de janelas será habilitada ou não. Este método atualiza a propriedade **DisableTaskSwitching** do objeto *Viewer*.

NOTAS:

- Se mais de uma instância do *Viewer* estiver rodando, e pelo menos uma destas instâncias utilizar o método **SetDisableTaskSwitching**, a alteração afetará todas as janelas.
- Caso seja necessário modificar o título ou o estilo da janela, o método **SetFrameOptions** deve ser utilizado após a chamada ao método **SetDisableTaskSwitching**.

8.1.13 SetFrameOptions

SetFrameOptions(Title, Flags)

Utilizado para configurar o título do frame na janela e o estilo da janela. O parâmetro *Title* é um **String** que contém o título da janela. Este texto será mostrado se a propriedade **Caption** da Tela estiver vazia.

O parâmetro *Flags* especifica o estilo da janela. Se este parâmetro for omitido, o valor padrão será -1. Este valor é usado para manter a configuração anterior da janela. Quando o valor especificado não for -1, pode-se modificar o estilo da janela especificando a soma dos valores das seguintes combinações:

Combinações possíveis para o parâmetro *Flags*

VALOR	DESCRIÇÃO
1	Habilita a barra de título na janela.
2	Habilita o botão Fechar na janela.
4	Habilita o botão Minimizar na janela.
8	Habilita o botão Maximizar na janela.
16	Habilita a borda na janela.

VALOR	DESCRIÇÃO
32	Especifica que a janela pode ser dimensionada. Para isso, é necessário que a janela em questão tenha borda.
64	Especifica que a janela pode ser movida.
256	Especifica que a janela ficará no topo da Tela.
512	Especifica que a janela será configurada no estilo Barra de Ferramentas.
1024	Desabilita os botões de objeto.

Exemplo:

```
Sub Tela_OnPreShow()
    Frame.SetFrameOptions("Tela de Alarmes", 114)
End Sub
```

No exemplo anterior, o valor 114 (2 + 16 + 32 + 64) indica que a janela terá o botão **Fechar** habilitado (2), terá uma borda (16), poderá ser dimensionada (32) e poderá ser movida (64). O título da janela será "Tela de Alarmes".

Nos picks **Abrir Tela** e **Abrir Tela Modal** também é possível configurar o estilo da janela durante a edição, através da caixa de diálogo Estilo da Janela. Para mais informações veja o tópico **Picks**.

NOTA: o método **SetFrameOptions** deve ser usado após a chamada ao método **SetDisableTaskSwitching** caso seja necessário modificar o título ou o estilo da janela.

8.2 Propriedades do Divisor

Esta seção contém informações sobre as propriedades do objeto Divisor.

8.2.1 IsHTML

A propriedade **IsHTML** retorna True se o divisor contém códigos HTML inseridos no quadro. Caso contrário, retorna False.

8.2.2 SplitBorder

Habilita ou desabilita a borda do divisor, e determina se a borda entre o divisor mandante e o divisor restante deve ser exibida em tempo de execução. Esta propriedade não tem efeito no divisor restante. O valor padrão desta propriedade é True.

8.2.3 SplitDockPosition

 Indica a posição do divisor desejado na Tela. As opções disponíveis são:

Opções disponíveis para SplitDockPosition

OPÇÃO	DESCRIÇÃO
0 - dockRemaining	Posiciona o divisor como restante, ou seja, ele ocupará o espaço que sobra na divisão horizontal ou vertical.
1 - dockTop	Posiciona o divisor como mandante, acima do restante.
2 - dockBottom	Posiciona o divisor como mandante, abaixo do restante.
3 - dockLeft	Posiciona o divisor como mandante, à esquerda do restante.
4 - dockRight	Posiciona o divisor como mandante, à direita do restante.

8.2.4 SplitLink

A propriedade **SplitLink** contém um *link* que deve ser mostrado no divisor. É possível especificar uma Tela do projeto, um executável ou um *link* na Internet. No caso de Telas, é possível especificar um percentual de zoom e a habilitação das barras de rolagem usando a chave "?", como no modelo <nome-da-tela>?<zoom>?<barra-de-rolagem>, onde <nome-da-tela> é o nome da Tela a ser aberta; <zoom> é o percentual de zoom e <barra-de-rolagem> é 1 para habilitar ou 0 para não habilitar. Os parâmetros <zoom> e <barra-de-rolagem> são válidos apenas se o *link* indicado se tratar de uma Tela. Caso contrário, serão ignorados. Se o parâmetro <zoom> não for informado, é assumido 100%. Se o parâmetro <barra-de-rolagem> não for informado, é assumido 1, ou seja, habilitado. Exemplo:

```
Sub CommandButton1_Click()  
Application.GetFrame("Divisor1").SplitLink = "Tela1?10?1"  
End Sub
```

8.2.5 SplitResizable

Determina se o divisor mandante pode ser redimensionado em tempo de execução. Esta propriedade não tem efeito no divisor restante. O valor padrão desta propriedade é True.

8.2.6 SplitValue

9 A propriedade **SplitValue** determina o valor que será atribuído ao divisor do quadro, que tanto pode ser % (percentagem), **hm** (HIMETRIC) ou **px** (pixels). Se a unidade for omitida, o valor será considerado em HIMETRIC. Exemplo:

```
Sub Divisor1_Click()  
    SplitValue = 10  
End Sub
```

8.3 Propriedades do Quadro

Esta seção contém informações sobre as propriedades do objeto Quadro.

8.3.1 Caption

A A propriedade **Caption** define o título da Quadro a ser mostrado na barra de título no Viewer.

CAPÍTULO

9 Associações

Links, Conexões ou **Associações** são ligações feitas entre objetos dos aplicativos e de seus módulos. A propriedade **Links** é uma coleção de links do objeto. Para maiores informações, veja o **Manual do Usuário**, tópico **Associações**.

9.1 Métodos Comuns

Esta seção contém informações sobre os métodos comuns ao objeto Associação.

9.1.1 CreateLink

CreateLink(*Property*, *Source* [, *Type*])

Este método permite criar uma conexão com uma propriedade do objeto. Em caso de sucesso, o método retorna o objeto criado. Caso contrário, ocorrerá um erro de script e o método retornará Nothing.

O método possui os seguintes parâmetros:

- **Property**: especifica o nome da propriedade para qual será criada a conexão.
- **Source**: especifica o nome do objeto-origem da conexão.
- **Type** (opcional): especifica o tipo de conexão a ser criada. Quando este parâmetro for omitido, será criada uma conexão simples.

NOTA: Nem todas as propriedades existentes em um objeto permitem a criação de conexões. Para verificar quais as propriedades permitem este recurso, acesse a aba **Associações**. Se a propriedade for inválida para a conexão, não existir ou já possuir uma conexão, ocorrerá um erro de script.

Opções disponíveis para o parâmetro Type

OPÇÃO	DESCRIÇÃO
0 - Conexão Simples	Na conexão simples, o valor da origem é copiado para a propriedade toda vez que ele for modificado.
1 - Conexão Bidirecional	Na conexão bidirecional ocorre o mesmo que na simples; porém, caso haja uma variação na propriedade, seu valor será copiado para a origem, gerando assim uma conexão de duas direções.
2 - Conexão Analógica	A conexão analógica estabelece uma escala de conversões entre a variável-origem e a propriedade.
3 - Conexão Digital	Na conexão digital, são especificados valores fixos ou alternantes para a propriedade, que são atribuídos de acordo com a fonte ser verdadeira ou falsa.
4 - Conexão por Tabela	Na conexão por tabela, pode-se estabelecer condições entre a variável, os valores e o destino. Na tabela são especificados os valores mínimos e máximos e demais configurações.
5 - Conexão Reversa	A conexão reversa é uma associação unidirecional da propriedade para a fonte.
6 - Conexão Múltipla	A conexão múltipla é semelhante a uma associação por tabela, exceto que cada linha da associação permite buscar seu valor de uma fonte diferente.

Exemplo:

```

Sub CommandButton1_Click()
    On Error Resume Next
    Dim Bind
    Set Bind = Screen.Item("Text01").Links.Item("Value")
    If Bind Is Nothing Then
        MsgBox "Text01 não está associado a nenhum objeto."
        Dim Source
        Source = "Dados.TagInterno1.Value"
        MsgBox "Criando uma conexão em " & Source & "."
        Set Bind = Screen.Item("Text01").Links._
            CreateLink("Value", Source, 0)
        Bind.BiDirectional = Screen.Item("BiDirectional").Value
        Bind.Reverse = Screen.Item("Reverse").Value
        MsgBox "Tipo: " & TypeName(Bind)
    Else
        MsgBox "Text01 já está ligado à " & Bind.Source & "."
    End If
End Sub

```

9.1.2 Item

Item (*Property, Index*)

Este método retorna um objeto *Conexão* de uma determinada propriedade de um objeto. Se for um texto, *Property* especifica o nome da propriedade cuja conexão se deseja acessar. A conexão também pode ser acessada numericamente pelo índice *Index*. Este índice deve ser de 1 até **Count**. Caso não exista conexão com a propriedade, ou o índice seja inválido, ocorrerá um erro de script. Como as demais coleções, *Links* permite o uso do comando **For Each** do VBScript. Exemplo:

```

Sub Texto1_Click()
    For Each Link In Links
        MsgBox "Origem do link: " & Link.Source
    Next
End Sub

```

9.1.3 RemoveLink

RemoveLink(*Property*)

Este método remove uma conexão com a propriedade especificada por *Property*, caso ela exista. Caso não exista conexão com a propriedade especificada, o método não tem efeito. Exemplo:

```

Sub CommandButton2_Click()
    On Error Resume Next
    Dim Bind
    Set Bind = Screen.Item("ScrollBar1").Links.Item("Value")
    If Bind Is Nothing Then
        ' Se a conexão não existe
        MsgBox "ScrollBar1 não está conectado."
    Else
        MsgBox "ScrollBar1 está ligado à " & Bind.Source & "."
        MsgBox "Removendo o link."
        Screen.Item("ScrollBar1").Links.RemoveLink("Value")
    End If
End Sub

```

9.2 Propriedades Comuns

Os **Links** trazem grande facilidade para criar diversos tipos de associações, visto que não é necessária a execução lógica ou via scripts para associar duas variáveis quaisquer. O objeto Links retorna uma coleção das associações de um objeto qualquer do E3. A propriedade comum à toda a coleção Links é **Count**.

Para acessar ou modificar uma conexão, são usadas as propriedades e métodos dos objetos de conexão (Links). Cada tipo de conexão tem propriedades específicas, exceto pelas propriedades **Property**, **Source** e **Type**, comuns a todos os tipos de conexão.

9.2.1 Count

9 Informa a quantidade de links do objeto. Retorna o valor 0 (zero) se o objeto não possuir links. Exemplo:

```
Sub CommandButton1_Click()  
    MsgBox Screen.Item("ScrollBar1").Links.Count  
End Sub
```

9.2.2 Property

A Especifica o nome da propriedade que está conectada. Ao ser modificada, permite deslocar a conexão para outra propriedade do mesmo objeto. Exemplo:

```
Sub CommandButton1_Click()  
    Dim bind  
    Set bind = Screen.Item("TableBind").Links.Item(1)  
    bind.Property = "Caption"  
End Sub
```

9.2.3 Source

A Especifica a fonte da conexão, que pode ser o nome de outro objeto da aplicação ou uma expressão mais complexa acessando vários objetos. Exemplo:

```
Sub CommandButton25_Click()  
    Dim bind  
    Set bind = Screen.Item("TableBind").Links.Item(1)  
    bind.Source = "Dados.TagDemo1.value"  
End Sub
```

9.2.4 Type

 Esta propriedade é apenas para leitura e informa o tipo de conexão. Os valores disponíveis são os seguintes:

Opções disponíveis para Type

OPÇÃO	DESCRIÇÃO
0 - bsSimple	Conexão simples.
1 - bsSimpleBiDir	Conexão bidirecional.
2 - bsAnalog	Conexão analógica.
3 - bsAnimation	Conexão digital.
4 - bsTable	Conexão por tabela.
5 - bsReverse	Conexão reversa.
6 - bsMultiSource	Conexão múltipla.

Exemplo:

```
Sub CommandButton1_Click()  
    Dim bind  
    Set bind = Screen.Item("TableBind").Links.Item(1)  
    MsgBox bind.Type & " - Conexão por Tabela"  
End Sub
```

9.3 Conexão Simples

Esta seção contém informações sobre as propriedades da Conexão Simples.

9.3.1 Propriedades

A Conexão Simples não possui nenhuma propriedade associada.

9.4 Conexão Bidirecional

Esta seção contém informações sobre propriedades da Conexão Bidirecional. Este objeto não possui eventos nem métodos associados.

9.4.1 Propriedades

Esta seção contém informações sobre as propriedades da Conexão Bidirecional.

9.4.1.1 BiDirectional

True se a conexão for bidirecional. False se a conexão for reversa ou simples.

Exemplo:

```
Sub CommandButton29_Click()  
    Dim bind  
    Set bind = Screen.Item("SimpleBind").Links.Item(1)  
    bind.BiDirectional = True  
End Sub
```

9.5 Conexão Reversa

Esta seção contém informações sobre propriedades da Conexão Reversa. Este objeto não possui eventos nem métodos associados.

9.5.1 Propriedades

Esta seção contém informações sobre as propriedades da Conexão Reversa.

9.5.1.1 Reverse

True se a conexão for reversa. False se a conexão for bidirecional ou simples.

Exemplo:

```
Sub CommandButton29_Click()  
    Dim bind  
    Set bind = Screen.Item("SimpleBind").Links.Item(1)  
    bind.Reverse = True  
End Sub
```

9.6 Conexão Digital

Esta seção contém informações sobre propriedades da Conexão Digital. Este objeto não possui eventos nem métodos associados.

9.6.1 Propriedades

Esta seção contém informações sobre as propriedades da Conexão Digital.

9.6.1.1 BlinkOff

☑ Quando esta propriedade estiver configurada para True, a propriedade conectada irá alternar periodicamente entre os valores das propriedades **OffValue** e **BlinkOffValue**, caso a fonte retorne False. Exemplo:

```
Sub BlinkOff_Change()  
    On Error Resume Next  
    Dim Bind  
    Set Bind = Screen.Item("Retangulo1")._  
        Links.Item("ForegroundColor")  
    If Bind Is Nothing Then  
        MsgBox "Retangulo1 não tem associação."  
    Else  
        MsgBox "Retangulo1 está associado a '" & Bind.Source & "'"  
        MsgBox "Mudando BlinkOff de '" & _  
            Bind.BlinkOff & "' para '" & Value  
        Bind.BlinkOff = Value  
    End If  
End Sub
```

9.6.1.2 BlinkOffValue

▼ Especifica o valor alternativo a ser assumido periodicamente pela propriedade quando a expressão da fonte resultar em False, e a propriedade **BlinkOff** estiver configurada em True. Exemplo:

```
Sub BlinkOffValue_Click()  
    On Error Resume Next  
    Dim Value  
    If Application.ShowPickColor_  
        (Value, ForegroundColor, 400, 300) Then  
        Dim Bind  
        Set Bind = Screen.Item("Retangulo1").Links._  
            Item("ForegroundColor")  
        If Bind Is Nothing Then  
            MsgBox "Retangulo1 não tem associação."  
        Else  
            MsgBox "Retangulo1 está associado a '" & Bind.Source & "'"  
            MsgBox "Mudando BlinkOffValue de '" & _  
                Bind.BlinkOffValue & "' para '" & Value  
            Bind.BlinkOffValue = Value  
        End If  
        ForegroundColor = Value  
    End If  
End Sub
```

9.6.1.3 BlinkOn

☑ Quando esta propriedade estiver configurada para True, a propriedade conectada irá alternar periodicamente entre os valores da propriedade **OnValue** e **BlinkOnValue**, caso a fonte retorne True. Exemplo:

```
Sub BlinkOn_Change()  
    On Error Resume Next  
    Dim Bind  
    Set Bind =_  
        Screen.Item("Retangulo1").Links.Item("ForegroundColor")  
    If Bind Is Nothing Then  
        MsgBox "Retangulo1 não tem associação."  
    Else  
        MsgBox "Retangulo1 está associado a '" & Bind.Source & "'"  
        MsgBox "Mudando BlinkOn de " & Bind._  
            BlinkOn & " para " & Value  
        Bind.BlinkOn = Value  
    End If  
End Sub
```

9.6.1.4 BlinkOnValue

▼ Especifica o valor alternativo a ser assumido periodicamente pela propriedade quando a expressão da fonte resultar em True e a propriedade **BlinkOn** estiver configurada para True. Exemplo:

```
Sub BlinkOnValue_Click()  
    On Error Resume Next  
    Dim Value  
    If Application.ShowPickColor_  
        (Value, ForegroundColor, 400, 300) Then  
        Dim Bind  
        Set Bind = Screen.Item("Retangulo1").Links._  
            Item("ForegroundColor")  
        If Bind Is Nothing Then  
            MsgBox "Retangulo1 não tem associação."  
        Else  
            MsgBox "Retangulo1 está associado a '" & Bind.Source & "'"  
            MsgBox "Mudando BlinkOnValue de " &_  
                Bind.BlinkOnValue & " para " & Value  
            Bind.BlinkOnValue = Value  
        End If  
        ForegroundColor = Value  
    End If  
End Sub
```

9.6.1.5 OffValue

▼ Especifica o valor assumido pela propriedade quando a expressão da fonte resultar em False. Exemplo:

```

Sub Offvalue_Click()
On Error Resume Next
Dim Value
If Application.ShowPickColor_
(Value, ForegroundColor, 400, 300) Then
Dim Bind
Set Bind = Screen.Item("Retangulo1").Links._
Item("ForegroundColor")
If Bind Is Nothing Then
MsgBox "Retangulo1 não tem associação."
Else
MsgBox "Retangulo1 está associado a '" & Bind.Source & "'"
MsgBox "Mudando Offvalue de " &
Bind.OffValue & " para " & Value
Bind.OffValue = Value
End If
ForegroundColor = value
End If
End Sub

```

9.6.1.6 OnValue

▼ Especifica o valor assumido pela propriedade quando a expressão da fonte for True.

Exemplo:

```

Sub Onvalue_Click()
On Error Resume Next
Dim Value
If Application.ShowPickColor_
(Value, ForegroundColor, 400, 300) Then
Dim Bind
Set Bind = Screen.Item("Retangulo1").Links._
Item("ForegroundColor")
If Bind Is Nothing Then
MsgBox "Retangulo1 não tem associação."
Else
MsgBox "Retangulo1 está associado a '" & Bind.Source & "'"
MsgBox "Changing Onvalue from " &
Bind.OnValue & " to " & value
Bind.OnValue = Value
End If
ForegroundColor = value
End If
End Sub

```

9.7 Conexão Analógica

Esta seção contém informações sobre propriedades da Conexão Analógica. Este objeto não possui eventos nem métodos associados.

9.7.1 Propriedades

Esta seção contém informações sobre as propriedades da Conexão Analógica.

9.7.1.1 DstHiValue

9 Especifica o valor máximo atingido na propriedade. Exemplo:

```
Sub DstHiValue_ValueChanged()  
    On Error Resume Next  
    Dim Bind  
    Set Bind = Screen.Item("ScrollBar1").Links.Item("Value")  
    Screen.Item("ScrollBar1").Max = Value  
    If Bind Is Nothing Then  
        MsgBox "ScrollBar1 não tem associação."  
    Else  
        MsgBox "ScrollBar1 está associado a '" & Bind.Source & "'" &  
        MsgBox "Mudando DstHiValue de '" &  
        Bind.DstHiValue & "' para '" & Value  
        Bind.DstHiValue = Value  
    End If  
End Sub
```

9.7.1.2 DstLoValue

9 Especifica o valor mínimo atingido na propriedade. Exemplo:

```
Sub DstLoValue_ValueChanged()  
    On Error Resume Next  
    Dim Bind  
    Set Bind = Screen.Item("ScrollBar1").Links.Item("Value")  
    Screen.Item("ScrollBar1").Min = Value  
    If Bind Is Nothing Then  
        MsgBox "ScrollBar1 não tem associação."  
    Else  
        MsgBox "ScrollBar1 está associado a '" & Bind.Source & "'" &  
        MsgBox "Mudando DstLoValue de '" &  
        Bind.DstLoValue & "' para '" & Value  
        Bind.DstLoValue = Value  
    End If  
End Sub
```

9.7.1.3 SrcHiValue

9 Especifica o valor máximo atingido na fonte. Exemplo:

```
Sub SrcHiValue_ValueChanged()  
    On Error Resume Next  
    Dim Bind  
    Set Bind = Screen.Item("ScrollBar1").Links.Item("Value")  
    Screen.Item("ScrollBar2").Max = Value  
    If Bind Is Nothing Then  
        MsgBox "ScrollBar1 não tem associado."  
    Else  
        MsgBox "ScrollBar1 está associado a '" & Bind.Source & "'" &  
        MsgBox "Mudando SrcHiValue de '" &  
        Bind.SrcHiValue & "' para '" & Value  
        Bind.SrcHiValue = Value  
    End If  
End Sub
```

9.7.1.4 SrcLoValue

9 Especifica o valor mínimo atingido na fonte. Exemplo:

```
Sub SrcLoValue_ValueChange()  
    On Error Resume Next  
    Dim Bind  
    Set Bind = Screen.Item("ScrollBar1").Links.Item("Value")  
    Screen.Item("ScrollBar2").Min = Value  
    If Bind Is Nothing Then  
        MsgBox "ScrollBar1 não tem associado."  
    Else  
        MsgBox "ScrollBar1 está associado a '" & Bind.Source & "'" &  
        MsgBox "Mudando SrcLoValue de " & _  
        Bind.SrcLoValue & " para " & Value  
        Bind.SrcLoValue = Value  
    End If  
End Sub
```

NOTA: Caso os valores especificados para as propriedades **SrcHiValue** e **SrcLoValue** sejam iguais, não é possível fazer a escala, e a conexão funcionará como se fosse uma conexão simples.

9.8 Conexão Por Tabela

Esta seção contém informações sobre métodos e propriedades da Conexão por Tabela. Este objeto não possui eventos associados.

9.8.1 Métodos

Esta seção contém informações sobre os métodos da Conexão por Tabela.

9.8.1.1 InsertRow

InsertRow([Row])

Insere uma nova linha na tabela. O parâmetro *Row* é opcional e especifica em que posição da tabela deve ser inserida a linha. Quando omitido, assume o comportamento padrão de inserir a linha no fim da tabela, o que equivale a usar *Row* igual a -1. Quando informado e não for -1, deve ser um valor entre 1 e **Count**, e a nova linha criada desloca as linhas de índice maior ou igual para a direção ascendente dos índices. Uma linha nova sempre assume os seguintes valores padrão para as propriedades:

- **Min:** 0.0
- **Max:** 1.0

- **Blink:** False
- **BlinkValue:** 0.0
- **Value:** 0.0

Exemplo:

```

Sub Retangulo1_Click()
    On Error Resume Next
    Dim Bind
    Set Bind = _
        Screen.Item("Retangulo1").Links.Item("ForegroundColor")
    If Bind Is Nothing Then
        MsgBox "Retangulo1 não tem nenhuma associação."
    Else
        Dim row
        row = Screen.Item("SetPointRow").Value
        MsgBox Bind.RowCount
        If (row < 1 OR row > Bind.RowCount) Then
            MsgBox "Número de linha inválido: " & row
        Else
            MsgBox "Acrescentando uma linha em: " & row
            Bind.InsertRow(row)
            If row = -1 Then
                row = Bind.RowCount
                Bind.Value(line) = _
                    Screen.Item("RectangleValue").ForegroundColor
                Bind.BlinkValue(line) = _
                    Screen.Item("RectangleBlinkValue").ForegroundColor
                Bind.Max(line) = Screen.Item("SetPointMax").Value
                Bind.Min(line) = Screen.Item("SetPointMin").Value
                Bind.Blink(line) = _
                    Screen.Item("CheckBoxBlink").Value
            End If
        End If
    End If
End Sub

```

9.8.1.2 RemoveRow

RemoveRow(*Row*)

Remove a linha no índice especificado. O parâmetro *Row* determina a linha da tabela a ser removida (deve ser de 1 até **Count**). Exemplo:

```

Sub RemoveRow_Click()
    On Error Resume Next
    Dim Bind
    Set Bind = _
        Screen.Item("Retangulo1").Links.Item("ForegroundColor")
    If Bind Is Nothing Then
        MsgBox "Retangulo1 não tem associação!"
    Else
        MsgBox "Retangulo1 está ligado à '" & Bind.Source & "'"
        Dim row
        row = Screen.Item("Row").Value
        MsgBox "Removendo a linha " & row
        Bind.RemoveRow row
    End If
End Sub

```

9.8.2 Propriedades

Esta seção contém informações sobre as propriedades da Conexão por Tabela.

9.8.2.1 Count

9 A propriedade **Count** informa o número de linhas da tabela. Esta propriedade é somente de leitura. Exemplo:

```
Sub Retangulo1_Click()  
    Dim Bind  
    Set Bind = Links.Item("ForegroundColor")  
    If Bind Is Nothing Then  
        MsgBox "Associação inexistente"  
    Else  
        MsgBox Bind.Count  
    End If  
End Sub
```

9.8.3 Linha da Conexão por Tabela

Esta seção contém informações sobre propriedades da Linha da Conexão por Tabela. Este objeto não possui eventos nem métodos associados.

9.8.3.1 Propriedades

Esta seção contém informações sobre as propriedades da Linha da Conexão por Tabela.

9.8.3.1.1 Blink

Determina que quando a fonte estiver no intervalo desta linha, a propriedade irá alternar periodicamente entre os valores especificados nas propriedades **Value** e **BlinkValue**. Exemplo:

```
Sub CheckBox1_Click()  
    Screen.Item("Retangulo1").Links.Item("ForegroundColor")._.  
        Item(1).Blink = Value  
    Screen.Item("Retangulo1").Links.Item("ForegroundColor")._.  
        Item(2).Blink = Value  
End Sub
```

9.8.3.1.2 BlinkValue

▼ Especifica o valor alternativo (para piscar) da propriedade a ser assumido quando a fonte estiver no intervalo especificado na linha, e a propriedade **Blink** estiver configurada para True. Exemplo:

```
Sub CommandButton1_Click()
    Dim Cor
    ' Escolhe uma cor
    Application.ShowPickColor Cor, 0, 100, 100
    Screen.Item("Retangulo1").Links.Item("ForegroundColor")._
        Item(1).BlinkValue = Cor
End Sub
```

9.8.3.1.3 Max

9 Especifica o valor máximo da fonte para uma linha da tabela.

9.8.3.1.4 Min

9 Especifica o valor mínimo da fonte para uma linha da tabela. Exemplo (para **Max** e **Min**):

```
Sub CommandButton1_Click()
    Set Bind = _
        Screen.Item("Retangulo1").Links.Item("ForegroundColor")
    Set Linha1 = Bind.Item(1)
    Linha1.Min = 0
    Linha1.Max = 20
    Set Linha2 = Bind.Item(2)
    Linha2.Min = 21
    Linha2.Max = 100
End Sub
```

9.8.3.1.5 Value

▼ Especifica o valor da propriedade a ser assumido quando a fonte estiver no intervalo especificado na linha. Exemplo:

```
Sub CommandButton1_Click()
    Dim Cor
    ' Escolhe uma cor
    Application.ShowPickColor Cor, 0, 100, 100
    Screen.Item("Retangulo1").Links.Item("ForegroundColor")._
        Item(1).Value = Cor
End Sub
```

9.9 Conexão Múltipla

Esta seção contém informações sobre métodos e propriedades da Conexão Múltipla. Este objeto não possui eventos associados.

9.9.1 Métodos

Esta seção contém informações sobre os métodos da Conexão Múltipla.

9.9.1.1 InsertRow

InsertRow(*InsertAtRow*)

Insere uma nova linha na tabela de conexões múltiplas.

9.9.1.2 RemoveRow

RemoveRow(*Row*)

Remove a linha no índice indicado pelo parâmetro *Row*.

9.9.2 Propriedades

Esta seção contém informações sobre as propriedades da Conexão Múltipla.

9.9.2.1 Count

9 Esta propriedade retorna o número de linhas da tabela de conexões múltiplas.

9.9.3 Linha da Conexão Múltipla

Esta seção contém informações sobre propriedades da Linha da Conexão Múltipla. Este objeto não possui eventos nem métodos associados.

9.9.3.1 Propriedades

Esta seção contém informações sobre as propriedades da Linha da Conexão Múltipla.

9.9.3.1.1 Max

9 Intervalo máximo para o valor da associação indicada na propriedade **Source**.

9.9.3.1.2 Min

9 Intervalo mínimo para o valor da associação indicada na propriedade **Source**.

CAPÍTULO

10

Viewer

Esta seção contém informações sobre eventos, métodos e propriedades do objeto Viewer.

10.1 Eventos

Esta seção contém informações sobre os eventos do objeto Viewer.

10.1.1 OnInactive

OnInactive()

Este evento ocorre enquanto o Viewer estiver inativo, se a propriedade **EnableInactivity** estiver em True. Ele inicia quando é constatado que o usuário está sem utilizar o Viewer por um período de tempo igual ou superior ao valor da propriedade **InactivityTime**.

Em um script para esse evento, o usuário pode programar o que deseja fazer quando o Viewer estiver inativo por um determinado período de tempo. Por exemplo, é possível determinar que após 20 minutos sem uso, será feito *logout* do Viewer. Exemplo:

```
Sub Viewer_OnInactive()  
    Logout(false)  
    If MsgBox("Essa sessão do Viewer foi fechada devido à inatividade.") = 0  
    Then  
        Application.GetFrame("").OpenScreen "TelaInicial", 0  
    End Sub
```

10.1.2 Eventos de Usuário

Esta seção contém informações sobre eventos de usuário do objeto Viewer.

10.1.2.1 OnLogin

OnLogin()

Ocorre quando um usuário executa um *login* de sistema (autenticação de usuário) com sucesso. O *login* de sistema pode ser feito através da execução do método **Login** ou quando um objeto que só pode ser acessado por usuários com um determinado nível de autorização exige autenticação. Exemplo:

```
Sub Viewer_OnLogin()  
  MsgBox "Usuário autorizado. Seja bem-vindo ao sistema!"  
End Sub
```

10.1.2.2 OnLogout

OnLogout()

Ocorre quando é executado um *logout*, ou seja, a saída do usuário do sistema. O *logout* é feito através da chamada do método **Logout**. Exemplo:

```
Sub Viewer_OnLogout()  
  MsgBox "O usuário saiu do sistema."  
End Sub
```

10.2 Métodos

Esta seção contém informações sobre os métodos do objeto Viewer.

10.2.1 CaptureScreen

CaptureScreen(*Filename*)

O método **CaptureScreen** captura a Tela atual e a salva em um arquivo. Este método é semelhante ao efeito realizado pela tecla PRINT SCREEN no Windows. O parâmetro *Filename* determina o nome do arquivo no qual a Tela será salva.

NOTA: O formato final do arquivo é Bitmap (.bmp), mesmo que no parâmetro *Filename* seja informada outra extensão (.gif, .jpg, etc.).

Exemplo:

```
Sub CommandButton1_Click()  
  Screen.Frame.CaptureScreen "c:\temp\te1a.bmp"  
End Sub
```

10.2.2 ChangePassword

ChangePassword()

O método abre uma caixa de diálogo para permitir a troca de senha do usuário atual. O método retorna True se o usuário atual tiver permissão para trocar a senha. Caso contrário, retorna False, indicando que houve falha na operação, ou que não é possível trocar a senha, porque o usuário não tem autorização. Exemplo:

```
Sub Texto1_Click()  
    If Application.ChangePassword() Then  
        MsgBox "A senha pode ser trocada."  
    Else  
        MsgBox "Não foi possível trocar a senha."  
    End If  
End Sub
```

10.2.3 DoModal

DoModal(Screen, Title, Left, Top, Width, Height, Arg, Flags)

Abre uma Tela modal. Uma Tela modal é uma janela que não permite clicar em outras Telas ou janelas enquanto ela não for fechada. O parâmetro de título passado neste método só será utilizado se a propriedade **Caption** estiver vazia; do contrário, o parâmetro deste método será ignorado.

Este método possui os seguintes parâmetros:

- **Screen**: determina o nome da Tela.
- **Title**: determina o título da janela modal.
- **Left, Top**: posição XY da janela modal em pixels.
- **Width**: largura da janela modal em pixels ou HIMETRIC.
- **Height**: altura da janela modal em pixels ou HIMETRIC.
- **Arg**: determina a variável a ser usada no evento OnPreShow da Tela.
- **Flags**: determina a combinação utilizada na janela modal. Tal combinação é feita somando-se os valores da tabela correspondentes as opções que o usuário deseja. Quando o valor especificado for -1, serão adotadas as configurações do Viewer para a janela modal. Quando este valor for diferente de -1, pode-se fazer as combinações descritas na tabela a seguir.

Combinações possíveis para o parâmetro Flags

VALOR	DESCRIÇÃO
1	Habilita a barra de título na janela
2	Habilita o botão Fechar na janela.
4	Habilita o botão Minimizar na janela.
8	Habilita o botão Maximizar na janela.
16	Habilita a borda na janela.
32	Especifica que a janela pode ser dimensionada.
64	Especifica que a janela pode ser movida.
256	Especifica que a janela ficará no topo da Tela.
512	Especifica que a janela será configurada no estilo Barra de Ferramentas.
1024	Desabilita os botões de objeto.
2048	Centraliza a Tela modal no Quadro tanto horizontal quanto verticalmente.

Exemplo:

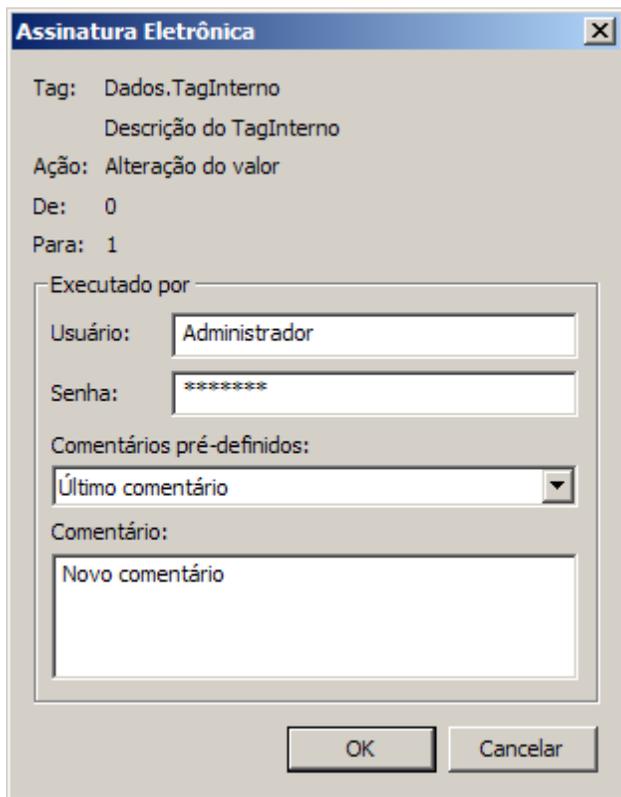
```
Sub Botao1_Click()  
    ' Ao clicar no Botão abre uma outra Tela modal  
    Application.DoModal "Tela1", "Titulo", 0, 0, 400, 200, 0, 3  
End Sub
```

NOTA: Os valores de tamanho neste método podem ser informados como números ou strings. No caso de números, são considerados em pixels. No caso de strings, se vierem acompanhados da unidade "hm", serão interpretados como HIMETRIC. Qualquer outro caso é considerado em pixels.

10.2.4 ESign

ESign(ObjName, Description, Action, [From], [To], [User], [Comment])

O método **ESign** é utilizado para fazer uma validação da alteração de um campo, através de assinatura eletrônica. Quando este método é utilizado, a seguinte caixa de diálogo é mostrada:



A caixa de diálogo intitulada "Assinatura Eletrônica" possui um cabeçalho azul com o ícone de fechar. O conteúdo é o seguinte:

- Tag: Dados.TagInterno
- Descrição do TagInterno
- Ação: Alteração do valor
- De: 0
- Para: 1
- Executado por (seção contida em uma caixa de diálogo interna):
 - Usuário: Administrador
 - Senha: *****
 - Comentários pré-definidos: Último comentário (menu suspenso)
 - Comentário: Novo comentário (área de texto)
- Botões: OK e Cancelar

Caixa de Diálogo de Assinatura Eletrônica

Parâmetros do método Esign

NOME	DESCRIÇÃO
ObjName	Texto que contém o nome de um tag ou outro objeto da aplicação.

NOME	DESCRIÇÃO
Description	Texto que contém a descrição de ObjName . Este parâmetro é opcional e, se for omitido, a caixa de diálogo tentará buscar os dados da propriedade DocString de ObjName .
Action	Texto que contém a ação a ser executada (por exemplo, "Alteração do valor"). Este parâmetro é opcional e o seu valor padrão é uma string vazia.
From	Variant que contém o valor original , ou o estado que vai ser alterado. Este parâmetro é opcional.
To	Variant que contém o novo valor do tag, ou o valor a ser aplicado em Action . Este parâmetro é opcional.
User	Texto de retorno. Recebe o nome de <i>login</i> digitado na caixa de diálogo. Este parâmetro é opcional.
Comment	Texto de retorno. Recebe o comentário digitado na caixa de diálogo. Este parâmetro é opcional.

Este método retorna True se o usuário pressionar **OK** e se os campos **Usuário** e **Senha** forem válidos. Caso contrário, se a caixa de diálogo for cancelada ou se o *login* ou a senha estiverem errados após três tentativas, o método retorna False. Em caso de falha, **User** e **Comment** são configurados para uma string vazia.

Os comentários pré-definidos são guardados no registro do Windows. Apenas os últimos 26 comentários são salvos. Cada vez que a janela é criada, os últimos comentários são buscados no registro e usados para preencher a lista de seleção. Se um novo comentário for informado pelo usuário, ele será salvo e o mais antigo será descartado, caso não haja posição livre. Se for utilizado um comentário pré-definido, este passa a ser o primeiro da lista de mais recentes. Exemplo:

```
Sub Button1_Click()
    Dim Tag, User, Comment
    Set Tag = Application.GetObject("IO.Inputs.I001")
    If Application.ESign(Tag.PathName, , "Alteração do Valor", _
        Tag.Value, 1, User, Comment) Then
        If Tag.writeEx = 1 Then
            Application.TrackEvent _
                "Tag IO.Inputs.I001 alterado para 1 " & _
                "pelo usuário" & User, Comment
        End If
    End If
End Sub
```

10.2.5 ExecuteExternalApp

ExecuteExternalApp(AppPath, Arguments, InitialDir, CmdShow, [ProcessId])

Este método executa um aplicativo externo de nome e caminho *AppPath*, com os argumentos *Arguments*, iniciando no diretório de trabalho *InitialDir*. Quando especificado um documento em *AppPath*, o aplicativo associado a este documento é executado, e o documento é passado como um dos parâmetros do aplicativo. *ProcessId* recebe o número que identifica o processo (esse número é utilizado no método **IsAppRunning** e é o mesmo valor que aparece no Gerenciador de Tarefas do Windows, na coluna **PID**). O parâmetro *CmdShow* especifica o modo de abertura da janela do aplicativo, como na tabela a seguir:

Opções disponíveis para o parâmetro CmdShow

OPÇÃO	DESCRIÇÃO
0	Esconde a janela e ativa outra janela.
1	Ativa e mostra a janela. Se a janela estiver maximizada ou minimizada, ela será restaurada para o tamanho e posição originais. Uma aplicação deve especificar esse valor quando estiver mostrando uma janela pela primeira vez.
2	Ativa a janela e mostra-a minimizada.
3	Ativa a janela e mostra-a maximizada.
4	Mostra a janela com seu tamanho e posição mais recentes. A janela ativa permanece ativa.
5	Ativa a janela e mostra-a com seu tamanho e posição atuais.
6	Minimiza a janela e ativa a próxima janela de nível superior.
7	Mostra a janela minimizada. A janela ativa permanece ativa.
8	Mostra a janela em seu estado atual. A janela ativa permanece ativa.
9	Ativa e mostra a janela. Se a janela estiver maximizada ou minimizada, ela será restaurada para o tamanho e posição originais. Uma aplicação deve especificar esse valor quando estiver restaurando uma janela que estava minimizada.

Exemplo:

```
Sub CommandButton1_Click()  
    Dim ret  
    Application.ExecuteExternalApp "calc.exe", "", "", 1, ret  
    Application.GetObject("Dados.TagInterno1").Value = ret  
End Sub
```

NOTA: O parâmetro retornado em *ProcessID* pode ser 0, caso nenhum processo tenha sido iniciado. Por exemplo, se um documento aberto for uma URL e uma instância do Internet Explorer já estiver rodando, ela mostrará o documento. Nenhum processo novo é iniciado, portanto *ProcessID* será 0.

10.2.6 Exit

Exit()

Este método fecha a janela no Viewer. Exemplo:

```
Sub Botao1_Click()  
    Application.Exit()  
EndSub
```

10.2.7 GetFrame

GetFrame([FrameName])

O método **GetFrame** procura por um objeto divisor que já esteja aberto no Viewer atual. Este método possui o parâmetro *FrameName*, que é opcional e determina o nome do quadro a ser buscado. Caso o valor especificado em *FrameName* seja vazio, ele retornará um quadro que contém todos os divisores ou a Tela ativa no momento. Com o retorno deste método, utilize os métodos do divisor, como por exemplo, o método **OpenScreen** para abrir uma outra Tela. Exemplo:

```
Sub Botao1_Click()  
    ' Ao clicar no Botão pega o quadro Menu  
    ' e substitui a Tela atual deste quadro pela tela Opções  
    Set frame = Application.GetFrame("Menu")  
    ' frame possui um objeto do tipo divisor  
    frame.OpenScreen "Opções", 0  
End Sub
```

10.2.8 GetFullUserName

GetFullUserName()

O método **GetFullUserName** retorna o nome completo do usuário logado no E3. Caso não haja nenhum usuário logado, é retornada uma string vazia.

10.2.9 GetValue

GetValue(*TagName*)

O método GetValue busca pelo valor de um objeto especificado no parâmetro *TagName*. Se *TagName* apontar para uma propriedade, este método retornará o valor da propriedade. Se por outro lado o parâmetro *TagName* especificar um objeto, o método retornará o valor da propriedade **Value** do objeto. Exemplo:

```
Sub Botao1_Click()  
    ' Ao clicar no Botão pega o valor de um tag  
    ' sendo executado em um DataServer  
    X = Application.GetValue("DataServer1.InternalTag1")  
End Sub
```

10.2.10 GetFormulaUnitDataObj

GetFormulaUnitDataObj(*FormulaName*)

Este método obtém a configuração das unidades existentes numa determinada fórmula. As unidades são o destino dos dados salvos na fórmula (valores). Este método possui o parâmetro *FormulaName*, que informa o nome da fórmula.

Utilize o método **GetFormulaUnitDataObj** para obter uma coleção das unidades de uma fórmula. Este método retorna True se conseguir realizar a operação, ou False, caso não consiga. Exemplo:

```
Sub Botao1_Click()  
    Dim val  
    ' Quando clica no botão, mostra uma caixa de  
    ' mensagem(MessageBox) com o número de Units e o nome da  
    ' Primeira Unit  
    Set obj = Application.GetFormulaUnitDataObj("Formula1")  
    MsgBox CStr(obj.Count)  
    MsgBox CStr(obj.Name(1))  
End Sub
```

10.2.11 GetFormulaValueDataObj

GetFormulaValueDataObj(*FormulaName*)

Este método obtém a configuração dos valores existentes numa determinada fórmula. Os valores são o conjunto de dados salvos na fórmula. Este método possui o parâmetro *FormulaName*, que informa o nome da fórmula.

Utilize o método **GetFormulaValueDataObj** para obter uma coleção de valores na fórmula. Este método retorna True se conseguir realizar a operação, ou False, caso não consiga. Exemplo:

```
Sub Botao1_Click()  
    Dim val  
    ' Quando clica no botão, mostra uma caixa de mensagem  
    ' (MessageBox) com o número de Conjuntos  
    ' e o nome do Primeiro Conjunto.
```

```

Set obj = Application.GetFormalValueDataObj("Formul1a1")
MsgBox CStr(obj.Count)
MsgBox CStr(obj.Name(1))
End Sub

```

10.2.12 IsAppRunning

IsAppRunning(*ProcessId*)

Indica se uma aplicação iniciada pelo método **ExecuteExternalApp** está sendo executada. Retorna True se o aplicativo identificado no sistema operacional por *ProcessId* estiver rodando. Caso contrário, retorna False. Exemplo:

```

Sub CommandButton1_Click()
Application.ExecuteExternalApp _
"www.elipse.com.br", "", "", 1, processID
While Application.IsAppRunning(processID)
' Espera pelo final da aplicação
Wend
MsgBox "Terminou a aplicação!"
End Sub

```

NOTA: O parâmetro *ProcessId* é o mesmo valor que aparece no Gerenciador de Tarefas do Windows, na coluna **PID**.

10.2.13 IsUserMemberOfGroup

IsUserMemberOfGroup(*GroupName*)

Este método verifica se o usuário logado no Viewer corrente pertence a um determinado grupo. Este método possui o parâmetro *GroupName*, que determina o nome do grupo de usuários que se quer verificar. O método retorna True se o usuário pertencer ao grupo *GroupName*, ou False, caso contrário. Exemplo:

```

Sub CommandButton1_Click()
If Application.IsUserMemberOfGroup("Suporte") Then
MsgBox ("O usuário " & Application.User & _
" é membro do grupo Suporte")
Else
MsgBox ("O usuário " & Application.User & _
" não é membro do grupo Suporte")
End If
End Sub

```

10.2.14 IsWebViewer

IsWebViewer()

Verifica se o aplicativo está sendo visualizado pelo E3 WebViewer. O método retorna True se a aplicação estiver sendo executada no E3 WebViewer. Caso contrário, retorna False.

10.2.15 LoadFormulaDlg

LoadFormulaDlg(*FormulaName*, [*UnitName*], [*ValueName*])

Este método apresenta uma caixa de diálogo que permite ao usuário escolher o conjunto de valores e a unidade de destino, carregando uma fórmula. Este método possui o parâmetro *FormulaName*, que determina o nome do objeto da fórmula que será operada.

Utilize o método **LoadFormulaDlg** para chamar uma caixa de diálogo para carregamento dos dados do objeto Fórmula especificado por *FormulaName*. Nesta caixa é possível especificar qual conjunto de valores (*UnitName*) será enviado para qual conjunto de tags (*ValueName*). Nesta caixa de mensagem o usuário conta com todos os conjuntos de valores e unidades disponíveis no objeto Fórmula, podendo atribuir um a outro livremente. Quando o usuário clicar em **OK**, o conjunto de valores será carregado na unidade especificada. Exemplo:

```
Sub Botao1_Click()  
    ' Chama a caixa de diálogo para operar  
    Dim val  
    Application.LoadFormulaDlg("Formula1")  
End Sub
```

10.2.16 LoadFormulaValues

LoadFormulaValues(*FormulaName*, *UnitName*, *ValueName*)

Este método carrega automaticamente um conjunto de valores para uma unidade de destino, apresentando uma caixa de diálogo que permite ao usuário informar valores diferentes daqueles definidos na Fórmula. Este método possui os seguintes parâmetros: *FormulaName* determina o nome da fórmula e *UnitName* determina o nome da unidade. O nome do conjunto de valores é configurado no parâmetro *ValueName*.

Uma caixa de mensagem aparecerá permitindo ao usuário informar valores diferentes daqueles definidos para cada um dos valores da Fórmula.

NOTA: O método retorna um valor lógico, ou seja, retorna True quando é executada com sucesso, e False quando falha, o que não significa que houve um erro no script.

Exemplo:

```
Sub Botao1_Click()  
    Application.LoadFormulaValues "Formula1", "Unit1", "Value1"  
End Sub
```

10.2.17 LoadFormulaValuesQuiet

LoadFormulaValuesQuiet(*FormulaName*, *UnitName*, *ValueName*)

Carrega um conjunto de valores para uma unidade de destino, sem apresentar nenhuma mensagem. Este método possui os seguintes parâmetros: *FormulaName* determina o nome da Fórmula e *UnitName* determina o nome da unidade. O nome do conjunto de valores é configurado no parâmetro *ValueName*. Exemplo:

```
Sub Botao1_Click()  
    Application.LoadFormulavaluesQuiet "Formula1", "Unidade3", "valor1"  
End Sub
```

NOTA: Este método também é acessado através do objeto **Formula**.

10.2.18 LoadReport

LoadReport(*ReportName*)

Carrega um modelo de relatório. O parâmetro *ReportName* é o nome do relatório a ser carregado. Exemplo:

```
Sub Rect_Click()  
    Carregando o relatório pré-definido  
    Set strRep = Application.LoadReport("[Relatório3]")  
    strRep.PrintPreview ' Visualizando a impressão  
End Sub
```

10.2.19 Login

Login([*Mode*])

Abre uma caixa de diálogo para o *login* (autenticação de usuário) na aplicação. O usuário logado permanece na memória até que seja feito outro *login* ou *logout* (saída do usuário da aplicação). Este método possui o parâmetro *Mode*, que é um booleano que determina se uma mensagem de confirmação ou falha da operação deverá ser mostrada (o padrão é False). Quando uma Tela estiver para ser aberta (através do método **OpenScreen**), é verificado se existe alguma configuração de segurança. Caso exista, a Tela só será aberta se o usuário logado tiver permissão. Caso não tenha permissão, é aberto um diálogo para o *login*. Exemplos:

```
Sub TelaInicial_OnPreShow(vArg)  
    Application.Login ' Usa False como padrão  
    Application.Login(True)  
    Application.Login(False)  
End Sub
```

10.2.20 LoginUser

LoginUser(*Username*, *UserPassword*)

Executa o *login* de um usuário específico sem mostrar nenhuma mensagem. O parâmetro *Username* é o nome do usuário e o parâmetro *UserPassword* é a senha desse usuário. O método retorna True se o *login* do usuário foi bem-sucedido, False caso contrário. Se o usuário passado no parâmetro *Username* estiver configurado para trocar de senha no próximo *login*, o método retorna False. Exemplo:

```
Sub CommandButton2_Click()  
    Dim a, b  
    a = Screen.Item("Usuario").Value  
    b = Screen.Item("Senha").Value  
    Screen.Item("Texto10").Value = Application.LoginUser(a, b)  
End Sub
```

10.2.21 Logout

Logout([*Mode*])

Executa o *logout* (saída do usuário atual da aplicação) do Viewer. Caso não haja um usuário logado, este método não tem efeito. A partir deste momento, será considerado que um usuário "anônimo" está utilizando a aplicação. (Pode-se utilizar o evento **OnLogout** para executar um script para ir para a Tela inicial ou finalizar a aplicação.) Este método possui o parâmetro opcional *Mode*, que é um booleano que determina se uma mensagem de confirmação ou falha da operação deverá ser mostrada (o padrão é False). Exemplo:

```
Sub TelaInicial_OnPreShow(vArg)  
    Application.Logout(True)  
End Sub
```

10.2.22 PasswordConfirm

PasswordConfirm(*Mode*)

O método abre um diálogo solicitando a reconfirmação da senha do usuário atualmente logado. O método retorna True se a senha for confirmada, ou False, caso contrário. O parâmetro booleano *Mode* determina se deve ser dado um *logout* em caso de falha na confirmação (True) ou não (False).

Se a caixa de diálogo for fechada através do botão **Cancelar**, o método retorna False. Caso não haja usuário logado, o método retorna False, mas sem abrir a caixa de diálogo. Caso a senha digitada não esteja correta, a solicitação é repetida até no máximo três vezes. Se o usuário digitar a senha incorreta nestas três vezes, a caixa de diálogo é fechada e o método retorna False. Exemplo:

```
Sub Texto1_Click()  
    If Application.PasswordConfirm(True) Then  
        MsgBox "ConfirmPassword retornou True"  
    Else
```

```
MsgBox "ConfirmPassword retornou False"  
End IF  
End Sub
```

10.2.23 Playsound

Playsound(*Filename*)

Toca um arquivo de som cujo caminho e nome estão indicados no parâmetro *Filename*. O arquivo deve seguir as seguintes especificações:

Deverá estar no formato de som do Windows (extensão .wav).

Se o arquivo estiver no projeto (adicionado através do comando **Inserir recurso**), o nome do arquivo deverá estar entre colchetes.

Se foi criada uma pasta no projeto e o arquivo foi adicionado através do comando **Inserir recurso**, o caminho deverá ser especificado entre aspas (por exemplo, "c:\som\ding.wav"). Se o arquivo estiver no diretório local, o nome não precisa de aspas, basta colocar o caminho (por exemplo: c:\som\ding.wav). Exemplo:

```
Sub TelaInicial_OnAlarm()  
    ' Se existir um alarme ativo, será tocado um som de alerta.  
    ' Quando o alarme é reconhecido, o som pára.  
    Set Alarme = Application._  
    GetObject("ConfigAlarmes1.Area1.FonteAlarme1")  
    If Alarme.ActiveNACKAlarms = True Then  
        Application.Playsound("[ringin.wav]")  
    End If  
End Sub
```

10.2.24 SelectMenu

SelectMenu(*Menu*, [*Left*], [*Top*])

Este método mostra um menu *pop-up* conforme especificado pelo parâmetro *Menu*. Este parâmetro é um texto que consiste em várias opções delimitadas pela barra vertical (|), sendo que cada um destes strings será uma opção do menu. Caso haja um conjunto de dois delimitadores sucessivos (||), um separador será inserido. Use os caracteres de abre e fecha chaves para criar um submenu. Um asterisco na frente de um string indica que a opção estará marcada. Já um sinal de exclamação fará com que a opção esteja desabilitada.

A posição da caixa de diálogo pode ser configurada através dos parâmetros *Left* e *Top*, que indicam a distância da margem esquerda e do topo da tela em pixels, respectivamente. Caso estes parâmetros não sejam informados, o menu é posicionado de acordo com o local onde o clique do mouse foi acionado.

Este método retornará 0 se nenhuma opção foi selecionada, ou o número da opção, sendo que 1 seria para a primeira opção contida no texto, 2 para a segunda e assim por diante. Exemplo:

```
Sub Botao1_Click()
```

```

op = _
Application.SelectMenu("Opção1|Opção2{*Opção2|Opção3}|Opção4|!Opção5")
If op = 1 Then
    MsgBox "Opção 1 foi escolhida"
ElseIf op = 2 Then
    MsgBox "Opção 2 foi escolhida"
ElseIf op = 3 Then
    MsgBox "Opção 3 foi escolhida"
ElseIf op = 4 Then
    MsgBox "Opção 4 foi escolhida"
ElseIf op = 0 Then
    MsgBox "Nenhuma opção foi escolhida"
End If
End Sub

```

10.2.25 SetValue

SetValue(*TagName*, *NewVal*)

Este método configura o valor de um objeto dentro do servidor. O método **SetValue** procura por um objeto ou propriedade sendo executado no servidor e atribui o valor especificado no parâmetro definido em *TagName*. O tipo e o valor do parâmetro *NewVal* deverão ser suportados pelo objeto especificado em *TagName*. Exemplo:

```

Sub Botao1_Click()
    ' Ao clicar em Botao1 atribui o valor 20
    ' para a propriedade Value do tag
    Application.SetValue "DataServer1.InternalTag1", 20
End Sub

```

10.2.26 ShowDatePicker

ShowDatePicker(*DateValue*, *Left*, *Top*, [*DefaultDate*])

Abre uma caixa de diálogo para a alteração da data e da hora. Este método retorna True se o usuário confirmar a data, ou False se o usuário cancelar a edição. A nova data é retornada no parâmetro *DateValue*. A posição da caixa de diálogo pode ser configurada através dos parâmetros *Left* e *Top*, que indicam a distância da margem esquerda e do topo da tela em pixels, respectivamente. Caso estes parâmetros não sejam informados, a caixa de diálogo fica centralizada. O valor do parâmetro *DefaultDate* é a data e a hora iniciais quando a caixa de diálogo for aberta. Se não for informada a data, o sistema assume a data atual; se não for informada a hora, iniciará como 00:00:00. Se nem a data nem a hora forem informadas, iniciará com data e hora atuais. Exemplos:

```

Sub Texto2_Click()
    Dim novaHora
    Application.ShowDatePicker novaHora, 300, 300
    MsgBox "A hora é : " & novaHora
End Sub

```

10.2.27 ShowFilePicker

ShowFilePicker(Open, FileName, [Extension], [Flags], [Filter])

Exibe as caixas de diálogo Salvar e Abrir Arquivo, do Windows. O parâmetro *Open* indica o diálogo a ser aberto; se True, abre Abrir Arquivo; se False, abre Salvar. O parâmetro *FileName* indica a variável onde será armazenado o nome do arquivo a ser salvo ou carregado caso o método retorne True. Este parâmetro deve ser obrigatoriamente uma variável. O parâmetro *Extension* é opcional e informa a extensão do arquivo padrão a ser anexada ao nome do arquivo na caixa de entrada, quando a extensão não for informada. Caso esteja vazio, nenhuma extensão será concatenada ao final do nome do arquivo. Extensões múltiplas podem ser especificadas usando-se o caractere de ponto e vírgula como delimitador. O string deve terminar com '|'|.

O parâmetro *Flags* é opcional e define o comportamento da caixa de diálogo. É um número inteiro, soma dos valores da tabela a seguir. O parâmetro *Filter* é opcional e define um conjunto de pares de strings que especificam filtros que podem ser aplicados aos arquivos. O primeiro string descreve o filtro, e o segundo indica o tipo de extensão a ser utilizada.

Combinações possíveis para o parâmetro Flags

VALOR	DESCRIÇÃO
1	CREATEPROMPT: Se o usuário especificar um arquivo que não existe, este flag faz com que ele seja questionado sobre a criação do arquivo. Se ele escolher criar o arquivo, a caixa de diálogo é fechada e o nome do arquivo é retornado no parâmetro <i>Filename</i> . Caso contrário, a caixa de diálogo permanece aberta.
2	FILEMUSTEXIST: Especifica que o usuário só pode digitar nomes de arquivos existentes. Caso contrário, a caixa de diálogo mostra um aviso na caixa de mensagens.
4	NOCHANGEDIR: Recupera o diretório atual para o valor original caso o usuário tenha mudado de diretório enquanto buscava o arquivo. Não tem efeito para Abrir Arquivo no Windows XP.
8	NODEREFERENCELINKS: Configura a caixa de diálogo para retornar o arquivo de atalho (.lnk) selecionado. Se este flag não for especificado, a caixa de diálogo retorna o caminho e nome de arquivo referenciados pelo <i>shortcut</i> .
16	NOREADONLYRETURN: Determina que o arquivo retornado não seja somente para leitura, e o diretório não é protegido contra

VALOR	DESCRIÇÃO
	gravação.
32	PATHMUSTEXIST: Especifica que o usuário pode informar apenas arquivos e diretórios válidos, caso contrário uma caixa de mensagem é mostrada para avisar o usuário.
64	READONLY: Faz o botão de opção Somente Leitura ser selecionado inicialmente quando o diálogo é criado.
128	OVERWRITEPROMPT: Faz a caixa de diálogo de Salvar Como gerar uma mensagem informando da existência do arquivo, e pede confirmação para sobrescrever o arquivo.

Exemplo de filtro:

```
"Arquivos de Chart (*.x1c)|*.x1c|Planilhas Excel (*.xls)|*.xls|Arquivos de
Dados (*.x1c;*.xls)|*.x1c; *.xls|Todos os arquivos (*.*)|*.*|"
```

10.2.28 ShowPickColor

ShowPickColor(*ColorValue*, *Color*, *Left*, *Top***)**

Abre a caixa de diálogo Cores do Windows para a escolha de uma cor. O valor decimal da cor escolhida é retornado no parâmetro *ColorValue*. O parâmetro *Color* indica uma cor previamente selecionada na paleta de cores. Se este parâmetro não for informado, ele assume 0 (preto). A posição da caixa de diálogo pode ser configurada através dos parâmetros *Left* e *Top*, que indicam respectivamente, a distância da margem esquerda e do topo da tela, em pixels. Caso estes parâmetros não sejam informados, a caixa de diálogo fica centralizada. Exemplo:

```
Sub CommandButton_Click()
    Dim novaCor
    Dim corPadrao
    corPadrao = 65280 ' Verde Claro
    If Application.ShowPickColor(novaCor, corPadrao, 90, 90) Then
        Screen.Item("Retangulo1").ForegroundColor = novaCor
        Screen.Item("Texto1").Value = novaCor
    End If
End Sub
```

10.2.29 Stopsound

Stopsound()

Pára um som que esteja tocando. Exemplo:

```
Sub CommandButton2_Click()  
    Application.StopSound()  
End Sub
```

10.2.30 ToggleValue

ToggleValue(TagName, ValA, ValB)

O método **ToggleValue** busca o valor do objeto ou propriedade sendo executado no servidor e compara com os parâmetros *ValA* e *ValB*. Se o valor buscado for igual ao *ValB*, o objeto ou propriedade especificado em *TagName* receberá o valor de *ValA*. Caso contrário, receberá o valor de *ValB*. Caso o valor de *TagName* não seja nem *ValA* nem *ValB*, o método **ToggleValue** atribuirá o valor especificado em *ValA*. Exemplo:

```
Sub Botao1_Click()  
    ' Ao clicar no Botão atribui o valor  
    ' a um tag sendo executado em um DataServer.  
    ' Atribui o valor 20 para o tag.  
    Application.SetValue "DataServer1.InternalTag1", 20  
    ' Já que o valor do InternalTag1 já é 20,  
    ' o método ToggleValue alternará o valor para 30.  
    Application.ToggleValue "DataServer1.InternalTag1", 30, 20  
End Sub
```

10.2.31 TrackEvent

TrackEvent(EventMessage, Comment, TimeStamp)

O método **TrackEvent** permite gerar eventos via script manualmente. Estes eventos podem ser gerados tanto no Viewer como no Servidor, e são registrados em uma tabela no banco de dados do aplicativo.

Parâmetros do método TrackEvent

NOME	DESCRIÇÃO
EventMessage	Contém a mensagem do evento (máximo 200 caracteres).
Comment	(Opcional) Contém comentários adicionais sobre o evento (máximo 200 caracteres).
TimeStamp	(Opcional) Indica a data/hora em que o evento ocorreu. Se não for especificado, o E3 assume a data/hora atual.

O método **TrackEvent** só vai registrar os eventos caso a opção de **Registro de Eventos** nas Opções do Domínio esteja habilitada. Os eventos são registrados numa tabela de um banco de dados que é definida também nas configurações de Registro de Eventos.

Para maiores informações sobre o Registro de Eventos do Domínio, consulte o **Manual do Usuário**. Exemplo:

```
Sub Button1_Click()  
    Dim Tag, User, Comment  
    Set Tag = Application.GetObject("IO.Inputs.IO01")  
    If Application.ESign(Tag.PathName, , "Alteração do Valor", _  
        Tag.Value, 1, User, Comment) Then  
        If Tag.WriteEx 1 Then  
            Application.TrackEvent _  
                "Tag IO.Inputs.IO01 alterado para 1 " &  
                "pele usuário" & User, Comment  
        End If  
    End If  
End Sub
```

10.2.32 UserAdministration

UserAdministration()

Este método abre uma caixa de diálogo que permite editar a lista de usuários do E3 Server. As funções disponíveis são:

- Mostrar a lista de todos os usuários.
- Apagar usuários (não é possível apagar o usuário atual).
- Adicionar e editar usuários.
- Editar as configurações de um usuário.
- Alterar a senha do usuário.
- Alterar os demais dados do usuário (*login*, nome, etc.).

IMPORTANTE: Apenas o Administrador pode ter acesso ao método **UserAdministration**. A caixa de configuração dos usuários é acessível somente ao usuário habilitado como Administrador. O usuário Administrador não pode se excluir e não pode marcar seu tipo para não-administrador.

Exemplo:

```
Sub Texto1_Click()
```

10.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto Viewer.

10.3.1 BlinkTime

9 Define o tempo, em milissegundos, entre cada mudança de estado quando um objeto tem que piscar (ou seja, toda vez que algum objeto de tela tem uma associação e é marcada a opção **Piscar**). O valor padrão desta propriedade é 200ms.

NOTA: O tempo mínimo de atualização da tela é 55ms; portanto, se esta propriedade for configurada com tempo inferior a 55ms, a configuração não terá efeito algum.

10.3.2 CacheEnable

Mantém em memória as Telas já abertas instanciadas no Viewer, permitindo uma troca mais rápida entre elas. Se a propriedade for habilitada, então o cache de Telas também é habilitado.

10.3.3 Caption

A Determina o nome da aplicação que está utilizando o Viewer. O valor padrão desta propriedade é vazio.

10.3.4 CenterWindow

Quando habilitada, determina que a janela de visualização do Viewer deverá iniciar centralizada. Caso contrário, será utilizada a configuração padrão. O valor padrão desta propriedade é True.

10.3.5 CloseButton

Se a opção estiver habilitada, o botão **Fechar** é habilitado no Viewer, e é possível utilizar este botão. Caso contrário, este botão não aparece na janela. O valor padrão desta propriedade é True.

10.3.6 DisableTaskSwitching

☑ Se estiver configurada para True, desabilita a troca de janelas no Viewer. Caso contrário, a troca de janelas é habilitada. O valor padrão desta propriedade é False. Essa propriedade pode ser modificada em tempo de execução utilizando o método **SetDisableTaskSwitching**.

10.3.7 EnableHeartbeat

☑ Habilita ou desabilita o envio de *heartbeat* (mensagem enviada em intervalos fixos, que indica que o E3 Server está ativo) entre o Viewer e o servidor. Caso o Viewer pare de receber as mensagens de *heartbeat*, é sinal de que algum problema ocorreu, portanto a conexão deve ser abortada. O valor padrão desta propriedade é False.

10.3.8 EnableInactivity

☑ Habilita ou desabilita a verificação do período de inatividade do usuário. Para maiores informações, veja o evento **OnInactive**, do objeto Viewer. O valor padrão desta propriedade é False.

10.3.9 EnableZoomMenu

☑ Se estiver configurada como True, habilita a exibição do menu de configuração do zoom da tela através do botão direito do mouse em tempo de execução, exceto quando algum script estiver configurado com informação divergente nos eventos **MouseDown** ou **MouseUp**. Caso contrário, o menu não será mostrado. O valor padrão desta propriedade é True.

10.3.10 HeartbeatPeriodMs

9 Indica o intervalo (em milissegundos) entre as mensagens de *heartbeat* enviadas pelo E3 Server. A mensagem de *heartbeat* sempre é enviada quando o servidor fica o período indicado por esta propriedade sem enviar nenhuma mensagem para o Viewer. O valor padrão desta propriedade é 2000 (dois segundos).

10.3.11 HeartbeatTimeoutMs

9 Indica o tempo, em milissegundos, que o Viewer tolera sem receber nenhuma mensagem do E3 Server. Se este tempo passar, e nenhuma mensagem for recebida, o Viewer assume que a conexão foi perdida, e inicia o processo de reconexão. Este tempo deve ser maior que o tempo determinado na propriedade **HeartbeatPeriodMs**, de preferência maior que o dobro. O valor padrão desta propriedade é 5000 (cinco segundos).

10.3.12 InactivityTime

9 Define o tempo máximo de espera por um evento de mouse ou de teclado antes do período de inatividade, em minutos. Para maiores informações, veja o evento **OnInactive** do objeto Viewer. O valor padrão desta propriedade é 5 (cinco) minutos.

Exemplo:

```
Sub CommandButton3_Click()  
    MsgBox "A aplicação ficará inativa em " & _  
        Application.InactivityTime & " minuto(s)."  
End Sub
```

10.3.13 InitialScreen

A Indica a Tela ou Quadro inicial que deverá ser mostrado quando o Viewer é chamado. Através da propriedade **WindowStyle**, é possível determinar se a janela deverá iniciar maximizada, janelada ou minimizada. O valor padrão desta propriedade é `TelaInicial`.

10.3.14 IsPlaybackMode

Se `True`, indica que o Viewer está rodando dentro de um `E3Playback`, em modo *playback*. Esta propriedade é somente de leitura e só está disponível em tempo de execução.

10.3.15 IsReadOnly

Se ajustada para `True`, indica que o Viewer está no modo **Somente Leitura** (acesso restrito).

10.3.16 LoginRetries

9 Especifica o número de tentativas de *login* do Viewer, isto é, quantas vezes a caixa de diálogo de *login* será exibida além da primeira vez. O valor padrão desta propriedade é 2.

10.3.17 MaximizeButton

Se a opção estiver habilitada (True), o botão **Maximizar** é habilitado no Viewer, e é possível utilizar esse botão. Caso contrário, esse botão não aparece na janela. O valor padrão desta propriedade é True.

10.3.18 MinimizeButton

Se a opção estiver habilitada (True), o botão **Minimizar** é habilitado no Viewer, e é possível utilizar esse botão. Caso contrário, esse botão não aparece na janela. O valor padrão desta propriedade é True.

10.3.19 Params

A Esta propriedade é um vetor de pares chave-valor que retorna os parâmetros passados para o Viewer através da opção de linha de comando **-params**. Todos os valores são retornados como Strings. Por exemplo, se a linha de comando do Viewer contiver os seguintes parâmetros:

```
Viewer -params Idioma=PTB
```

Pode-se utilizar o seguinte código para verificar qual o idioma de inicialização do Viewer.

```
Sub TelaInicial_OnStartRunning()
  Select Case Application.Params("Idioma")
    Case "PTB"
      Item("Text01").Value = "Português do Brasil"
    Case Else
      Item("Text01").Value = "Idioma não reconhecido"
  End Select
End Sub
```

NOTA: A String da chave não faz diferença entre maiúsculas e minúsculas (pode ser tanto idioma quanto Idioma), mas os valores retornados, principalmente se utilizados com um comando **Select**, fazem a diferenciação entre maiúsculas e minúsculas.

10.3.20 ReconnectDialogDelaySec

9 Indica o número de segundos que o Viewer irá esperar durante uma possível reconexão ao servidor antes de mostrar uma mensagem avisando o usuário desta ação (esta propriedade não afeta a primeira conexão). Se for igual a 0, a mensagem de reconexão sempre será mostrada. Para evitar que a mensagem seja mostrada, é recomendável colocar um número muito grande (1 bilhão, por exemplo).

NOTA: Quando a reconexão ocorre em silêncio, todas as janelas do Viewer ativo ficam desabilitadas e uma ampulheta é mostrada indicando que o aplicativo está indisponível. Durante o tempo da reconexão, não é permitido ao usuário cancelar o processo.

10.3.21 TargetDPIX

9 Define o valor de pontos por polegada, na horizontal, do monitor do computador de destino. O valor padrão desta propriedade é -1, que faz com que seja assumido o valor do computador atual.

10.3.22 TargetDPIY

9 Define o valor de pontos por polegada, na vertical, do monitor do computador de destino. O valor padrão desta propriedade é -1, que faz com que seja assumido o valor do computador atual.

10.3.23 TargetMarginX

9 Define o número de pixels que devem ser descontados da resolução horizontal da tela (a área útil do Viewer). O valor padrão da propriedade é -1, que faz com que seja assumida a configuração da janela do Viewer (com ou sem a barra de título, com ou sem borda) em conjunto com a configuração do computador atual (largura de borda e barra de título definidas pelo Windows).

10.3.24 TargetMarginY

9 Define o número de pixels que devem ser descontados da resolução vertical da tela (a área útil do Viewer). O valor padrão da propriedade é -1, que faz com que seja assumida a configuração da janela do Viewer (com ou sem a barra de título, com ou sem borda) em conjunto com a configuração do computador atual (largura de borda e barra de título definidas pelo Windows).

10.3.25 TargetResolutionX

9 Define a resolução horizontal da tela para a qual esta aplicação está destinada (em pixels). O valor padrão da propriedade é -1, que faz com que seja assumida a resolução do computador atual.

10.3.26 TargetResolutionY

9 Define a resolução vertical da tela para a qual esta aplicação está destinada (em pixels). O valor padrão da propriedade é -1, que faz com que seja assumida a resolução do computador atual.

10.3.27 TitleBar

Se a opção estiver habilitada (True), a barra de título do Viewer é mostrada, conforme as especificações da propriedade **Caption**. Do contrário, ela é escondida. O valor padrão desta propriedade é True.

10.3.28 User

A Contém o nome do usuário que está utilizando o Viewer. Esta propriedade é somente para leitura.

10.3.29 WindowBorder

Habilita ou desabilita uma borda na janela do Viewer. O valor padrão dessa propriedade é verdadeiro. Essa propriedade é de leitura e escrita, mas alterar seu valor em tempo de execução não altera a configuração do Viewer já aberto, apenas das janelas que forem abertas utilizando a configuração do Viewer.

10.3.30 WindowHeight

A Determina a altura da janela do Viewer, em pixels. O valor padrão dessa propriedade é 300.

10.3.31 WindowMovable

☑ Indica se a janela pode ser movida. O valor padrão dessa propriedade é verdadeiro. Essa propriedade é de leitura e escrita, mas alterar seu valor em tempo de execução não altera a configuração do Viewer já aberto, apenas das janelas que forem abertas utilizando a configuração do Viewer.

10.3.32 WindowResizable

☑ Indica se a janela pode ser redimensionada. Esta propriedade só tem efeito se a propriedade **WindowBorder** for verdadeira. O valor padrão dessa propriedade é verdadeiro. Essa propriedade é de leitura e escrita, mas alterar seu valor em tempo de execução não altera a configuração do Viewer já aberto, apenas das janelas que forem abertas utilizando a configuração do Viewer.

10.3.33 WindowSmallTitle

☑ Indica se a janela do Viewer deve ter a barra de título pequena. Esta propriedade só tem efeito se a propriedade **TitleBar** for verdadeira. O valor padrão dessa propriedade é falso. Essa propriedade é de leitura e escrita, mas alterar seu valor em tempo de execução não altera a configuração do Viewer já aberto, apenas das janelas que forem abertas utilizando a configuração do Viewer.

10.3.34 WindowStayOnTop

☑ Indica se a janela do Viewer deve sempre se sobrepor às outras janelas. O valor padrão dessa propriedade é falso. Essa propriedade é de leitura e escrita, mas alterar seu valor em tempo de execução não altera a configuração do Viewer já aberto, apenas das janelas que forem abertas utilizando a configuração do Viewer.

10.3.35 WindowStyle

📄 Define o estilo inicial de janela do Viewer. As opções disponíveis são:

- **0 - Maximized**: inicia o Viewer maximizado.
- **1 - Windowed**: inicia o Viewer janelado.
- **2 - Minimized**: inicia o Viewer minimizado.

10.3.36 WindowWidth

9 Determina a largura da janela do Viewer, em pixels. O valor padrão dessa propriedade é 400.

10.4 Objeto _top

Esta seção contém informações sobre métodos e propriedades do objeto `_top`. Este objeto não possui eventos associados.

10.4.1 Métodos

Esta seção contém informações sobre os métodos do objeto `_top`.

10.4.1.1 BringToFront

BringToFront()

Traz para frente um divisor que esteja escondido ou abaixo de outro. Exemplo:

```
Sub [_top_OnStartRunning]()
    BringToFront()
End Sub
```

10.4.1.2 Close

Close(*Code*)

Utilize o método **Close** para fechar a janela do objeto `_top`. O parâmetro *Code* terá o valor de retorno para o método **DoModal**, se o objeto em questão tiver sido chamado por este método. Exemplo:

```
Sub BotaoFecha_Click()
    ' Quando BotaoFecha for clicado, fecha o objeto _top
    Item("_top").close()
End Sub
```

10.4.1.3 FlashWindow

FlashWindow(*Number*, *Time*)

Este método faz com que o ícone do Viewer comece a piscar na barra de tarefas do Windows. O parâmetro *Number* determina o número de vezes que a tarefa deve piscar e *Time* determina o tempo (em milissegundos) entre um piscar e outro. Exemplo:

```
Sub Texto1_Click()
    set quadro = Application.GetFrame("_top")
    quadro.FlashWindow 50, 500
End Sub
```

```
End Sub
```

10.4.1.4 MaximizeFrame

MaximizeFrame()

Maximiza um quadro ou Tela modal. Exemplo:

```
Sub CommandButton4_Click()  
    Application.Item("_top").MaximizeFrame()  
End Sub
```

10.4.1.5 MinimizeFrame

MinimizeFrame()

Minimiza um quadro ou Tela modal. Exemplo:

```
Sub CommandButton4_Click()  
    Application.Item("_top").MinimizeFrame()  
End Sub
```

10.4.1.6 MoveFrame

MoveFrame([PosX], [PosY], [SizeX], [SizeY])

Move e redimensiona um Divisor (Frame) para uma coordenada e um tamanho específicos. Os parâmetros *PosX* e *PosY* informam a nova posição, em pixels, com relação à esquerda e ao topo, respectivamente. Os parâmetros *SizeX* e *SizeY* informam o novo comprimento e a nova altura, respectivamente, em pixels ou em HIMETRIC. Todos os parâmetros são opcionais. Exemplo:

```
Sub Tela2_OnPreShow(vArg)  
    ' Quando a Tela2 é aberta no objeto _top, altera a posição  
    ' e o tamanho do frame  
    Application.Item("_top").MoveFrame 100, 100, 200, 200  
End Sub
```

NOTA: Os valores de tamanho neste método podem ser informados como números ou strings. No caso de números, são considerados em pixels. No caso de strings, se vierem acompanhados da unidade "hm", serão interpretados como HIMETRIC. Qualquer outro caso é considerado em pixels.

10.4.1.7 OpenScreen

OpenScreen(ScreenName, Arg)

O método **OpenScreen** abre uma Tela dentro de um divisor. O parâmetro *ScreenName* determina o nome da Tela a ser aberta. É possível também especificar o percentual de zoom da Tela e a habilitar a barra de rolagem através da chave "?" como o modelo a

seguir:

```
<nome-da-tela>?<zoom>?<habilitar-rolagem>
```

onde <nome-da-tela> é o nome da Tela a ser aberta; <zoom> é o percentual de zoom e <habilitar-rolagem> habilita ou desabilita a rolagem.

O percentual de zoom da Tela pode assumir os seguintes valores:

- **1**: toda a página.
- **2**: a largura da Tela ocupa 100% da largura do divisor, com altura proporcional.
- **3**: a altura da Tela ocupa 100% da altura do divisor, com largura proporcional.
- **4**: a Tela preenche totalmente o divisor.
- **5 a 100**: equivale ao percentual de zoom da Tela propriamente dito.

A habilitação de rolagem pode assumir os seguintes valores:

- **0**: desabilita a rolagem.
- **1**: habilita a rolagem.

O parâmetro *Arg* permite repassar o valor especificado à Tela através do evento **OnPreShow**. Exemplo:

```
Sub Botao1_Click()  
    ' Quando clica no botao abre a Tela2 no frame Teste  
    ' e passa o valor 1 que será usado no evento OnPreShow  
    Application.Item("_top").  
        OpenScreen "Tela2?100?0", "Isso é um teste."  
End Sub  
  
Sub Tela2_OnPreShow(vArg)  
    ' A caixa de mensagem vai mostrar a  
    ' frase "Isso é um Teste."  
    MsgBox vArg  
End Sub
```

10.4.1.8 Refresh

Refresh(*Force*)

O método **Refresh** permite forçar o redesenho do conteúdo de uma Tela ou divisor. Ele deve ser usado em scripts de Viewer com processamento massivo (por exemplo, laços), ou em chamadas de métodos que demandem muito tempo e exijam indicações visuais ao usuário do progresso do processo.

Devido ao redesenho em geral ser uma operação pesada, a versão padrão do método

Refresh (sem parâmetros) é otimizada de forma a ignorar pedidos de redesenho muito próximos, incluindo redesenho do próprio E3. Este comportamento padrão é ideal para indicações de progresso em laços, onde muitos redeseños são feitos em seqüência. O parâmetro **Force** desabilita esta otimização, garantindo que para cada chamada do método **Refresh** seja feito um redesenho. Porém, ao utilizar-se esta opção, o método **Refresh** não pode ser chamado repetidamente, como dentro de um laço, por exemplo. Exemplo:

```
Sub CommandButton1_Click()  
    ' Desenha uma barra de progresso de uma operação  
    while i < 31  
        Application.Item("_top").  
            .HorizontalPercentFill = (i / 30) * 100  
        Frame.Refresh True  
        ' <-- alguma operação demorada -->  
        i = i + 1  
    wend  
End Sub
```

10.4.1.9 RestoreFrame

RestoreFrame()

Permite restaurar a janela do quadro para o seu tamanho original. Exemplo:

```
Sub CommandButton1_Click()  
    Application.Item("_top").RestoreFrame()  
End Sub
```

10.4.1.10 SetForegroundWnd

SetForegroundWnd()

O método **SetForegroundWnd** ativa e dá o foco para a janela do Viewer. Este método é útil quando se deseja chamar a atenção do operador para algum evento ocorrido, quando a janela do Viewer encontra-se escondida ou minimizada. Exemplo:

```
Sub CommandButton1_Click()  
    Application.Item("_top").SetForegroundWnd()  
End Sub
```

10.4.1.11 setFrameOptions

SetFrameOptions(*Title, Flags*)

Utilizado para configurar o título do frame na janela e o estilo da janela. O parâmetro *Title* é um **String** que contém o título da janela. Este texto será mostrado se a propriedade **Caption** da Tela estiver vazia.

O parâmetro *Flags* especifica o estilo da janela. Se este parâmetro for omitido, o valor padrão será -1. Este valor é usado para manter a configuração anterior da janela. Quando o valor especificado não for -1, pode-se modificar o estilo da janela especificando a soma dos valores das seguintes combinações:

Combinações possíveis para o parâmetro Flags

VALOR	DESCRIÇÃO
1	Habilita a barra de título na janela.
2	Habilita o botão Fechar na janela.
4	Habilita o botão Minimizar na janela.
8	Habilita o botão Maximizar na janela.
16	Habilita a borda na janela.
32	Especifica que a janela pode ser dimensionada. Para isso, é necessário que a janela em questão tenha borda.
64	Especifica que a janela pode ser movida.
256	Especifica que a janela ficará no topo da Tela.
512	Especifica que a janela será configurada no estilo Barra de Ferramentas.
1024	Desabilita os botões de objeto.

Exemplo:

```
Sub Tela_OnPreShow()  
    Application.Item("_top").SetFrameOptions("Tela de Alarmes", 114)  
End Sub
```

No exemplo anterior, o valor 114 (2 + 16 + 32 + 64) indica que a janela terá o botão **Fechar** habilitado (2), terá uma borda (16), poderá ser dimensionada (32) e poderá ser movida (64). O título da janela será "Tela de Alarmes".

Nos picks **Abrir Tela** e **Abrir Tela Modal** também é possível configurar o estilo da janela durante a edição, através da caixa de diálogo Estilo da Janela. Para mais informações veja o tópico **Picks**.

10.4.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto `_top`.

10.4.2.1 IsHTML

A propriedade **IsHTML** retorna True se o divisor contém códigos HTML inseridos no quadro. Caso contrário, retorna False.

10.4.2.2 Screen

 Retorna a Tela-pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

10.4.2.3 SplitBorder

Habilita ou desabilita a borda do divisor, e determina se a borda entre o divisor mandante e o divisor restante deve ser exibida em tempo de execução. Esta propriedade não tem efeito no divisor restante. O valor padrão desta propriedade é True.

10.4.2.4 SplitDockPosition

 Indica a posição do divisor desejado na Tela. As opções disponíveis são:

Opções disponíveis para SplitDockPosition

OPÇÃO	DESCRIÇÃO
0 - dockRemaining	Posiciona o divisor como restante, ou seja, ele ocupará o espaço que sobra na divisão horizontal ou vertical.
1 - dockTop	Posiciona o divisor como mandante, acima do restante.
2 - dockBottom	Posiciona o divisor como mandante, abaixo do restante.
3 - dockLeft	Posiciona o divisor como mandante, à esquerda do restante.
4 - dockRight	Posiciona o divisor como mandante, à direita do restante.

10.4.2.5 SplitLink

A A propriedade **SplitLink** contém um *link* que deve ser mostrado no divisor. É possível especificar uma Tela do projeto, um executável ou um *link* na Internet. No caso de Telas, é possível especificar um percentual de zoom e a habilitação das barras de rolagem usando a chave "?", como no modelo <nome-da-tela>?<zoom>?<barra-de-rolagem>, onde <nome-da-tela> é o nome da Tela a ser aberta; <zoom> é o percentual de zoom e <barra-de-rolagem> é **1** para habilitar ou **0** para não habilitar. Os parâmetros <zoom> e <barra-de-rolagem> são válidos apenas se o *link* indicado se tratar de uma Tela. Caso contrário, serão ignorados. Se o parâmetro <zoom> não for informado, é assumido 100%. Se o parâmetro <barra-de-rolagem> não for informado, é assumido 1, ou seja, habilitado. Exemplo:

```
Sub CommandButton1_Click()  
    Application.Item("_top").SplitLink = "Tela1?10?1"  
End Sub
```

10.4.2.6 SplitResizable

Determina se o divisor mandante pode ser redimensionado em tempo de execução. Esta propriedade não tem efeito no divisor restante. O valor padrão desta propriedade é True.

10.4.2.7 SplitValue

9 A propriedade **SplitValue** determina o valor que será atribuído ao divisor do quadro, que tanto pode ser % (percentagem), **hm** (HIMETRIC) ou **px** (pixels). Se a unidade for omitida, o valor será considerado em HIMETRIC. Exemplo:

```
Sub Divisor1_Click()  
    Application.Item("_top").SplitValue = 10  
End Sub
```

CAPÍTULO

11

Banco de Dados

Esta seção contém informações sobre métodos e propriedades do objeto Banco de Dados. Este objeto não possui eventos associados.

11.1 Métodos

Esta seção contém informações sobre os métodos do objeto Banco de Dados.

11.1.1 SetDBParameters

SetDBParameters(*ServerName, UserName, Password, DBName*)

String de conexão com o DB nas propriedades do objeto Banco de Dados. O parâmetro *ServerName* determina o nome do servidor. O parâmetro *UserName* determina o nome do usuário. O parâmetro *Password* determina a senha do login para se conectar ao banco. O parâmetro *DBName* é o nome do banco de dados utilizado no SQL Server. Para os demais bancos de dados esse parâmetro não é utilizado.

11.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Banco de Dados.

11.2.1 ConnectionActive

Mostra o status da conexão principal do Banco de Dados. Se a conexão estiver ativa, a propriedade tem o valor True. Caso contrário, tem o valor False.

11.2.2 EnableLocalCreation

Define se o Viewer criará a conexão do banco de dados localmente ou não. A conexão local será utilizada somente para a criação de consultas. Se a propriedade estiver habilitada (True), o Viewer tenta criar a conexão localmente, e se isso não for possível, busca a consulta pedida no servidor. O valor padrão desta propriedade é False.

11.2.3 EnableSynchronization

☑ Indica ao E3, se habilitada (True), que ele também deve fazer a gravação dos dados em uma segunda base de dados simultaneamente, para permitir uma maior segurança. Se a propriedade estiver habilitada e houver um servidor StandBy, o E3 faz a sincronização entre os bancos de dados dos dois servidores. O valor padrão desta propriedade é False, ou seja, a sincronia está desabilitada.

11.2.4 NetLibrary

📁 Configura o tipo de biblioteca de rede do Banco de Dados. As opções disponíveis são:

Opções disponíveis para NetLibrary

OPÇÃO	DESCRIÇÃO
0 - Default	Biblioteca do tipo padrão.
1 - NamedPipes	Biblioteca do tipo Named Pipes.
2 - Tcplp	Biblioteca do tipo Winsock TCP/IP.
3 - Spxlpx	Biblioteca do tipo SPX/IPX.
4 - BanyanVines	Biblioteca do tipo Banyan Vines.
5 - MultiProtocol	Biblioteca do tipo Multi-protocolo (RPC).

NOTA: a propriedade **NetLibrary** somente está disponível no Banco de Dados do tipo SQL Server.

11.2.5 ReconnectDelay

9 A propriedade **ReconnectDelay** determina o tempo de espera (em milissegundos) para o sistema fazer uma tentativa de restabelecer uma conexão perdida com o banco de dados. O valor padrão desta propriedade é 2000 (2 segundos).

11.2.6 SourceDatabase

A Caso o Banco de Dados seja Access, este é o nome do arquivo .mdb. Caso o Banco de Dados seja SQL Server, este é o nome do servidor SQL concatenado com o banco de dados escolhido, no formato Servidor/Banco de Dados. Caso o Banco de Dados seja Oracle, este é o nome da conexão criada. Esta propriedade é somente para leitura.

11.2.7 SourceType

 Indica o banco de dados utilizado pelo objeto. As opções disponíveis são:

- **0 - stOracle:** Banco de dados Oracle.
- **1 - stAccess:** Banco de dados Access.
- **2 - stSQLServer:** Banco de dados SQL Server.

Exemplo:

```
Sub Grupo1_Click()  
    op = Application.SelectMenu("Oracle|Access|SQLServer")  
    If op = 1 Then  
        Application.GetObject("ServidorDB").SourceType = 0  
        MsgBox "Está sendo usado um servidor Oracle."  
    ElseIf op = 2 Then  
        Application.GetObject("ServidorDB").SourceType = 1  
        MsgBox "Está sendo usado um servidor Access."  
    ElseIf op = 3 Then  
        Application.GetObject("ServidorDB").SourceType = 2  
        MsgBox "Está sendo usado um servidor SQL Server."  
    End If  
End Sub
```

11.2.8 TimeoutCommand

9 Contém o tempo de espera para qualquer operação do banco de dados, antes que o sistema gere um erro por *timeout*. O valor padrão é 180 (3 minutos).

11.2.9 TimeoutConnection

9 Contém o tempo de espera pela realização da conexão com o Banco de Dados, antes que o sistema gere um erro por *timeout*. O valor padrão é 15 segundos.

11.2.10 TotalFailedWrites

9 Contém o número total de erros em operações de banco de dados que aconteceram até o momento.

11.2.11 UserName

A Login utilizado para se conectar ao Banco de Dados. Esta propriedade é somente para leitura.

11.2.12 UseTransaction

Define se o DBServer irá utilizar as transações do banco de dados ou não. Se True, cada bloco de operações (200 operações) de DB (operações do Histórico, Storage, Formulas e Alarmes) serão executadas de uma vez só, ou seja, em uma transação.

CAPÍTULO

12

Alarmes

Esta seção contém informações sobre eventos, métodos e propriedades dos objetos Configuração de Alarmes, Áreas de Alarme, Fontes de Alarme e Servidor de Alarmes.

12.1 Configuração de Alarmes

O objeto Configuração de Alarmes não contém eventos, métodos ou propriedades específicos, somente gerais. Estes podem ser visualizados na seção **Eventos, Métodos e Propriedades Gerais dos Objetos**.

12.2 Áreas de Alarme

Esta seção contém informações sobre propriedades do objeto Área de Alarme. Este objeto não possui eventos nem métodos associados.

12.2.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Área de Alarme.

12.2.1.1 ActiveAlarms

9 Determina o número de alarmes ativos dentro da área do sistema. Se esta propriedade estiver com valor 0, os alarmes que estão desativados serão listados através da propriedade **ActiveNACKAlarms**. Esta propriedade é somente de leitura. O valor padrão desta propriedade é 0.

12.2.1.2 ActiveHighAlarms

9 Indica o número de alarmes ativos com severidade alta. Esta propriedade é somente de leitura.

12.2.1.3 ActiveHighNACKAlarms

9 Indica o número de alarmes não-reconhecidos com severidade alta. Esta propriedade é somente de leitura.

12.2.1.4 ActiveLowAlarms

9 Indica o número de alarmes ativos com severidade baixa. Esta propriedade é somente de leitura.

12.2.1.5 ActiveLowNACKAlarms

9 Indica o número de alarmes não-reconhecidos com severidade baixa. Esta propriedade é somente de leitura.

12.2.1.6 ActiveMedAlarms

9 Indica o número de alarmes ativos com severidade média. Esta propriedade é somente de leitura.

12.2.1.7 ActiveMedNACKAlarms

9 Indica o número de alarmes não-reconhecidos com severidade média. Esta propriedade é somente de leitura.

12.2.1.8 ActiveNACKAlarms

9 Indica o número de alarmes que estão sem reconhecimento dentro da área. Esta propriedade é somente de leitura.

12.2.1.9 Alarm

Estabelece a existência de alarmes ativos dentro da área. Se a opção estiver True, o sistema tem pelo menos um alarme ativo dentro da área, e a propriedade **ActiveAlarms** fará a leitura no sistema, indicando a quantidade de alarme ativos. Caso contrário, a propriedade **ActiveNACKAlarms** fará a leitura dos alarmes sem reconhecimento. Esta propriedade é somente de leitura.

12.2.1.10 AlarmVerify

☑ Habilita a verificação de todos os alarmes dentro da área. Depois de habilitada esta verificação (True), se a propriedade **ActiveAlarms** possuir valor maior que 0, o sistema verifica tanto os alarmes ativos quanto os sem reconhecimento, listando estes últimos pela propriedade **ActiveNACKAlarms**. Esta propriedade é útil para evitar-se o efeito de avalanche de alguns sistemas, onde o acontecimento de um evento gera uma grande quantidade de alarmes correlacionados.

12.3 Fontes de Alarme

Esta seção contém informações sobre métodos e propriedades comuns ao objeto Fonte de Alarme. Este objeto não possui eventos comuns associados.

NOTA: Quando um objeto Fonte de Alarme é desabilitado ou desativado, as propriedades **ActiveNACKAlarm**, **Alarm**, **CurrentSeverity**, **CurrentSubConditionName**, **FormattedValue**, **RawAlarm** e **Value** assumem seus valores padrão, indicando que a Fonte de Alarme não está vinculada a uma mensagem de alarme ativa. Neste caso, os valores das propriedades **Alarm** e **ActiveNACKAlarm** são propagados para os contadores das áreas superiores.

Para que as propriedades citadas recebam um valor diferente do padrão, é necessário que as seguintes condições estejam presentes:

- A Fonte de Alarme deve estar habilitada e ativada.
- As **Áreas de Alarme** (e todos os objetos hierarquicamente superiores) devem estar habilitadas e ativadas.
- O **Servidor de Alarmes** deve estar ativado.

12.3.1 Métodos Comuns

Esta seção contém informações sobre os métodos comuns aos objetos Fonte de Alarme.

12.3.1.1 Ack

Ack(ActorID)

Executa o reconhecimento de um alarme configurado no objeto Fonte de Alarmes. O método retorna um booleano que indica o sucesso ou não da operação. O parâmetro *ActorID* informa o nome do usuário responsável pelo reconhecimento dos alarmes.

12.3.1.2 GetAlarm

GetAlarm()

Retorna um objeto que permite o acesso às configurações específicas de cada tipo de alarme. Isto permite verificar ou modificar em tempo de execução as propriedade de um determinado alarme. Dependendo do tipo de alarme, o método retornará as seguintes propriedades:

- **Alarme Digital:** Responsável pelas configurações de alarme digital.

Propriedades do Alarme Digital

ITEM	DESCRIÇÃO
 DigitalReturnMessageText	Mensagem de retorno do alarme digital.
<input checked="" type="checkbox"/> Digital	Habilita ou desabilita a verificação de alarme digital.
 DigitalLimit	Limite para o alarme digital.
 DigitalMessageText	Texto da mensagem de alarme digital.
 DigitalSeverity	Severidade do alarme digital. Conjunto de valores: <ul style="list-style-type: none">• 0: Alta• 1: Média• 2: Baixa
<input checked="" type="checkbox"/> DigitalAckRequired	Necessidade de reconhecimento para este tipo de alarme (digital).

- **Alarme Analógico:** Responsável pelas configurações do alarme analógico. Propriedades deste objeto (tem quatro níveis de alarme):

Propriedades do Alarme Analógico

ITEM	DESCRIÇÃO
9 LevelDeadBand	Banda morta para o alarme de níveis.
A LevelReturnMessageText	Mensagem de retorno de alarme.
ALARME LOLO (MUITO BAIXO)	
ITEM	DESCRIÇÃO
<input checked="" type="checkbox"/> LoLo	Habilita ou desabilita a verificação de alarme Muito Baixo.
9 LoLoLimit	Limite para o nível de alarme Muito Baixo.
A LoLoMessageText	Texto da mensagem de alarme Muito Baixo.
 LoLoSeverity	<p>Importância do nível de alarme Muito Baixo. Conjunto de valores:</p> <ul style="list-style-type: none"> • 0: Alta • 1: Média • 2: Baixa
<input checked="" type="checkbox"/> LoLoAckRequired	Necessidade de reconhecimento para este nível de alarme (Muito Baixo).
ALARME LO (BAIXO)	
ITEM	DESCRIÇÃO
<input checked="" type="checkbox"/> Lo	Habilita ou desabilita a verificação de alarme Baixo.
9 LoLimit	Limite para o nível de alarme Baixo.
A LoMessageText	Texto da mensagem de alarme Baixo.
 LoSeverity	<p>Importância do nível de alarme Baixo. Conjunto de valores:</p> <ul style="list-style-type: none"> • 0: Alta • 1: Média • 2: Baixa

ITEM	DESCRIÇÃO
<input checked="" type="checkbox"/> LoAckRequired	Necessidade de reconhecimento para este nível de alarme (Baixo).
ALARME HI (ALTO)	
ITEM	DESCRIÇÃO
<input checked="" type="checkbox"/> Hi	Habilita ou desabilita a verificação de alarme Alto.
9 HiLimit	Limite para o nível de alarme Alto.
A HiMessageText	Texto da mensagem de alarme Alto.
 HiSeverity	<p>Importância do nível de alarme Alto. Conjunto de valores:</p> <ul style="list-style-type: none"> • 0: Alta • 1: Média • 2: Baixa
<input checked="" type="checkbox"/> HiAckRequired	Necessidade de reconhecimento para este nível de alarme (Alto).
ALARME HIHI (MUITO ALTO)	
ITEM	DESCRIÇÃO
<input checked="" type="checkbox"/> HiHi	Habilita ou desabilita a verificação de alarme Muito Alto.
9 HiHiLimit	Limite para o nível de alarme Muito Alto.
A HiHiMessageText	Texto da mensagem de alarme Muito Alto.
 HiHiSeverity	<p>Importância do nível de alarme Muito Alto. Conjunto de valores:</p> <ul style="list-style-type: none"> • 0: Alta • 1: Média • 2: Baixa
<input checked="" type="checkbox"/> HiHiAckRequired	Necessidade de reconhecimento para este nível de alarme (Muito Alto).

- **Alarme Taxa de Variação:** Responsável por configurações de alarme de taxa de variação.

Propriedades do Alarme Taxa de Variação

ITEM	DESCRIÇÃO
A ROCReturnMessageText	Mensagem de retorno da taxa de mudança.
<input checked="" type="checkbox"/> ROC	Habilita ou desabilita a verificação de alarme de taxa de variação.
9 ROCLimit	Limite para o alarme de taxa de mudança. Para o alarme ocorrer, basta que o valor do tag associado ultrapasse esse valor em um segundo.
A ROCMessageText	Texto da mensagem de alarme de taxa de mudança.
9 ROCSeverity	Importância do alarme de taxa de mudança. Conjunto de valores: <ul style="list-style-type: none"> • 0: Alta • 1: Média • 2: Baixa
<input checked="" type="checkbox"/> ROCAckRequired	Necessidade de reconhecimento para este tipo de alarme (taxa de mudança).

- **Alarme Banda Morta:** Responsável por configurações de alarme de banda morta.

Propriedades do Alarme Banda Morta

ITEM	DESCRIÇÃO
9 DeadBandSetPoint	Limite de banda morta do alarme. Cada vez que o valor do tag associado ultrapassar o valor desta propriedade para mais ou para

ITEM	DESCRIÇÃO
	menos do valor de DeadBandLimit , o alarme ocorrerá.
A DeadBandReturnMessageText	Mensagem de retorno de alarme de banda morta.
<input checked="" type="checkbox"/> DeadBand	Habilita ou desabilita a verificação de alarme de banda morta.
9 DeadBandLimit	Limite para o alarme de banda morta.
A DeadBandMessageText	Texto da mensagem de alarme de banda morta.
9 DeadBandSeverity	<p>Importância do alarme de banda morta. Conjunto de valores:</p> <ul style="list-style-type: none"> • 0: Alta • 1: Média • 2: Baixa
<input checked="" type="checkbox"/> DeadBandAckRequired	Necessidade de reconhecimento para este tipo de alarme (banda morta).

Exemplo:

```

Sub Botao1_Click()
Dim val
' Quando clica no botao muda o nivel de alarme Lo do
' AlarmSource BatteryLevel
Application.GetObject("ConfigAlarmes1.Area1.BatteryLevel")_
    .GetAlarm().LoLimit = 10.2
End Sub

```

NOTA: as propriedades relativas a cada tipo de alarme podem ser acessadas diretamente por scripts e associações, bem como visualizadas na Lista de Propriedades do objeto, não mais sendo obrigatória sua edição via método **GetAlarm**.

12.3.2 Propriedades Comuns

Esta seção contém informações sobre as propriedades comuns aos objetos Fonte de Alarme.

12.3.2.1 ActiveNACKAlarm

Se ajustada para True, indica que a Fonte não foi reconhecida desde a última ativação. Esta propriedade é somente de leitura. O valor padrão desta propriedade é False.

12.3.2.2 Alarm

Se ajustada para True, indica a condição ativa do alarme. O valor padrão desta propriedade é False.

12.3.2.3 AlarmVerify

Se ajustada para True, habilita a verificação da fonte de alarme (ou seja, a geração do alarme).

12.3.2.4 CurrentSeverity

 Indica a última severidade do alarme ativo, a saber:

- **0**: Alta
- **1**: Média
- **2**: Baixa

O valor padrão desta propriedade é -1, indicando que a Fonte de Alarme não está ativa.

12.3.2.5 CurrentSubConditionName

A Determina o nome da última condição de alarme ativa. As opções disponíveis para esta propriedade são:

Opções disponíveis para CurrentSubConditionName

OPÇÃO	DESCRIÇÃO
LOLO	Alarme Analógico na faixa LOLO.
LO	Alarme Analógico na faixa LO.
HI	Alarme Analógico na faixa HI.
HIHI	Alarme Analógico na faixa HIHI.
BM	Alarme de Banda Morta.
TV	Alarme Taxa de Variação.
DIG	Alarme Digital.

O valor padrão desta propriedade é uma **String** vazia.

12.3.2.6 Delay

9 Especifica um tempo de atraso para o alarme (em ms), tanto na entrada quanto na saída da condição. Quando essa propriedade estiver em 0 (que será sempre o valor padrão), não é aplicado nenhum atraso. Quando for diferente de 0, o alarme só será ativado ou desativado se permanecer na mesma condição por um tempo maior ou igual ao especificado.

12.3.2.7 DoubleAckRequired

Quando configurada como True, indica que o alarme pode ser reconhecido tanto quando está ativo como quando passa a inativo, ou seja, pode ser reconhecido duplamente. Quando configurada como False, indica que o alarme só pode ser reconhecido uma vez, e ao ser reconhecido sai da lista de alarmes. Alarmes que não necessitem reconhecimento (propriedade **AckRequired**) não possibilitam esta personalização. Aplicações anteriores à versão 2.5 vêm com esta propriedade

configurada como False.

12.3.2.8 Event

☑ Quando configurada como True, indica que o alarme é do tipo evento. Um alarme do tipo evento, ao ser ativado, é reconhecido pelo usuário "System". Portanto, ao ser reconhecido nada acontece (não tem efeito algum), assim como não incrementa o número de alarmes ativos nem o número de alarmes não-reconhecidos. Não pode ser modificada em tempo de execução.

12.3.2.9 Format

A propriedade **Format** especifica que tipo de formato será atribuído ao objeto. Permite mudar a forma como os dados são apresentados sem mudar o valor que existe por trás deles. Esta propriedade pode ser editada manualmente ou configurada através da janela de formatação. Seu uso é semelhante aos formatadores usados nas planilhas de dados, seguindo a mesma sintaxe básica. São suportados os seguintes tipos de dados:

Tipos de dados suportados por Format

DADOS	DESCRIÇÃO
Numérico	Saída decimal, científica, hexadecimal, binária e octal.
Texto	Textos em geral.
Booleano	Valores booleanos.
Data/Hora	Calendário Gregoriano.

12.3.2.10 FormattedValue

A contém o valor do alarme conforme a propriedade **Value**, formatado de acordo com a propriedade **Format**. Esta propriedade é somente de leitura. O valor padrão desta propriedade é Null.

12.3.2.11 RawAlarm

Indica se o alarme deve estar ativo, independentemente da espera. Quando a espera for igual a 0 (zero), o valor de **RawAlarm** será o mesmo da propriedade **Alarm**. Esta propriedade é somente de leitura. O valor padrão desta propriedade é False.

12.3.2.12 Source

A Contém a expressão que deverá ser avaliada para determinar se o alarme deve ocorrer ou não.

12.3.2.13 Value

V Contém o valor do que foi avaliado para determinar se o alarme ocorre ou não. O valor padrão desta propriedade é Null.

12.3.3 Fonte de Alarme Analógico

Esta seção contém informações sobre propriedades do objeto Fonte de Alarme Analógico. Este objeto não possui eventos nem métodos associados.

12.3.3.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Fonte de Alarme Analógico.

12.3.3.1.1 Hi

Habilita ou desabilita a verificação de alarmes do tipo Hi.

12.3.3.1.2 HiAckRequired

Indica que o alarme do tipo Hi requer reconhecimento.

12.3.3.1.3 HiHi

Habilita ou desabilita a verificação de alarmes do tipo HiHi.

12.3.3.1.4 HiHiAckRequired

Indica que o alarme do tipo HiHi requer reconhecimento.

12.3.3.1.5 HiHiLimit

9 Indica o nível em que o alarme HiHi será ativado.

12.3.3.1.6 HiHiMessageText

A Ajusta a mensagem de texto do limite do alarme de tipo HiHi.

12.3.3.1.7 HiHiSeverity

 Indica o nível de importância do alarme de tipo HiHi. As opções são:

- **0**: Alta
- **1**: Média
- **2**: Baixa

12.3.3.1.8 HiLimit

9 Indica o nível em que o alarme Hi será ativado.

12.3.3.1.9 HiMessageText

A Ajusta a mensagem de texto do limite do alarme de tipo Hi.

12.3.3.1.10 HiSeverity

 Indica o nível de importância do alarme de tipo Hi. As opções são:

- **0**: Alta
- **1**: Média
- **2**: Baixa

12.3.3.1.11 LevelDeadBand

9 Banda morta para os limites do nível do alarme.

12.3.3.1.12 LevelReturnMessageText

A Ajusta a mensagem de retorno do nível do alarme.

12.3.3.1.13 Lo

Habilita ou desabilita a verificação de alarmes do tipo Lo.

12.3.3.1.14 LoAckRequired

Indica que o alarme do tipo Lo requer reconhecimento.

12.3.3.1.15 LoLimit

9 Indica o nível em que o alarme Lo será ativado.

12.3.3.1.16 LoLo

Habilita ou desabilita a verificação de alarmes do tipo LoLo.

12.3.3.1.17 LoLoAckRequired

 Indica que o alarme do tipo LoLo requer reconhecimento.

12.3.3.1.18 LoLoLimit

 Indica o nível em que o alarme LoLo será ativado.

12.3.3.1.19 LoLoMessageText

 Ajusta a mensagem de texto do limite do alarme de tipo LoLo.

12.3.3.1.20 LoLoSeverity

 Indica o nível de importância do alarme de tipo LoLo. As opções são:

- **0:** Alta
- **1:** Média
- **2:** Baixa

12.3.3.1.21 LoMessageText

 Ajusta a mensagem de texto do limite do alarme de tipo Lo.

12.3.3.1.22 LoSeverity

 Indica o nível de importância do alarme de tipo Lo. As opções são:

- **0:** Alta
- **1:** Média
- **2:** Baixa

12.3.4 Fonte de Alarme Digital

Esta seção contém informações sobre propriedades do objeto Fonte de Alarme Digital. Este objeto não possui eventos nem métodos associados.

12.3.4.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Fonte de Alarme Analógico.

12.3.4.1.1 Digital

Habilita ou desabilita a verificação de alarme digital.

12.3.4.1.2 DigitalAckRequired

Necessidade de reconhecimento para o alarme digital.

12.3.4.1.3 DigitalLimit

9 Limite para o alarme digital.

12.3.4.1.4 DigitalMessageText

A Texto da mensagem de alarme digital.

12.3.4.1.5 DigitalReturnMessageText

A Mensagem de retorno do alarme digital.

12.3.4.1.6 DigitalSeverity

 Severidade do alarme digital. Conjunto de valores:

- **0**: Alta
- **1**: Média
- **2**: Baixa

12.3.5 Fonte de Alarme Banda Morta

Esta seção contém informações sobre propriedades do objeto Fonte de Alarme Banda Morta. Este objeto não possui eventos nem métodos associados.

12.3.5.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Fonte de Alarme Analógico.

12.3.5.1.1 DeadBand

Habilita ou desabilita a verificação de alarme de banda morta.

12.3.5.1.2 DeadBandAckRequired

Necessidade de reconhecimento para o alarme de banda morta.

12.3.5.1.3 DeadBandLimit

9 Limite para o alarme de banda morta.

12.3.5.1.4 DeadBandMessageText

A Texto da mensagem de alarme de banda morta.

12.3.5.1.5 DeadBandReturnMessageText

A Mensagem de retorno de alarme de banda morta.

12.3.5.1.6 DeadBandSetpoint

A Limite de banda morta do alarme. Cada vez que o valor do tag associado ultrapassar o valor desta propriedade para mais ou para menos do valor de **DeadBandLimit**, o alarme ocorrerá.

12.3.5.1.7 DeadBandSeverity

9 Importância do alarme de banda morta. Conjunto de valores:

- **0**: Alta
- **1**: Média
- **2**: Baixa

12.3.6 Fonte de Alarme Taxa de Variação

Esta seção contém informações sobre propriedades do objeto Fonte de Alarme Taxa de Variação. Este objeto não possui eventos nem métodos associados.

12.3.6.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Fonte de Alarme Taxa de Variação.

12.3.6.1.1 ROC

Habilita ou desabilita a verificação de alarme de taxa de variação.

12.3.6.1.2 ROCAckRequired

Necessidade de reconhecimento para o alarme de taxa de variação.

12.3.6.1.3 ROCLimit

9 Limite para o alarme de taxa de variação. Para o alarme ocorrer, basta que o valor do tag associado ultrapasse esse valor em um segundo.

12.3.6.1.4 ROCMessageText

A Texto da mensagem do alarme de taxa de variação.

12.3.6.1.5 ROCReturnMessageText

A Mensagem de retorno do alarme de taxa de variação.

12.3.6.1.6 ROCSeverity

 Importância do alarme de taxa de variação. Conjunto de valores:

- **0**: Alta
- **1**: Média
- **2**: Baixa

12.4 Servidor de Alarmes

Esta seção contém informações sobre métodos e propriedades do objeto Servidor de Alarme. Este objeto não possui eventos associados.

12.4.1 Métodos

Esta seção contém informações sobre os métodos do objeto Servidor de Alarme.

12.4.1.1 AckAllAlarms

AckAllAlarms(*ActorID*)

Executa o reconhecimento de todos os alarmes do servidor, independente da área. O método retorna um booleano que indica o sucesso ou não da operação.

O parâmetro *ActorID* informa o nome do usuário responsável pelo reconhecimento dos alarmes. Exemplo:

```
Sub Botao1_Click()  
    ' Quando clica no botao reconhece todos os alarmes  
    Application.GetObject("ServidorAlarmes1")._  
    AckAllAlarms(Application.User)  
End Sub
```

12.4.1.2 AckArea

AckArea(Area, ActorID)

Executa o reconhecimento dos alarmes em uma determinada área. O método retorna um booleano que indica o sucesso ou não da operação.

O parâmetro *Area* especifica o nome da(s) área(s) cujos alarmes serão reconhecidos, através da comparação entre os trechos iniciais dos nomes. Por exemplo, AckArea ("ANA") reconheceria alarmes das áreas "ANALOGICOS", "ANA.AREA2", etc. Se o valor deste parâmetro for vazio, o método terá o mesmo comportamento de **AckAllAlarms**. O parâmetro *ActorID* informa o nome do usuário responsável pelo reconhecimento dos alarmes. Exemplo:

```
Sub Botao1_Click()  
    'Quando clica no botão reconhece os alarmes da Area1  
    Application.GetObject("ServidorAlarmes1").AckArea "Area1", _  
        Application.User  
End Sub
```

12.4.1.3 LogTrackingEvent

LogTrackingEvent(Message, ActorID, Area, Severity, EventTime, Source, EventCategory, EventType, UserFields, AlarmSourceName, FullAlarmSourceName)

Simula um evento ou alarme e o envia direto para o banco de dados do servidor de alarmes, sem passar pelo E3Alarm. Por isso, este evento não pode ser visto no E3Alarm, nem tampouco o alarme pode ser reconhecido.

Cada parâmetro do método permite especificar o valor do campo de mesmo nome no evento. Os campos do evento são preenchidos de acordo com os parâmetros do método:

Parâmetros do método LogTrackingEvent

NOME	DESCRIÇÃO
Message	Parâmetro texto que especifica o conteúdo do campo de Mensagem do evento. Se omitido, assume uma string vazia.
ActorID	Parâmetro texto que especifica o conteúdo do campo Operador do evento. Se omitido, assume o valor "System".
Area	Parâmetro texto que especifica o conteúdo do campo Área do evento. Se omitido, assume uma string vazia.

NOME	DESCRIÇÃO
Severity	Parâmetro numérico que determina a severidade do evento. Se omitido, assume o valor 0, ou seja, severidade alta.
EventTime	Especifica o <i>timestamp</i> do evento. Se omitido, assume o valor do <i>timestamp</i> do momento da chamada do método.
Source	Parâmetro texto que especifica o conteúdo do campo Fonte do evento. Se omitido, assume uma string vazia.
EventCategory	Parâmetro texto que especifica a categoria do evento. Se omitido, assume uma string vazia.
EventType	Parâmetro texto que especifica o tipo do evento. Se omitido, assume o valor "Tracking".
UserFields	Parâmetro array de quatro posições ou mais onde cada posição assume o valor do campo especificado pelo usuário.
AlarmSourceName	Parâmetro texto que especifica o nome da fonte de alarme. Se omitido, assume uma string vazia.
FullAlarmSourceName	Parâmetro texto que especifica o caminho completo da fonte de alarme, incluindo a área, a configuração do alarme e as possíveis pastas onde o alarme está inserido. Se omitido, assume uma string vazia.

Os demais campos do evento não podem ser especificados e sempre assumem os valores a seguir:

- **Valor (CurrentValue):** 0.0
- **Qualidade (Quality):** ""
- **Condição Ativa (ConditionActive):** 0 (False)
- **Nome da Condição (ConditionName):** ""
- **Nome da Subcondição (SubConditionName):** ""

- **Reconhecido (Acked):** 1 (True)
- **Precisa Ack (AckRequired):** 0 (False)
- **Habilitado (Enabled):** 1 (True)
- **EventTimeUTC:** *Sempre igual a **EventTime** (como em eventos de alarme)
- **Máscara (ChangeMask):** 0
- **Cookie:** 0

NOTA: O método falha se a opção **Guarde alarmes no banco de dados** estiver desmarcada, ou quando falha o armazenamento no banco de dados.

Exemplo:

```
Sub CommandButton1_Click()
    ' No parâmetro UserFields, para cada elemento do array,
    ' será mostrado o valor atribuído a ele.
    Application.GetObject("ServidorAlarmes1").LogTrackingEvent_
    "Clicou no botão", Application.User, "Operação", 2, , _
    "Botão1", , , array(1, 2, "a", "b")
End Sub
```

12.4.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Servidor de Alarme.

12.4.2.1 ActiveAlarms

9 Determina o número total de alarmes ativos do sistema. Esta propriedade é somente de leitura.

12.4.2.2 ActiveHighAlarms

9 Indica o número de alarmes ativos com severidade alta. Esta propriedade é somente de leitura.

12.4.2.3 ActiveHighNACKAlarms

9 Indica o número de alarmes não-reconhecidos com severidade alta. Esta propriedade é somente de leitura.

12.4.2.4 ActiveLowAlarms

9 Indica o número de alarmes ativos com severidade baixa. Esta propriedade é somente de leitura.

12.4.2.5 ActiveLowNACKAlarms

9 Indica o número de alarmes não-reconhecidos com severidade baixa. Esta propriedade é somente de leitura.

12.4.2.6 ActiveMedAlarms

9 Indica o número de alarmes ativos com severidade média. Esta propriedade é somente de leitura.

12.4.2.7 ActiveMedNACKAlarms

9 Indica o número de alarmes não-reconhecidos com severidade média. Esta propriedade é somente de leitura.

12.4.2.8 ActiveNACKAlarms

9 Indica o número total de alarmes que estão sem reconhecimento no sistema (ativos ou não). Esta propriedade é somente de leitura.

12.4.2.9 BackupDiscardInterval

9 Indica a quantidade de unidades de tempo durante o qual os dados de backup serão mantidos na tabela principal e na tabela de backup, até serem descartados (por exemplo, para manter os dados por 24 meses na tabela principal, e mais seis meses na tabela de backup, o valor da propriedade deve ser de 30 meses). Esta propriedade trabalha em conjunto com a propriedade **BackupDiscardTimeUnit**. O valor padrão desta propriedade é 12 (doze unidades de tempo indicadas em **BackupDiscardTimeUnit**).

NOTA: O tempo total indicado pela combinação das propriedades **BackupDiscardInterval** e **BackupDiscardTimeUnit** deve ser maior que o tempo indicado pelas propriedades **DiscardInterval** e **DiscardTimeUnit**.

12.4.2.10 BackupDiscardTimeUnit

 A propriedade **BackupDiscardTimeUnit** indica a unidade de tempo em que os dados de backup ficarão armazenados até serem descartados. As opções disponíveis são:

- **0 - dtHour:** horas.
- **1 - dtDay:** dias.
- **2 - dtMonth:** meses (padrão).
- **3 - dtMinute:** minutos.

Esta propriedade trabalha em conjunto com a propriedade **BackupDiscardInterval**.

12.4.2.11 DataSource

A Define o objeto Banco de Dados que será utilizado para gravação dos dados dos alarmes. O valor padrão desta propriedade é vazio, ou seja, não há nenhum Banco de Dados para armazenar os dados.

12.4.2.12 DiscardInterval

9 Esta propriedade trabalha em conjunto com a propriedade **DiscardTimeUnit**. A propriedade **DiscardInterval** indica o intervalo de tempo durante o qual os dados do Histórico ficarão armazenados na tabela do banco de dados, até serem descartados. O valor padrão desta propriedade é 1 (uma unidade de tempo indicada em **DiscardTimeUnit**). Se esta propriedade for configurada com um valor menor ou igual ao valor da propriedade **BackupDiscardInterval**, o E3 automaticamente ajusta o valor de **BackupDiscardInterval** como o dobro do valor de **DiscardInterval**. Esta propriedade pode ser modificada em tempo de execução.

12.4.2.13 DiscardTimeUnit

 Esta propriedade trabalha em conjunto com a propriedade **DiscardInterval**. A propriedade **DiscardTimeUnit** indica a unidade de tempo em que os dados da tabela ficarão armazenados até serem descartados. As opções disponíveis são:

- **0 - dtHour**: horas.
- **1 - dtDay**: dias.
- **2 - dtMonth**: meses (padrão).
- **3 - dtMinute**: minutos.

12.4.2.14 EnableBackupTable

Cria uma tabela de backup onde os dados descartados permanecerão por segurança. Se configurada para True, a tabela será criada; do contrário, não haverá tabela de backup. O valor padrão desta propriedade é True.

12.4.2.15 EnableDiscard

Indica descarte dos dados dos alarmes após um certo tempo. Se configurada para False, os dados serão armazenados indefinidamente na tabela; do contrário, serão descartados após determinado tempo. O valor padrão dessa propriedade é False.

12.4.2.16 Logging

Cria o registro das informações dos alarmes no banco de dados especificado pela propriedade **DataSource**. Se configurado para False, o registro não é criado; do contrário, o registro é feito. O valor padrão desta propriedade é False.

12.4.2.17 TableName

A Define um nome para a tabela de alarmes. O valor padrão é "Alarms". Pode ser modificada em tempo de execução e seu efeito será imediato.

12.4.2.18 VerificationInternal

9 Esta propriedade trabalha em conjunto com a propriedade **VerificationUnit** para controlar o intervalo de tempo que o E3 verifica a antiguidade dos dados, para depois descartá-los. O valor padrão desta propriedade é 1 (uma unidade de tempo indicada em **VerificationUnit**).

12.4.2.19 VerificationUnit

Esta propriedade trabalha em conjunto com a propriedade **VerificationInternal**. A propriedade **VerificationUnit** indica a unidade de tempo em que será feita a verificação para descarte dos dados. As opções disponíveis são:

- **0 - dtHour**: horas.
- **1 - dtDay**: dias.
- **2 - dtMonth**: meses (padrão).
- **3 - dtMinute**: minutos.

CAPÍTULO

13

E3Alarm

Esta seção contém informações sobre eventos, métodos e propriedades do objeto E3Alarm.

13.1 Eventos

Esta seção contém informações sobre os eventos do objeto E3Alarm.

13.1.1 KeyPress

KeyPress(*KeyAscii*)

Ocorre quando o E3Alarm tem o foco de teclado e o usuário pressiona uma tecla que corresponde a um caractere que pode ser mostrado em tela (uma tecla ANSI, de código indicado na variável *KeyAscii*). Ou seja, o evento ocorre quando alguma das teclas a seguir são pressionadas:

- Qualquer caractere do teclado que possa ser impresso.
- Tecla CTRL combinada com qualquer caractere do alfabeto padrão.
- Tecla CTRL combinada com qualquer caractere especial.
- Tecla BACKSPACE.
- Tecla ESC.

Este evento não ocorre nas seguintes condições:

- Pressionando a tecla TAB.
- Pressionando a tecla ENTER.
- Pressionando a tecla DEL (essa tecla não é ANSI).

- Pressionando as setas do teclado.
- Quando uma tecla faz com que o foco vá de um objeto a outro.

Enquanto um usuário pressiona uma tecla que produz um código ANSI, o E3Alarm recebe os eventos **KeyDown** e **KeyPress** repetidamente. Quando o usuário libera a tecla, o evento **KeyUp** ocorre. Para monitorar o estado físico do teclado ou manipular teclas não reconhecidas pelo evento **KeyPress** (como teclas de função, navegação, etc.), utiliza-se os eventos **KeyDown** e **KeyUp**.

13.1.2 MouseMove

MouseMove()

Ocorre quando o ponteiro do mouse se move sobre o objeto E3Alarm.

13.2 Métodos

Esta seção contém informações sobre os métodos do objeto E3Alarm.

13.2.1 AckAll

AckAll([Operator])

Permite reconhecer globalmente todos os alarmes. *Operator* é um string opcional que indica o nome do operador que reconheceu o alarme. Esse valor será mostrado na coluna **Operator** do E3Alarm. Se for omitido, será utilizado o usuário atual do Viewer, ou então "(Sem usuário)", caso não haja nenhum usuário logado. Para o reconhecimento em si é necessário que o usuário logado tenha permissão para reconhecer alarmes.

13.2.2 AckCurrentFilter

AckCurrentFilter([Operator])

Permite reconhecer todos os alarmes do filtro corrente. *Operator* é um string opcional que indica o nome do operador que reconheceu o alarme. Esse valor será mostrado na coluna **Operator** do E3Alarm. Se for omitido, será utilizado o usuário atual do Viewer, ou então "(Sem usuário)", caso não haja nenhum usuário logado. Para o reconhecimento em si é necessário que o usuário logado tenha permissão para reconhecer alarmes.

13.2.3 AckSelected

AckSelected([Operator])

Permite reconhecer os alarmes selecionados. Se não houver alarme selecionado no E3Alarm, o método falha. O usuário poderá reconhecer o alarme (neste caso, será inserido um novo registro no Banco de Dados indicando o reconhecimento), e no E3Alarm a linha correspondente indicará que ele foi reconhecido. *Operator* é um string opcional que indica o nome do operador que reconheceu o alarme. Esse valor será mostrado na coluna **Operator** do E3Alarm. Se for omitido, será utilizado o usuário atual do Viewer, ou então "(Sem usuário)", caso não haja nenhum usuário logado. Para o reconhecimento em si é necessário que o usuário logado tenha permissão para reconhecer alarmes.

13.2.4 GetFocusedEvent

GetFocusedEvent()

Este método retorna um objeto com as propriedades do evento selecionado (aquele que tem o foco atual) no E3Alarm, caso haja algum evento selecionado. Se nenhum evento estiver selecionado, o método retornará um objeto Nothing.

As propriedades do objeto retornado contém os valores dos campos do evento selecionado. O objeto contém uma cópia dos valores no momento da chamada do método, portanto se houver uma mudança no evento selecionado, as propriedades não serão atualizadas automaticamente, sendo necessário utilizar o método sempre que houver necessidade de obter informações atualizadas do evento selecionado. As propriedades do objeto retornado pelo método são as seguintes:

Propriedades do objeto retornado por GetFocusedEvent

NOME	DESCRIÇÃO
Acked	Informa se o alarme foi reconhecido ou não. Este campo pode assumir os valores 0 , não-reconhecido e 1 , reconhecido.
AckRequired	Determina o reconhecimento automático do alarme. Este campo apresenta os valores 0 , reconhecimento automático e 1 , reconhecimento manual.
AckTime	Registra a data/hora do E3 no momento em que o alarme é reconhecido, ou zero (30/12/1899), enquanto o alarme não for reconhecido. No caso de alarmes que não pedem reconhecimento, o campo assume a data/hora do E3 no momento em que o alarme fica ativo.

NOME	DESCRIÇÃO
ActorID	<p>Nome do operador que reconheceu o alarme. Pode ser:</p> <ul style="list-style-type: none"> • O usuário logado no Viewer, quando o reconhecimento é feito no E3Alarm (ou Sem usuário, se não há usuário logado). • System, quando o reconhecimento é automático (ou seja, para alarmes que pedem reconhecimento). • Um nome passado por Scripts (por exemplo, usando AckArea, AckAllAlarms, LogTrackingEvent do Servidor de Alarmes, ou Ack da Fonte de Alarmes). <p>O limite deste campo quando armazenado no Banco de Dados é de 50 caracteres.</p>
AlarmSourceName	<p>Registra o nome da fonte de alarme. O limite deste campo quando armazenado no Banco de Dados é de 100 caracteres. Por exemplo, FonteAlarme1.</p>
Area	<p>Para eventos de alarmes, é o nome da área a qual a fonte de alarmes pertence. Para outros eventos (por exemplo, usando o método LogTrackingEvent do Servidor de Alarmes), pode ser um texto definido pelo usuário. O limite deste campo quando armazenado no Banco de Dados é de 100 caracteres.</p>
ConditionActive	<p>Indica se a fonte de alarmes está em alarme. Este campo apresenta os estados 0, condição não-ativa e 1, condição ativa.</p>
ConditionName	<p>Nome da condição, se for um evento de alarme. Este campo pode apresentar os seguintes valores:</p> <ul style="list-style-type: none"> • Banda Morta: Fonte de alarme do tipo banda morta. • Digital: Fonte de alarme do tipo digital. • Nível: Fonte de alarme do tipo analógico. • TaxaDeVariação: Fonte de alarme do tipo taxa de variação. <p>Se o evento não for um alarme (por exemplo, usando o método LogTrackingEvent do Servidor de Alarmes), este valor será sempre uma string vazia. O limite deste campo quando armazenado no Banco de Dados é de 100 caracteres.</p>

NOME	DESCRIÇÃO
CurrentValue	Determina o valor da fonte de alarme (convertido para Double) no momento do evento. Para outros eventos (por exemplo, usando o método LogTrackingEvent), o valor será sempre zero (0). O limite deste campo quando armazenado no Banco de Dados é de 100 caracteres.
Enabled	Determina se a verificação do alarme está habilitada. Este campo apresenta os estados 0 , verificação da fonte de alarmes desabilitada e 1 , verificação da fonte de alarme habilitada.
EventCategory	<p>Categoria do evento. Para alarmes, este campo pode assumir os seguintes valores:</p> <ul style="list-style-type: none"> • Banda Morta: Fonte de alarme do tipo banda morta. • Digital: Fonte de alarme do tipo digital. • Nível: Fonte de alarme do tipo analógico. • TaxaDeVariação: Fonte de alarme do tipo taxa de variação. <p>Para outros eventos (por exemplo, usando o método LogTrackingEvent do Servidor de Alarmes), o campo pode assumir valores definidos pelo usuário. O limite deste campo quando armazenado no Banco de Dados é de 100 caracteres.</p>
EventCLSID	Identificador único para toda a vida de um alarme. Quando um alarme novo ocorre em uma Fonte, é gerado um novo número de EventCLSID; assim, ele irá manter esse mesmo CLSID no banco de dados enquanto não sair da lista de alarmes ativos e não-reconhecidos.
EventTime	Data/hora do valor da fonte do alarme no momento do evento.
EventTimeUTC	Data/hora do valor da fonte do alarme no momento do evento, com relação ao horário de Greenwich. O valor é o mesmo do campo EventTime , sendo mantido no E3 por questões de compatibilidade.
EventType	Tipo do evento. Para eventos de alarme, é sempre Condition . Para outros eventos, pode ser um texto definido pelo usuário, como por exemplo, usando o método LogTrackingEvent do Servidor de Alarmes (Tracking, Simple, etc.). O limite deste campo quando armazenado no Banco de Dados é de 100

NOME	DESCRIÇÃO
	caracteres.
FormattedValue	Mostra o valor formatado da fonte de alarme que vai para o seu evento. O limite deste campo quando armazenado no Banco de Dados é de 100 caracteres. NOTA: Este campo é somente de leitura.
FullAlarmSourceName	Registra o caminho completo da fonte de alarme, incluindo áreas, nome da configuração de alarmes e de possíveis pastas onde esta possa estar inserida. Por exemplo, Pasta1.ConfigAlarmes1.Area1.FonteAlarme1.
InTime	Registra a data/hora do valor no momento em que ele entra na condição de alarme.
Message	É o texto configurado na fonte de alarme, ou especificado por um outro evento (por exemplo, usando o método LogTrackingEvent do Servidor de Alarmes). O limite deste campo quando armazenado no Banco de Dados é de 200 caracteres.
OutTime	Registra a data/hora do valor no momento em que ele sai da condição de alarme; ou zero (30/12/1899), enquanto o alarme ainda não tenha saído da condição ativa.
Quality	<p>Qualidade do valor de fonte do alarme no momento do evento. Este campo assume os valores numéricos</p> <ul style="list-style-type: none"> • 0 - 63: qualidade ruim. • 64 - 127: qualidade incerta. • 128 - 191: valor não-definido. • 192 - 255: qualidade boa. <p>Se o evento não for um alarme (por exemplo, se ele estiver usando o método LogTrackingEvent do Servidor de Alarmes), este campo será igual a uma string vazia. Exemplo: Ruim (0); Incerta (64); ?? (128); Boa (192).</p>
Severity	É o valor de severidade configurado na fonte de alarmes. Este campo pode assumir os valores 0 , alta; 1 , média; 2 , baixa. Ele pode também assumir outro valor definido pelo usuário se for um evento, como por exemplo quando estiver usando o

NOME	DESCRIÇÃO
	método LogTrackingEvent .
Source	Para eventos de alarme, informa a expressão utilizada para avaliar as condições de alarme. O limite deste campo quando armazenado no Banco de Dados é de 100 caracteres. Por exemplo, Dados.TagDemo1.Value.
SubConditionName	<p>Nome da subcondição, se for um evento de alarme. Este campo pode assumir os valores</p> <ul style="list-style-type: none"> • BM: Alarme de Banda Morta. • DIG: Alarme Digital. • TV: Alarme Taxa de Variação. • LOLO: Alarme Analógico na faixa Muito Baixo. • LO: Alarme Analógico na faixa Baixo. • HI: Alarme Analógico na faixa Alto. • HIHI: Alarme Analógico na faixa Muito Alto. <p>Se o evento não for um alarme (por exemplo, se ele estiver usando o método LogTrackingEvent do Servidor de Alarmes), este campo será igual a uma string vazia. O limite deste campo quando armazenado no Banco de Dados é de 100 caracteres.</p>
UserField	Propriedade de leitura indexada cujo índice varia de 1 até 4, contendo os campos criados pelo usuário.

Exemplo:

```

Dim evt
Set evt = GetFocusedEvent()
If NOT(evt Is Nothing) Then
    Dim str
    str = str & "EventTime = " & evt.EventTime & Chr(13)
    str = str & "EventTimeUTC = " & evt.EventTimeUTC & Chr(13)
    str = str & "InTime = " & evt.InTime & Chr(13)
    str = str & "OutTime = " & evt.OutTime & Chr(13)
    str = str & "AckTime = " & evt.AckTime & Chr(13)
    str = str & "CurrentValue = " & evt.CurrentValue & Chr(13)
    str = str & "ActorID = " & evt.ActorID & Chr(13)
    str = str & "Area = " & evt.Area & Chr(13)
    str = str & "AlarmSourceName = " & evt.AlarmSourceName & Chr(13)

```

```

str = str & "FullAlarmSourceName = " & evt.FullAlarmSourceName & Chr(13)
str = str & "ConditionName = " & evt.ConditionName & Chr(13)
str = str & "EventCategory = " & evt.EventCategory & Chr(13)
str = str & "EventType = " & evt.EventType & Chr(13)
str = str & "Message = " & evt.Message & Chr(13)
str = str & "Quality = " & evt.Quality & Chr(13)
str = str & "Source = " & evt.Source & Chr(13)
str = str & "SubConditionName = " & evt.SubConditionName & Chr(13)
str = str & "FormattedValue = " & evt.FormattedValue & Chr(13)
str = str & "UserField(1) = " & evt.UserField(1) & Chr(13)
str = str & "UserField(2) = " & evt.UserField(2) & Chr(13)
str = str & "UserField(3) = " & evt.UserField(3) & Chr(13)
str = str & "UserField(4) = " & evt.UserField(4) & Chr(13)
str = str & "Severity = " & evt.Severity & Chr(13)
str = str & "Acked = " & evt.Acked & Chr(13)
str = str & "AckRequired = " & evt.AckRequired & Chr(13)
str = str & "ConditionActive = " & evt.ConditionActive & Chr(13)
str = str & "Enabled = " & evt.Enabled & Chr(13)
str = str & "EventCLSID = " & evt.EventCLSID & Chr(13)
MsgBox str
Else
MsgBox "Nenhum evento selecionado."
End If

```

13.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto E3Alarm.

NOTA: O E3 utiliza para definição de coordenadas e espessura o sistema HIMETRIC. Neste sistema, cada unidade lógica equivale a um milésimo de centímetro; ou seja, cada 1000 unidades equivalem a 1 centímetro. Assim, este é o padrão adotado na descrição das propriedades do E3, quando aplicável.

13.3.1 ActiveAlarms

9 Determina o número total de alarmes ativos do objeto. Esta propriedade é somente de leitura.

13.3.2 ActiveHighAlarms

9 Indica o número de alarmes ativos com severidade alta. Esta propriedade é somente de leitura.

13.3.3 ActiveHighNACKAlarms

9 Indica o número de alarmes não-reconhecidos com severidade alta. Esta propriedade é somente de leitura.

13.3.4 ActiveLowAlarms

9 Indica o número de alarmes ativos com severidade baixa. Esta propriedade é somente de leitura.

13.3.5 ActiveLowNACKAlarms

9 Indica o número de alarmes não-reconhecidos com severidade baixa. Esta propriedade é somente de leitura.

13.3.6 ActiveMedAlarms

9 Indica o número de alarmes ativos com severidade média. Esta propriedade é somente de leitura.

13.3.7 ActiveMedNACKAlarms

9 Indica o número de alarmes não-reconhecidos com severidade média. Esta propriedade é somente de leitura.

13.3.8 ActiveNACKAlarms

9 Indica o número total de alarmes que estão sem reconhecimento no objeto (ativos ou não). Esta propriedade é somente de leitura.

13.3.9 AllowAckAll

Habilita a opção presente no menu *pop-up* do E3Alarm, que permite o reconhecimento de todos os alarmes. O valor padrão desta propriedade é True.

13.3.10 AllowAckCurrentFilter

Habilita a opção presente no menu *pop-up* do E3Alarm, que permite o reconhecimento de todos os alarmes do filtro atual. Se não houverem alarmes visíveis, a propriedade não tem efeito. O valor padrão desta propriedade é True.

13.3.11 AllowAckSelected

☑ Habilita a opção presente no menu *pop-up* do E3Alarm, que permite o reconhecimento de alarmes selecionados. Se não houverem alarmes selecionados, a propriedade não tem efeito. O valor padrão desta propriedade é True.

13.3.12 AllowColumnClick

☑ Habilita ou desabilita a seleção dos campos e a direção de ordenação destes através do clique no cabeçalho das colunas do E3Alarm em tempo de execução. Se esta opção estiver configurada para True e o cabeçalho for visível (ver propriedade **ColumnHeader**), ao clicar no título da coluna, os dados serão ordenados tendo esta coluna como chave. Clicando novamente nesta mesma coluna, a ordenação será feita na direção contrária (muda-se a ordem ascendente para descendente e vice-versa). Ao clicar na coluna com a tecla SHIFT pressionada, o campo é usado como segunda chave. Como na chave primária, um segundo clique com SHIFT inverte a ordenação do campo secundário.

13.3.13 AlarmServer

A Nome do servidor de alarmes único existente na aplicação.

13.3.14 ColumnHeader

☑ Quando configurada como True, esta propriedade habilita a visualização do cabeçalho do E3Alarm. O cabeçalho também permite que seja feita a reordenação dos dados da tabela visualmente (ver propriedade **AllowColumnClick**). O valor padrão desta propriedade é True.

13.3.15 Domain

A Especifica o domínio ao qual o E3Alarm irá se conectar. O padrão desta propriedade é vazio, ou seja, o E3Alarm se conecta ao mesmo domínio do Viewer onde ele se encontra. Por exemplo, \\NomeDeOutroServidor.

13.3.16 Enabled

☑ Habilita o objeto ActiveX no projeto. O valor padrão desta propriedade é True.

13.3.17 Font

A Determina a fonte (tipo de letra) do cabeçalho e das linhas do E3Alarm. Esta propriedade é somente de leitura e só pode ser modificada via Studio, não em tempo de execução.

13.3.18 Filter

A Controla as áreas de alarmes visíveis no E3Alarm. Se seu valor não for vazio, serão apresentados eventos cujo nome das áreas comecem com o texto indicado. Por exemplo, se Filter é "Ana", serão mostrados os alarmes de áreas como "Analogicos.Producao" ou "Analise", mas não "Digitais.Analise" ou "Digitais.Producao". Quando a propriedade **SimpleAreaFilter** for False, a área de alarme também permitirá o uso de coringas para a filtragem (* ou ?) e permitirá múltiplos filtros de área, separados por dois pontos.

Os coringas permitidos são:

- "*" : aceita nenhum ou qualquer quantidade de caracteres.
- "?" : aceita um caractere qualquer.
- "#" : aceita um dígito qualquer.
- "[]" : permite especificar um conjunto de caracteres.
- "[ab]" : aceita um caractere se for "a" ou "b".
- "[f-h]" : aceita um caractere entre "f" e "h".
- "[!cz]" : aceita um caractere que não seja nem "c" nem "z".
- "[!m-p]" : aceita um caractere que não seja de "m" até "p".

O padrão dessa propriedade é vazio, ou seja, sem filtragem por área (ver também propriedades **SimpleAreaFilter**, **ShowHighPriority**, **ShowMediumPriority** e **ShowLowPriority**). Exemplo:

```
Sub ComboBox1_Change()  
    ' O valor da ComboBox indica o filtro de area do E3Alarm  
    Screen.Item("E3Alarm1").Filter = Value  
End Sub
```

13.3.19 FilterType

 Efetua os filtros de alarme. As opções disponíveis são:

- **1 - OnlyAlarms:** mostra apenas alarmes.
- **2 - Only Events:** mostra apenas eventos.
- **3 - AlarmsAndEvents:** mostra tanto alarmes quanto eventos.

13.3.20 Frame

 Retorna o quadro-pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

13.3.21 GridBkColor

 Esta propriedade determina a cor de fundo do E3Alarm. O valor padrão desta propriedade é a cor configurada no Windows para o item **Janela (Painel de Controle - Vídeo - Aparência - Avançada)**.

13.3.22 HasFocus

 Esta propriedade determina que o objeto selecionado está com o foco. Esta propriedade é acessível apenas em tempo de execução.

13.3.23 Height

 A propriedade **Height** define a altura do E3Alarm, em unidades Himetric.

13.3.24 Layer

 Esta propriedade define em quais camadas o objeto deve aparecer. O valor representa uma máscara de 32 bits, um bit para cada camada. Portanto, podem ser definidas até 32 camadas individuais. Assim, objetos podem ser agrupados logicamente e mostrados ou escondidos apenas com a modificação da máscara da propriedade **Layer**.

13.3.25 MouseOver

☑ A propriedade **MouseOver** informa se o ponteiro do mouse encontra-se em cima da Tela. Caso afirmativo, a propriedade **MouseOver** é habilitada; do contrário, será False. Esta propriedade é somente de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

13.3.26 MouseOverChild

☑ A propriedade **MouseOverChild** informa se o ponteiro do mouse encontra-se em cima de um dos objetos inseridos na Tela. Caso afirmativo, a propriedade **MouseOverChild** é habilitada. Caso contrário, será False. Esta propriedade é somente de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

13.3.27 PopupMenu

☑ Habilita um menu *pop-up* acessado ao clicar com o botão direito do mouse sobre o E3Alarm. O valor padrão desta propriedade é True.

13.3.28 PrimarySortAscending

☑ Quando esta propriedade estiver configurada para False, a ordenação de eventos pelo campo primário será no sentido descendente. Caso contrário, o sentido será ascendente. O valor padrão desta propriedade é False.

13.3.29 PrimarySortField

A Determina o campo primário para a ordenação dos eventos no E3Alarm. O nome do campo deve ser sempre especificado em inglês (ver tabela de campos dos alarmes no apêndice deste manual). O valor padrão dessa propriedade é "EventTime". Quando esta opção está vazia, a propriedade **SecondarySortField** não tem efeito.

13.3.30 SecondarySortAscending

☑ Quando esta propriedade estiver configurada para True, a ordenação de eventos pelo campo secundário será no sentido ascendente. Caso contrário, o sentido será descendente. O valor padrão desta propriedade é False.

13.3.31 SecondarySortField

A Determina o campo secundário para a ordenação dos eventos no E3Alarm. O nome do campo deve ser sempre especificado em inglês (ver tabela de campos dos alarmes no apêndice deste manual). O valor padrão dessa propriedade é vazio. Essa propriedade não tem efeito quando a propriedade **PrimarySortField** estiver vazia.

13.3.32 ShowHighPriority

Filtra quais alarmes serão mostrados ou não, de acordo com a sua severidade. Quando estiver em True, serão mostrados os alarmes de severidade alta; do contrário, estes alarmes não serão mostrados. O valor padrão desta propriedade é True.

13.3.33 ShowLowPriority

Filtra quais alarmes serão mostrados ou não, de acordo com a sua severidade. Quando estiver em True, serão mostrados os alarmes de severidade baixa; do contrário, estes alarmes não serão mostrados. O valor padrão desta propriedade é True.

13.3.34 ShowMediumPriority

Filtra quais alarmes serão mostrados ou não, de acordo com a sua severidade. Quando estiver em True, serão mostrados os alarmes de severidade média; do contrário, estes alarmes não serão mostrados. O valor padrão desta propriedade é True.

13.3.35 SimpleAreaFilter

Quando esta propriedade for True, o comportamento de filtragem pelo nome de área de alarmes é baseado apenas na coincidência da parte inicial do nome. Quando for False, o comportamento leva em conta todo o nome da área, mas permite o uso de caracteres-coringa e múltiplos filtros de área que devem ser separados por dois pontos. Ver também a propriedade **Filter**, que especifica o filtro por nome de área.

13.3.36 TabStop

☑ Esta propriedade determina a utilização da tecla TAB no sistema. Se o campo estiver True, é possível utilizar a tecla; caso contrário, a tecla não pode ser utilizada.

13.3.37 Tip

A A propriedade **Tip** mostra um texto *popup* quando o mouse estiver por um breve momento sobre o objeto em execução.

13.3.38 Visible

☑ Esta propriedade define se o objeto deve ou não estar visível. Se a propriedade estiver configurada para True, o objeto fica visível, desde que os seguintes itens sejam também obedecidos, obrigatoriamente: o objeto-pai deste objeto também deve estar visível, e a propriedade **Layer** do objeto deve estar presente na camada de Tela. Do contrário, o objeto fica invisível.

13.3.39 Width

9 A propriedade **Width** define a largura, em unidades Himetric, do objeto.

13.3.40 X

9 A propriedade **X** define a coordenada horizontal esquerda, em unidades Himetric, do objeto.

13.3.41 Y

9 Esta propriedade define a coordenada vertical superior, em unidades Himetric, do objeto.

CAPÍTULO

14 Históricos

Esta seção contém informações sobre métodos e propriedades do objeto Histórico. Este objeto não possui eventos associados.

14.1 Métodos

Esta seção contém informações sobre os métodos do objeto Histórico.

14.1.1 StartAcquisition

StartAcquisition()

Habilita o Histórico a gravar os valores do seus campos periodicamente, a partir da taxa especificada na propriedade **ScanTime**. Este método pode ser chamado a qualquer momento após a chamada do método **StopAcquisition**. O comportamento padrão deste método é iniciar a aplicação habilitado, ou seja, este método é sempre executado internamente ao se iniciar o Histórico. Exemplo:

```
Sub Botao1_Click()  
    ' Quando clica no botão, habilita o histórico.  
    Application.GetObject("Hist1").StartAcquisition()  
End Sub
```

14.1.2 StopAcquisition

StopAcquisition()

Desabilita a gravação dos registros por período no Histórico, independente do valor especificado na propriedade **ScanTime**. A gravação por período fica desabilitada até o método **StartAcquisition** ser chamado. O comportamento padrão do objeto Histórico é iniciar a aplicação com a gravação habilitada. Exemplo:

```
Sub Hist1_OnStartRunning()  
    ' Desabilita o Histórico assim que iniciar.  
    StopAcquisition()  
End Sub
```

14.1.3 WriteRecord

WriteRecord()

Insera uma nova linha de informação no banco de dados. Os valores são obtidos a partir dos valores atuais de cada uma das variáveis especificadas como fontes de dados dos campos do histórico. Utiliza-se este método em duas situações:

- Para gravar uma nova linha de dados antes do tempo previsto para a próxima gravação quando o Histórico está habilitado por tempo.
- Para gravar um novo conjunto de dados quando o Histórico está desabilitado.

Exemplo:

```
Sub Tag1_OnValueChanged()  
    ' Grava uma nova linha em um histórico  
    ' quando um tag muda de valor.  
    Application.GetObject("Hist1").WriteRecord()  
End Sub
```

14.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Histórico.

14.2.1 BackupDiscardInterval

9 Indica a quantidade de unidades de tempo durante o qual os dados de backup serão mantidos na tabela principal e na tabela de backup, até serem descartados (por exemplo, para manter os dados por 24 meses na tabela principal, e mais seis meses na tabela de backup, o valor da propriedade deve ser de 30 meses). Esta propriedade trabalha em conjunto com a propriedade **BackupDiscardTimeUnit**. O valor padrão desta propriedade é 12 (doze unidades de tempo indicadas em **BackupDiscardTimeUnit**).

NOTA: O tempo total indicado pela combinação das propriedades **BackupDiscardInterval** e **BackupDiscardTimeUnit** deve ser maior que o tempo indicado pelas propriedades **DiscardInterval** e **DiscardTimeUnit**.

14.2.2 BackupDiscardTimeUnit

 A propriedade **BackupDiscardTimeUnit** indica a unidade de tempo em que os dados de backup ficarão armazenados até serem descartados. As opções disponíveis são:

- **0 - dtHour**: horas.
- **1 - dtDay**: dias.
- **2 - dtMonth**: meses (padrão).
- **3 - dtMinute**: minutos.

Esta propriedade trabalha em conjunto com a propriedade **BackupDiscardInterval**.

14.2.3 CacheSize

 Define o tamanho do bloco de registros que deve ser lido pelo Histórico antes de ser enviado para o banco de dados. Por exemplo, se **CacheSize** é igual a 4, serão enviados blocos contendo quatro registros cada um para o DBServer associado. Os valores válidos para esta propriedade devem estar no intervalo entre 1 e 4. O valor padrão desta propriedade é 1.

NOTA: O bloco de registros é enviado a cada 1 segundo, mesmo que não tenha atingido o tamanho configurado na propriedade **CacheSize**.

14.2.4 CompressedTable

Habilita o uso de banda morta para a gravação de dados. O valor padrão desta propriedade é False.

14.2.5 DBServer

 Indica o objeto Banco de Dados utilizado no histórico para criação de tabelas e registros de dados. O valor padrão desta propriedade é vazio.

14.2.6 DeadBand

 Esta propriedade trabalha em conjunto com a propriedade **CompressedTable**. Indica o valor calculado sobre o último valor gravado (em percentagem) que define se este novo valor será gravado. Se o valor gravado não for numérico, sua modificação faz com que todos os valores sejam gravados.

14.2.7 DiscardInterval

 Esta propriedade trabalha em conjunto com a propriedade **DiscardTimeUnit**. A propriedade **DiscardInterval** indica o intervalo de tempo durante o qual os dados do Histórico ficarão armazenados na tabela do banco de dados, até serem descartados. O valor padrão desta propriedade é 1 (uma unidade de tempo indicada em **DiscardTimeUnit**). Se esta propriedade for configurada com um valor menor ou igual ao valor da propriedade **BackupDiscardInterval**, o E3 automaticamente ajusta o valor de **BackupDiscardInterval** como o dobro do valor de **DiscardInterval**. Esta propriedade pode ser modificada em tempo de execução.

14.2.8 DiscardTimeUnit

 Esta propriedade trabalha em conjunto com a propriedade **DiscardInterval**. A propriedade **DiscardTimeUnit** indica a unidade de tempo em que os dados da tabela ficarão armazenados até serem descartados. As opções disponíveis são:

- **0 - dtHour**: horas.
- **1 - dtDay**: dias.
- **2 - dtMonth**: meses (padrão).
- **3 - dtMinute**: minutos.

14.2.9 EnableBackupTable

Cria uma tabela de backup onde os dados descartados permanecerão por segurança. Se configurada para True, a tabela será criada; do contrário, não haverá tabela de backup. O valor padrão desta propriedade é False.

14.2.10 EnableDiscard

Indica descarte dos dados do Histórico após um certo tempo. Se configurada para False, os dados serão armazenados indefinidamente na tabela; do contrário, serão descartados após determinado tempo. O valor padrão dessa propriedade é False.

14.2.11 EnableQualityLogs

Quando em True, ao iniciar o Histórico o E3 gera um registro igual ao primeiro registro coletado mas com qualidade ruim (0) e *timestamp* de um segundo antes.

14.2.12 ScanTime

 Define a variação de intervalo do tempo, em milissegundos, que o Histórico irá esperar para fazer a aquisição e gravação de um novo registro na tabela. Utilize esta propriedade se houver necessidade de maior ou menor quantidade de dados gerados por segundo. O valor padrão desta propriedade é 1000.

14.2.13 TableName

A Define o nome da tabela que será utilizada no Histórico.

14.2.14 UserTable

Quando esta propriedade está configurada para True, identifica que o Histórico é do usuário, ou seja, os dados da tabela foram importados do banco. Caso contrário, é um histórico normal do E3. Esta propriedade é apenas de leitura.

14.2.15 UseTagQuality

Se for True, o Histórico vai usar o valor de qualidade da fonte do tag; caso contrário, será utilizado o método antigo de avaliação (0 = valor duvidoso; 1 = valor bom).

14.2.16 VerificationInternal

9 Esta propriedade trabalha em conjunto com a propriedade **VerificationUnit** para controlar o intervalo de tempo que o E3 verifica a antiguidade dos dados, para depois descartá-los. O valor padrão desta propriedade é 1 (uma unidade de tempo indicada em **VerificationUnit**).

14.2.17 VerificationUnit

Esta propriedade trabalha em conjunto com a propriedade **VerificationInternal**. A propriedade **VerificationUnit** indica a unidade de tempo em que será feita a verificação para descarte dos dados. As opções disponíveis são:

- **0 - dtHour**: horas.
- **1 - dtDay**: dias.
- **2 - dtMonth**: meses (padrão).
- **3 - dtMinute**: minutos.

CAPÍTULO

15

Storage

Esta seção contém informações sobre métodos e propriedades do objeto Storage. Este objeto não possui eventos associados.

15.1 Métodos

Esta seção contém informações sobre os métodos do objeto Storage.

15.1.1 CreateNewSession

CreateNewSession(*[DefaultType]*, *[DefaultMinRecTime]*, *[DefaultMaxRecTime]*, *[DefaultDeadBand]*, *[DefaultUnit]*)

Gera uma sessão que tem a capacidade de incluir dados de um E3Storage de forma independente da aquisição normal. Os parâmetros opcionais são utilizados nas configurações dos tags da sessão, caso não informados em suas criações. São eles, respectivamente: tipo do dado, intervalo mínimo de tempo entre gravações, intervalo máximo de tempo sem gravações, banda morta e unidade da banda morta do tag.

Como usar:

Criação de uma sessão

```
Set Session = Application.GetObject("Storage1").CreateNewSession(0, 0, 3600, 10, 1)
```

Adição de um tag à sessão (método AddField)

```
result = session.AddField("TempTag", 0, 0, 100000, 15, 1)
```

Adição de valores (método AddValue)

```
result = session.AddValue("TempTag", Now, 192, 10)
```

15.1.2 StartAcquisition

StartAcquisition()

Inicia ou retoma a geração de dados que vão para o Banco de Dados. O Storage recebe notificações sobre quais tags cadastrados sofreram modificação, e quando isto acontece ele verifica se os registros serão gravados ou não. Quando este método é chamado, a notificação de mudança e a geração de gravações serão iniciadas ou retomadas. Exemplo:

```
Sub Botao1_Click()  
    Quando clica no botão, habilita o Storage.  
    Application.GetObject("Storage1").StartAcquisition()  
End Sub
```

15.1.3 StopAcquisition

StopAcquisition()

Pára a geração de dados que vão para o Banco de Dados. O Storage recebe notificações sobre quais tags cadastrados sofreram modificação, e quando isto acontece ele verifica se os registros serão gravados ou não. Quando este método é chamado, a notificação de mudança e a geração de gravações são paralisadas.

Exemplo:

```
Sub Storage1_OnStartRunning()  
    Desabilita o Storage assim que iniciar.  
    StopAcquisition()  
End Sub
```

15.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Storage.

15.2.1 BackupDiscardInterval

9 Indica a quantidade de unidades de tempo durante o qual os dados de backup serão mantidos na tabela principal e na tabela de backup, até serem descartados (por exemplo, para manter os dados por 24 meses na tabela principal, e mais seis meses na tabela de backup, o valor da propriedade deve ser de 30 meses). Esta propriedade trabalha em conjunto com a propriedade **BackupDiscardTimeUnit**. O valor padrão desta propriedade é 12 (doze unidades de tempo indicadas em **BackupDiscardTimeUnit**).

NOTA: O tempo total indicado pela combinação das propriedades **BackupDiscardInterval** e **BackupDiscardTimeUnit** deve ser maior que o tempo indicado pelas propriedades **DiscardInterval** e **DiscardTimeUnit**.

15.2.2 BackupDiscardTimeUnit

 A propriedade **BackupDiscardTimeUnit** indica a unidade de tempo em que os dados de backup ficarão armazenados até serem descartados. As opções disponíveis são:

- **0 - dtHour**: horas.
- **1 - dtDay**: dias.
- **2 - dtMonth**: meses (padrão).
- **3 - dtMinute**: minutos.

Esta propriedade trabalha em conjunto com a propriedade **BackupDiscardInterval**.

15.2.3 CacheSize

 Define o tamanho do bloco de registros que deve ser lido pelo Storage antes de ser enviado para o banco de dados. Por exemplo, se **CacheSize** é igual a 4, serão enviados blocos contendo quatro registros cada um para o DBServer associado. O valor padrão desta propriedade é 10.

NOTA: O bloco de registros é enviado a cada 1 segundo, mesmo que não tenha atingido o tamanho configurado na propriedade **CacheSize**.

15.2.4 CompressionRate

 Mostra a taxa de compressão de dados obtida até o momento.

15.2.5 DBServer

 Indica o objeto Banco de Dados utilizado no Storage para criação de tabelas e registros de dados. O valor padrão desta propriedade é vazio.

15.2.6 DiscardInterval

9 Esta propriedade trabalha em conjunto com a propriedade **DiscardTimeUnit**. A propriedade **DiscardInterval** indica o intervalo de tempo durante o qual os dados do Histórico ficarão armazenados na tabela do banco de dados, até serem descartados. O valor padrão desta propriedade é 1 (uma unidade de tempo indicada em **DiscardTimeUnit**). Se esta propriedade for configurada com um valor menor ou igual ao valor da propriedade **BackupDiscardInterval**, o E3 automaticamente ajusta o valor de **BackupDiscardInterval** como o dobro do valor de **DiscardInterval**. Esta propriedade pode ser modificada em tempo de execução.

15.2.7 DiscardTimeUnit

 Esta propriedade trabalha em conjunto com a propriedade **DiscardInterval**. A propriedade **DiscardTimeUnit** indica a unidade de tempo em que os dados ficarão armazenados até serem descartados. As opções disponíveis são:

- **0 - dtHour**: horas.
- **1 - dtDay**: dias.
- **2 - dtMonth**: meses (padrão).
- **3 - dtMinute**: minutos.

15.2.8 EnableBackupTable

Cria uma tabela de backup onde os dados descartados permanecerão por segurança. Se configurada para True, a tabela será criada; do contrário, não haverá tabela de backup. O valor padrão desta propriedade é False.

15.2.9 EnableDiscard

Habilita o descarte dos dados após um certo tempo. Se configurada para False, os dados serão armazenados indefinidamente na tabela; do contrário, serão descartados após determinado tempo. O valor padrão dessa propriedade é False.

15.2.10 Fields

 Coleção que aponta para os campos criados no Storage. Para cada campo é possível visualizar as propriedades **Name** e **Link**, e modificar as propriedades **Type**, **MinRecTime**, **MaxRecTime**, **DeadBand** e **DeadBandUnit**.

15.2.11 StringFieldSize

 Esta propriedade especifica o tamanho máximo que os campos do tipo **String** do objeto Storage poderão ter (será o tamanho utilizado na criação do campo **Value** da tabela de strings).

15.2.12 TableName

 Define o nome da tabela que será utilizada no Storage.

15.2.13 VerificationInternal

 Esta propriedade trabalha em conjunto com a propriedade **VerificationUnit** para controlar o intervalo de tempo que o E3 verifica a antiguidade dos dados, para depois descartá-los. O valor padrão desta propriedade é 1 (uma unidade de tempo indicada em **VerificationUnit**).

15.2.14 VerificationUnit

 Esta propriedade trabalha em conjunto com a propriedade **VerificationInternal**. A propriedade **VerificationUnit** indica a unidade de tempo em que será feita a verificação para descarte dos dados. As opções disponíveis são:

- **0 - dtHour**: horas.
- **1 - dtDay**: dias.
- **2 - dtMonth**: meses (padrão).
- **3 - dtMinute**: minutos.

15.3 Campo do Storage

Esta seção contém informações sobre propriedades do objeto Campo do Storage. Este objeto não possui eventos nem métodos associados.

15.3.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Campo do Storage.

15.3.1.1 DeadBand

9 Banda morta para cálculo do algoritmo.

15.3.1.2 DeadBandUnit

9 Valor absoluto ou a porcentagem de modificação.

15.3.1.3 Link

A Determina a fonte de dados que será associada ao campo. Propriedade apenas de leitura.

15.3.1.4 MaxRecTime

9 Tempo máximo de espera por um dado. Se este tempo for ultrapassado, o dado torna-se antigo e é armazenado.

15.3.1.5 MinRecTime

9 Tempo mínimo para inserção de um dado na tabela. Este tempo é calculado em milissegundos.

15.3.1.6 Name

A Propriedade que retorna o nome configurado para este campo no objeto Storage. Através dele, é possível buscar itens na coleção.

15.3.1.7 Type

 Retorna o tipo do objeto no formato do Storage (**0 - Double, 1 - Bit, 2 - String**). Esta propriedade é de leitura e escrita, mas só aceita mudanças enquanto a coleta de dados para este campo ainda não começou.

CAPÍTULO

16

Consultas

Esta seção contém informações sobre eventos, métodos e propriedades do objeto Consulta.

16.1 Eventos

Esta seção contém informações sobre os eventos do objeto Consulta.

16.1.1 OnAsyncQueryFinish

OnAsyncQueryFinish(Recordset, Error)

Ocorre quando o método **GetAsyncADORecordset** é retornado. O parâmetro *Recordset* é o ADO Recordset gerado pela consulta, e o parâmetro *Error* é um booleano que, quando verdadeiro, mostra que o objeto não conseguiu ser gerado. Exemplo:

```
Sub Consulta1_OnAsyncQueryFinish(Recordset, Error)
    MsgBox "Retornou " + CStr(Recordset.RecordCount) + " registros"
End Sub
```

16.2 Métodos

Esta seção contém informações sobre os métodos do objeto Consulta.

16.2.1 AddField

AddField(Name, [Table])

O método **AddField** adiciona um novo campo da tabela na consulta. O parâmetro *Name* determina o nome do novo campo que será adicionado à consulta. O parâmetro *Table* determina o nome da tabela a qual o campo pertence. Este método foi desenvolvido apenas para manter a compatibilidade com o antigo objeto Consulta do E3Chart. Exemplo:

```
Sub Botao1_Click()
    Screen.Item("E3Browser").Item("Consulta").AddField "Campo1"
End Sub
```

16.2.2 AddStorageTag

AddStorageTag(*Name*, *FieldType*)

Adiciona um tag pertencente ao Storage a ser adicionado à consulta. O parâmetro *Name* recebe o nome do tag a ser adicionado. O parâmetro *FieldType* indica o tipo deste tag (**0 - Double**, **1 - Bit**, **2 - String**). Retorna um booleano indicando o sucesso (ou não) da operação.

16.2.3 AddTable

AddTable(*Name*)

Adiciona uma tabela do banco de dados na consulta. O parâmetro *Table* determina o nome da tabela a ser adicionada.

16.2.4 Execute

Execute(*ImmediateExecute*)

O método **Execute** executa um comando SQL que não tenha retorno (como **DELETE**, **UPDATE** ou **INSERT**), configurado no SQLQuery. O parâmetro *ImmediateExecute* indica se a operação passará pelas filas de operações do DB (arquivos .e3i e .e3o) antes de chegar ao banco (caso esteja configurada como False), ou se será enviada diretamente para o Banco de Dados (caso esteja configurada como True). A vantagem de se utilizar a consulta para executar comandos é a utilização de variáveis, como em uma consulta simples.

Exemplo de comandos SQL:

```
DELETE FROM teste WHERE cod > 10
UPDATE teste SET cod = 10 WHERE cod > 10
INSERT INTO teste(cod) VALUES(10)
```

Exemplo:

```
Sub CommandButton1_Click()
    Screen.Item("consulta1").Execute
End Sub
```

16.2.5 GetADORecordSet

GetADORecordSet()

O método **GetADORecordSet** retorna um Recordset do tipo ADO (*ActiveX Data Object*), resultante da execução da consulta configurada. Exemplo:

```

Sub Botao1_Click()
    Set rec = Screen.Item("Consulta1").GetADORecordSet()
    strDadas = ""
    i = 0

    ' Mostra uma mensagem com os 10 primeiros
    ' registros da coluna E3TimeStamp
    while (NOT rec.EOF AND i < 10)
        strDadas = strDadas & CStr(rec.Fields.Item("E3TimeStamp").Value) & _
            Chr(10) & Chr(13)
        i = i + 1
        rec.MoveNext()
    wend
    MsgBox strDadas
End Sub

```

16.2.6 GetAsyncADORecordSet

GetAsyncADORecordSet()

Cria uma consulta e, quando ela estiver finalizada, gera o evento **OnAsyncQueryFinish** do próprio objeto, passando para este evento o resultado (Recordset) gerado pela consulta.

16.2.7 GetE3QueryFields

GetE3QueryFields()

O método **GetE3QueryFields** retorna a coleção de campos (colunas) de uma consulta. Cada um destes campos possui as seguintes propriedades que podem ser modificadas, a saber:

Propriedades dos campos da consulta

Nome	Tipo	Descrição
ColumnName	Texto	Nome da coluna. Este nome deve existir nas tabelas adicionadas nesta consulta.
TableName	Texto	Nome da tabela a que pertence a coluna. Esta tabela deve ter sido adicionada na configuração da consulta.
Alias	Texto	Apelido da coluna na consulta.
Criteria	Texto	Filtro sobre a coluna.
OrderBy	Texto	Ordenação dos dados da coluna. Os valores válidos são "ASC" (ordenação ascendente), "DESC" (ordenação descendente) ou "" (sem

Nome	Tipo	Descrição
		ordenação). Qualquer valor diferente significa que o campo não possui ordenação.
OrderNumber	Número	Número de ordem da coluna em relação às outras colunas que compõem a ordenação da consulta. Este valor só será aceito como maior que 0 se a coluna tiver um tipo de ordenação. Este valor terá que ser menor ou igual ao número de campos que compõem a ordenação da consulta.
Function	Texto	Função para a qual a coluna pode ser passada como parâmetro.
GroupBy	Booleano	Em True, indica que a coluna faz parte de um agrupamento.
Visible	Booleano	Em True, indica que a coluna está visível.

Exemplo:

```

Sub Botao1_Click()
    ' Percorre a coleção Campos, mostrando-os em uma caixa
    ' de mensagem e ajustando-os como visíveis na configuração da
    ' consulta.
    Set Browser = Screen.Item("E3Browser")
    Set Consulta = Browser.Item("Consulta")
    Set Campos = Consulta.GetE3QueryFields()
    For Each campo In Campos
        MsgBox CStr(campo.TableName) & "-" & CStr(campo.ColumnName)
        campo.Visible = TRUE
    Next
    ' Refaz a consulta do E3Browser, que acabou de ser
    ' modificada, para que todos os campos apareçam.
    Browser.RetrieveE3QueryFields()
    Browser.Requery()
End Sub

```

NOTA: Para usar este método, a consulta deverá ser previamente criada em tempo de configuração.

16.2.8 RemoveField

RemoveField(*FieldName*, [*Table*])

O método **RemoveField** remove um campo previamente incluído em uma consulta. O parâmetro *FieldName* determina o nome do campo que será removido. O parâmetro *Table* determina o nome da tabela a qual o campo pertence. Este método, a exemplo do método **AddField**, foi incluído para manter a compatibilidade com as versões antigas da consulta do E3Chart. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Browser").Item("Consulta")._  
        RemoveField "Campo1"  
End Sub
```

16.2.9 RemoveStorageTag

RemoveStorageTag(*Name*)

Remove um tag previamente configurado na consulta. O parâmetro *Name* indica o nome do tag. Retorna um booleano indicando o sucesso (ou não) da operação.

16.2.10 RemoveTable

RemoveTable(*TableName*)

Remove uma tabela da consulta. O parâmetro *TableName* determina o nome da tabela a ser removida.

16.2.11 SetVariableValue

SetVariableValue(*VarName*, *Value*)

O método **SetVariableValue** ajusta o valor de uma variável configurada na consulta, de forma que este valor possa ser informado como um filtro ou parâmetro antes de a mesma ser realizada. Deve ser definido o nome da variável (*VarName*) e seu valor (*Value*), que pode ser um número, texto ou data/hora. Exemplo:

```
Sub CommandButton_Click()  
    Set cons = Screen.Item("E3Browser1").Item("Consulta1")  
    DataInicial = now - 1  
    DataFinal = now  
    cons.SetVariableValue "DataIni", DataInicial  
    cons.SetVariableValue "DataFim", DataFinal  
End Sub
```

16.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto Consulta.

NOTA: Não é recomendável o acesso direto destas propriedades via scripts. O ideal é acessar o objeto Consulta, passando parâmetros através do método **SetVariableValue** e modificando filtros ou campos através da coleção retornada com o método **GetE3QueryFields**.

16.3.1 CursorLocation

 Define o local onde a consulta será gerada e manipulada, sob o ponto de vista do SGBD (Sistema Gerenciador do Banco de Dados). As opções disponíveis são:

- **0 - clServer:** a consulta será gerada no SGBD (servidor).
- **1 - clClient:** a consulta será gerada no E3 Server (cliente).

O valor padrão desta propriedade é **0 - clServer**. Ver também a propriedade **CursorType**.

NOTA: A utilização da opção **1 - clClient** requer um certo cuidado, pois neste modo a consulta será gerada em memória no E3 Server e pode comprometer a performance do sistema, dependendo do tamanho da consulta.

16.3.2 CursorType

 Define o tipo de consulta de acordo com a visualização dos dados. O valor padrão desta propriedade é **0 - ctKeyset**. Todas as opções disponíveis estão na tabela a seguir.

Opções disponíveis para **CursorType**

OPÇÃO	DESCRIÇÃO
0 - ctKeyset	Qualquer mudança nos registros inicialmente retornados pela consulta será visível (Padrão).
1 - ctStatic	Nenhuma mudança nos registros inicialmente retornados pela consulta será visível.

OPÇÃO	DESCRIÇÃO
2 - ctDynamic	Todos os novos registros adicionados na consulta serão visíveis, além das mudanças nos registros inicialmente retornados pela consulta.

16.3.3 DataSource

A Indica o objeto Banco de Dados ou o objeto Storage que será usado na consulta. Esta propriedade é somente de leitura, mas pode ser modificada em tempo de execução.

16.3.4 Fields

A Texto com os campos a serem visualizados na consulta, separados por vírgula. Corresponde ao argumento da cláusula **SELECT** do código SQL da consulta. Em branco (vazio), determina que a consulta deve mostrar todos os campos da tabela. Esta propriedade é somente de leitura, mas pode ser modificada em tempo de execução.

16.3.5 Frame

 Retorna o quadro pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

16.3.6 FunctionSubType

 Especifica o subtipo da função indicada por **FunctionType**. Apenas as opções **1 - ftArchivedValue**, **2 - ftTagAttribute** e **6 - ftCalculatedData** possuem subtipos. Para as outras funções, **FunctionSubType** assume o valor **-1 - fsNoSubType**. A tabela a seguir mostra os possíveis valores da propriedade, de acordo com a função escolhida na propriedade **FunctionType**:

Subtipos para função ArchivedValue (FunctionType = 1)

SUBTIPO	DESCRIÇÃO
0 - fsPreviousArchivedValue	Valor armazenado imediatamente anterior ao <i>timestamp</i> fornecido.
1 - fsInterpolatedArchivedValue	Valor calculado a partir do valor anterior e posterior.

SUBTIPO	DESCRIÇÃO
2 - fsNextArchivedValue	Valor armazenado imediatamente posterior ao <i>timestamp</i> .
3 - fsExactArchivedValue	Se for encontrado um valor armazenado no exato instante que foi fornecido pelo <i>timestamp</i> .

Subtipos para função TagAttribute (FunctionType = 2)

SUBTIPO	DESCRIÇÃO
0 - fsTagAttributeDescription	Significado ou descrição do tag.
1 - fsTagAttributeSource	Caminho do Tag que está sendo armazenado.
2 - fsTagAttributeType	Tipo de dado: Double, Boolean, String .
3 - fsTagAttributeEU	Unidades de engenharia.
4 - fsTagAttributeLowEng	Limite inferior.
5 - fsTagAttributeHighEng	Limite superior.
6 - fsTagAttributeDeadBand	Banda Morta para gravação.
7 - fsTagAttributeDeadBandUnit	Unidade da Banda Morta (valor absoluto ou porcentagem).
8 - fsTagAttributeMinRecTime	Tempo mínimo para gravação (variações menores que esse intervalo são desprezadas).
9 - fsTagAttributeMaxRecTime	Tempo máximo para gravação (a ausência de variação nesses intervalos força uma gravação).

Subtipos para função CalculatedData (FunctionType = 6)

SUBTIPO	DESCRIÇÃO
0 - fsTotalCalculatedData	Total dos valores.
1 - fsMinimumCalculatedData	Valor mínimo.
2 - fsMaximumCalculatedData	Valor máximo.
3 - fsStandardCalculatedData	Desvio padrão.
4 - fsRangeCalculatedData	Amplitude dos valores.
5 - fsMeanCalculatedData	Média dos valores.
6 - fsMedianCalculatedData	Mediana dos valores.

16.3.7 FunctionType

 Esta propriedade é válida quando um objeto Storage é fonte da Consulta (isto é indicado pela propriedade **DataSource**). Ela especifica a função que irá definir os dados gerados pela Consulta. Algumas funções possuem subfunções, que podem ser indicadas na propriedade **FunctionSubType**. A propriedade **FunctionType** pode assumir os seguintes valores:

Opções disponíveis para FunctionType

OPÇÃO	DESCRIÇÃO
-1 - ftNoFunction	Não há nenhuma função definida.
0 - ftLastValue	Retorna o último valor armazenado na Base de Dados.
1 - ftArchivedValue	Retorna um valor armazenado em relação a um determinado instante de tempo definido na variável TimeStamp . O tipo de relação é definido em FunctionSubType .
2 - ftTagAttribute	Devolve um atributo do Tag, definido em FunctionSubType .

OPÇÃO	DESCRIÇÃO
3 - ftCompressedDataNValues	Retorna, para um único Tag, <i>N</i> valores definidos na variável NumVals, armazenados a partir de um instante inicial, definido na variável StartTime.
4 - ftCompressedDataStartEndTime	Retorna, para um único Tag, os valores armazenados que estão entre o intervalo definido pelas variáveis StartTime e EndTime.
5 - ftSampledData	Retorna, para um ou mais Tags, os valores interpolados (ou seja, estimados) entre os instantes definidos pelas variáveis StartTime e EndTime, a intervalos fixos definidos pela variável TimeInterval.
6 - ftCalculatedData	Retorna, para um ou mais Tags, o resultado de operações matemáticas aplicadas aos dados entre os instantes definidos pelas variáveis StartTime e EndTime, a intervalos fixos definidos pela variável TimeInterval.

NOTA: As variáveis também podem ser definidas em tempo de execução a partir da chamada do método **SetVariableValue** da Consulta.

16.3.8 GroupBy

A Texto correspondente ao argumento da cláusula **GROUP BY** do código SQL da consulta. Esta propriedade é somente de leitura, mas pode ser modificada em tempo de execução.

16.3.9 HasFocus

Esta propriedade determina que o objeto selecionado está com o foco. Esta propriedade é acessível apenas em tempo de execução.

16.3.10 Having

A Texto correspondente ao argumento da cláusula **HAVING** do código SQL da consulta. Esta propriedade é normalmente utilizada com a propriedade **GroupBy**. Esta propriedade é somente de leitura, mas pode ser modificada em tempo de execução.

16.3.11 MouseOver

A propriedade **MouseOver** informa se o ponteiro do mouse encontra-se em cima da Tela. Caso afirmativo, a propriedade **MouseOver** é habilitada; do contrário, será False. Esta propriedade é somente de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

16.3.12 MouseOverChild

A propriedade **MouseOverChild** informa se o ponteiro do mouse encontra-se em cima de um dos objetos inseridos na Tela. Caso afirmativo, a propriedade **MouseOverChild** é habilitada. Caso contrário, para False. Esta propriedade é para somente leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

16.3.13 OrderBy

A Texto correspondente ao argumento da cláusula **ORDER BY** do código SQL da consulta. Esta propriedade é somente de leitura, mas pode ser modificada em tempo de execução.

16.3.14 Screen

Retorna a Tela-pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

16.3.15 SQL

A Contém o código SQL especificado para a consulta. Esta propriedade é somente para leitura, mas pode ser modificada em tempo de execução.

16.3.16 Table

A A propriedade **Table** contém as tabelas que serão consultadas (por exemplo, Alarms é a tabela de alarmes ou eventos). Corresponde ao argumento da cláusula **FROM** do código SQL da consulta. Esta propriedade é somente de leitura, mas pode ser modificada em tempo de execução.

16.3.17 Where

A Determina a condição da consulta que filtra os registros da tabela a serem visualizados, ou seja, só serão visualizados os registros que satisfizerem a condição. Corresponde ao argumento **WHERE** do código SQL da consulta. Esta propriedade é somente de leitura, mas pode ser modificada em tempo de execução.

CAPÍTULO

17 E3Browser

Esta seção contém informações sobre eventos, métodos e propriedades do objeto E3Browser.

17.1 Eventos

Esta seção contém informações sobre os eventos do objeto E3Browser.

17.1.1 KeyPress

KeyPress (*KeyAscii*)

Ocorre quando o E3Browser tem o foco de teclado e o usuário pressiona uma tecla que corresponde a um caractere que pode ser mostrado em tela (uma tecla ANSI, de código indicado na variável *KeyAscii*). Ou seja, o evento ocorre quando alguma das teclas a seguir são pressionadas:

- Qualquer caractere do teclado que possa ser impresso.
- Tecla CTRL combinada com qualquer caractere do alfabeto padrão.
- Tecla CTRL combinada com qualquer caractere especial.
- Tecla BACKSPACE.
- Tecla ESC.

Este evento não ocorre nas seguintes condições:

- Pressionando a tecla TAB.
- Pressionando a tecla ENTER.
- Pressionando a tecla DEL (essa tecla não é ANSI).

- Pressionando as setas do teclado.
- Quando uma tecla faz com que o foco vá de um objeto a outro.

Enquanto um usuário pressiona uma tecla que produz um código ANSI, o E3Browser recebe os eventos **KeyDown** e **KeyPress** repetidamente. Quando o usuário libera a tecla, o evento **KeyUp** ocorre. Para monitorar o estado físico do teclado ou manipular teclas não reconhecidas pelo evento **KeyPress** (como teclas de função, navegação, etc.), utiliza-se os eventos **KeyDown** e **KeyUp**.

17.1.2 OnDrawRow

OnDrawRow(*bSelected*, *nLine*, *cTextColor*, *cBackColor*)

Este evento passa quatro parâmetros. *bSelected* indica se a linha está selecionada; *nLine* indica o número da linha sendo desenhada; *cTextColor* indica a cor do texto da linha; e *cBackColor* indica a cor de fundo do texto. Se a cor for modificada dentro deste evento, esta modificação será usada pelo E3Browser no desenho da linha. Outra modificação importante é que se o método **GetColumnValue** for chamado de dentro do evento, os valores retornados serão os da linha sendo desenhada, e não os da linha selecionada.

17.1.3 MouseMove

MouseMove()

Ocorre quando o ponteiro do mouse se move sobre o objeto E3Browser.

17.2 Métodos

Esta seção contém informações sobre os métodos do objeto E3Browser.

17.2.1 ClearFields

ClearFields()

Limpa a formatação de colunas e linhas do E3Browser. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Browser1").ClearFields()  
End Sub
```

17.2.2 GetColumnValue

GetColumnValue(*Index*)

Retorna o valor de uma célula, na coluna informada e na linha selecionada. Este método possui o parâmetro *index*, que determina o índice da coluna desejada.

Exemplo:

```
Sub E3Browser1_Db1Click()  
    Screen.Item("Texto1").Value = GetColumnValue(0)  
    Screen.Item("Texto2").Value = GetColumnValue(1)  
    Screen.Item("Texto3").Value = GetColumnValue(2)  
    Screen.Item("Texto4").Value = GetColumnValue(3)  
End Sub
```

17.2.3 Requery

Requery()

O método **Requery** atualiza a consulta utilizando a configuração atual da consulta e retorna os dados para o E3Browser. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Browser1").Requery()  
End Sub
```

17.2.4 RetrieveE3QueryFields

RetrieveE3QueryFields()

O método **RetrieveE3QueryFields** lê a estrutura de dados da consulta e atualiza a formatação do E3Browser com os campos definidos na consulta. Se for bem sucedido retorna True. Caso contrário, retorna False. Este método é especialmente útil quando se precisa usar um único E3Browser para exibir dados de diferentes tabelas ou consultas. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Browser1").RetrieveE3QueryFields()  
End Sub
```

17.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto E3Browser.

17.3.1 AllowColumnResize

Habilita ou desabilita a configuração do tamanho das colunas da grade do objeto, em tempo de execução. Se esta propriedade estiver configurada para False, o tamanho das colunas é fixo e não poderá ser modificado.

17.3.2 AllowRowResize

Habilita ou desabilita a configuração do tamanho das linhas da grade do objeto, em tempo de execução. Se esta propriedade estiver configurada para False, o tamanho das linhas é fixo e não poderá ser modificado.

17.3.3 ColumnWidth

9 Determina a largura das colunas do E3Browser, em pixels.

17.3.4 CurSel

9 Indica a posição atual do cursor do E3Browser, ou seja, o índice da linha em que ele está posicionado.

17.3.5 E3Query

 Retorna o objeto Consulta do E3Browser, para que se possa acessar suas propriedades.

17.3.6 Enabled

Habilita ou desabilita o E3Browser. Se configurado como True, pode-se usar o *scroll* e selecionar linhas do E3Browser. Do contrário, nenhum clique do mouse no E3Browser terá efeito.

17.3.7 Fields

 Retorna o objeto coleção que contém a lista com todos os campos da tabela, tornando possível a sua referência através dos itens dessa coleção. O valor padrão desta propriedade é vazio. Exemplo:

```

Sub E3Browser1_Click()
    Troca a cor do campo Campo1
    Set fields = Screen.Item("E3Browser").Fields
    Set campo1 = fields.Item("Campo1")
    campo1.BkColor = RGB(255, 0, 0) ' Vermelho
    ' Mostra quantos campos tem o E3Browser
    MsgBox fields.Count
    ' Mostra o nome dos campos do E3Browser
    For Each field In fields
        MsgBox field.Name
    Next
End Sub

```

17.3.8 FixedBkColor

 Especifica a cor do fundo da primeira coluna do E3Browser. O valor padrão desta propriedade é a cor bege (**RGB(236, 233, 216)**).

17.3.9 FixedColumnWidth

 Determina a largura da primeira coluna do E3Browser, em pixels. O valor padrão desta propriedade é 30 pixels.

17.3.10 FixedRowFont

 Determina a fonte (texto) utilizada na linha de cabeçalho do E3Browser. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada somente via Studio. O valor padrão desta propriedade é a fonte Arial.

17.3.11 FixedRowHeight

 Determina a altura (em pixels) da linha de cabeçalho do E3Browser. O valor padrão desta propriedade é 20.

17.3.12 FixedTextColor

 Muda a cor do cabeçalho do E3Browser.

17.3.13 Frame

 Retorna o quadro-pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

17.3.14 GridBkColor

 Determina a cor de fundo da área de dados do E3Browser. O valor padrão desta propriedade é a cor branca (RGB(255, 255, 255)).

17.3.15 GridFont

 Determina a fonte utilizada nos textos da área de dados do E3Browser. O valor padrão desta propriedade é a fonte Arial. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada somente via Studio.

17.3.16 GridLineColor

 Determina a cor das linhas da grade de dados do E3Browser. O valor padrão desta propriedade é cinza (RGB(192, 192, 192)).

17.3.17 GridLinesType

 Determina o tipo de linhas a serem desenhadas na grade de dados do E3Browser.

Opções disponíveis para GridLinesType

OPÇÃO	DESCRIÇÃO
0 - GLNone	Sem linhas separadoras.
1 - GLHorz	Somente as linhas horizontais (padrão).
2 - GLVert	Somente as linhas verticais.
3 - GLBoth	Linhas verticais e horizontais.

17.3.18 HasFocus

Em True, indica que o E3 Browser está com o foco do teclado.

17.3.19 Height

 Essa propriedade determina a altura do objeto, em unidades Himetric.

17.3.20 Layer

 Indica o número da camada onde o E3Browser está posicionado.

17.3.21 MouseOver

 A propriedade **MouseOver** informa se o ponteiro do mouse encontra-se em cima da Tela. Caso afirmativo, a propriedade **MouseOver** é habilitada; do contrário, será False. Esta propriedade é somente de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

17.3.22 MouseOverChild

 A propriedade **MouseOverChild** informa se o ponteiro do mouse encontra-se em cima de um dos objetos inseridos na Tela. Caso afirmativo, a propriedade **MouseOverChild** é habilitada. Caso contrário, será False. Esta propriedade é somente de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

17.3.23 RefreshTime

 Especifica o tempo de atualização da consulta em relação ao banco de dados. Através desta propriedade é possível verificar as atualizações dos dados no histórico relacionado referente a um determinado tempo estipulado (em milissegundos). Quando a propriedade **RefreshTime** for igual a 0, não há atualização dos dados, e eles permanecem inalterados.

17.3.24 RowHeight

 Define a altura das linhas da E3Browser, em pixels. O valor padrão desta propriedade é 20.

17.3.25 Screen

 Contém uma referência para o objeto Tela onde o E3 Browser está incluído.

17.3.26 SelectRow

 Estabelece a possibilidade de seleção das linhas do E3 Browser. Se estiver configurada para True, será possível selecionar as linhas; do contrário, as linhas não podem ser selecionadas.

17.3.27 SourceQuery

 Contém a referência para o objeto Query ao qual o E3Browser está associado.

NOTA: Para trocar a consulta do E3Browser por meio de um script (caso a nova consulta modifique os campos da consulta original), além de alterar a propriedade **SourceQuery**, é preciso utilizar os métodos **RetrieveE3QueryFields** e **Requery**.

17.3.28 TabStop

 Indica se o objeto E3Browser receberá o foco do teclado quando o usuário utiliza a tecla TAB para trocar entre os diversos campos em uma Tela.

17.3.29 TextBkColor

 Especifica a cor de fundo das células de dados do E3Browser. O valor padrão desta propriedade é a cor branca (**RGB(255, 255, 255)**).

17.3.30 TextColor

 Especifica a cor do texto do E3Browser. O valor padrão desta propriedade é a cor preta (**RGB(0,0,0)**).

17.3.31 Tip

A Contém um texto de ajuda apresentado como dica quando o usuário coloca o ponteiro do mouse sobre o objeto.

17.3.32 TitleTipBkColor

 Especifica a cor do fundo do texto da dica do E3Browser. O valor padrão desta propriedade é a cor preta (**RGB(0, 0, 0)**).

17.3.33 TitleTipTextColor

 Especifica a cor do texto da dica do E3Browser. O valor padrão desta propriedade é a cor cinza (**RGB(204, 204, 204)**).

17.3.34 ToolbarBkColor

 Especifica a cor de fundo da barra de funções do E3 Browser. O valor padrão desta propriedade é bege (**RGB(236, 233, 216)**).

17.3.35 ToolbarFont

A Determina a fonte utilizada nos textos da barra de funções do E3 Browser. Esta propriedade não pode ser utilizada em scripts ou em associações, sendo configurada somente via Studio.

17.3.36 ToolbarForeColor

 Especifica a cor de frente da barra de funções do E3 Browser. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

17.3.37 Visible

Habilita ou desabilita a visibilidade do campo selecionado no E3Browser. Se a propriedade estiver configurada para True, o campo será visível no E3Browser. Caso contrário, o campo não é mostrado no E3Browser, em tempo de execução. O valor padrão é True.

17.3.38 Width

 Determina a largura do objeto, em unidades Himetric.

17.3.39 X

 A propriedade X define a coordenada horizontal esquerda, em unidades Himetric, do objeto.

17.3.40 Y

 Esta propriedade define a coordenada vertical superior, em unidades Himetric, do objeto.

17.4 Campos do E3Browser

Esta seção contém informações sobre propriedades dos campos do objeto E3Browser. Estes campos não possuem eventos nem métodos associados.

17.4.1 Propriedades

Esta seção contém informações sobre as propriedades dos campos do objeto E3Browser.

17.4.1.1 BkColor

 Determina a cor de fundo do campo do E3Browser. O valor padrão é a cor configurada no Windows para o item **Janela** do **Painel de Controle (Painel de Controle - Vídeo - Aparência - Avançada)**.

17.4.1.2 Color

 Retorna a cor do texto do campo. O valor padrão dessa propriedade é preto (**RGB(0, 0, 0)**).

17.4.1.3 Format

A Configura o formataador usado na coluna.

17.4.1.4 Name

A Retorna o nome do campo.

17.4.1.5 Visible

Habilita ou desabilita a visibilidade do campo selecionado na consulta do E3Browser. Se a propriedade estiver configurada para True, o campo será visível no E3Browser. Caso contrário, o campo não é mostrado no E3Browser, em tempo de execução. O valor padrão é True.

17.4.1.6 Width

9 Retorna a largura do campo, em unidades Himetric.

CAPÍTULO

18

E3Chart

Esta seção contém informações sobre eventos, métodos e propriedades do objeto E3Chart.

18.1 Eventos

Esta seção contém informações sobre os eventos do objeto E3Chart.

18.1.1 OnCursorChange

OnCursorChange()

Ocorre quando o cursor do E3Chart muda de posição. Por exemplo, pode-se criar um script para este evento quando for necessário mostrar os valores de posição do cursor na Tela. Exemplo:

```
Sub E3Chart1_OnCursorChange()  
    Set Chart = Application.GetFrame("").Screen.Item("E3Chart1")  
    Set Pena = Chart.Pens.Item(0)  
    ' O objeto Text1 deve mostrar a posição atual do cursor  
    Set Texto = Application.GetFrame("").Screen.Item("Text1")  
    If Pena.GetCursorPos(aa, bb) Then  
        Texto.Value = "Posição X = " & aa & "; Posição Y = " & bb  
    End If  
End Sub
```

18.1.2 OnLegendClick

OnLegendClick(*Row, Col, RowData*)

Ocorre quando o usuário clica em uma linha da legenda. Os parâmetros *Row* e *Col* indicam, respectivamente, a linha e a coluna clicadas. O parâmetro *RowData* é o índice da pena da legenda onde ocorreu o clique. Exemplo:

```
Sub E3Chart1_OnLegendClick(Row, Col, RowData)  
    Set texto = Screen.Item("Text01")  
    texto.Value = Legend.Item(col).Name & " " & _  
        Pens.Item(RowData).name  
End Sub
```

18.1.3 OnQueryFinish

OnQueryFinish()

Ocorre quando uma ou mais consultas são encerradas no objeto. Quando o evento é gerado, a chamada do método **FitAll** ou **FitPen** pode causar problemas caso se esteja utilizando a consulta automática, já que o método ativa outras consultas até que todos os dados sejam lidos. Neste caso, é recomendado que o valor passado pelos parâmetros destes métodos seja 1, o que enquadra verticalmente as penas.

18.2 Métodos

Esta seção contém informações sobre os métodos do objeto E3Chart.

18.2.1 CopyConfig

CopyConfig(SourceChart, [Flags])

O método **CopyConfig** copia as configurações de um E3Chart para outro. O parâmetro *SourceChart* indica o E3Chart de origem cujas propriedades serão copiadas para o E3Chart que chamou o método.

NOTA: No caso dos Relatórios, o método **CopyConfig** irá funcionar somente com penas do tipo Histórica.

Por exemplo, para copiar a configuração de um E3Chart que está numa Tela (ScreenChart) para outro que está dentro de um relatório (ReportChart), o seguinte script pode ser adicionado no objeto E3Report associado ao relatório:

```
Sub OnBeforePrint
  Set Chart = _
  Report.Sections("PageHeader").Controls("ReportChart")
  Chart.CopyConfig(Application.GetFrame()._
  Screen.Item("ScreenChart"))
  Chart.LoadData()
  Chart.FitAll()
End Sub
```

NOTA: este método também possui o parâmetro opcional e não usado *Flags*, somente para fins de compatibilidade com versões anteriores.

18.2.2 FitAll

FitAll([FitStyle])

Enquadra todas as penas no E3Chart. O parâmetro opcional *FitStyle* indica o modo de enquadramento das penas em tempo de execução:

- **0**: enquadra ambos os eixos ao mesmo tempo.
- **1**: enquadra apenas o eixo vertical.
- **2**: enquadra apenas o eixo horizontal.

Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").FitAll()  
End Sub
```

18.2.3 FitPen

FitPen(Pen, [FitStyle])

Enquadra uma pena no E3Chart especificada pelo índice ou pelo nome. O parâmetro *Pen* define a pena que será enquadrada no E3Chart (índice ou nome da pena). O parâmetro opcional *FitStyle* indica o modo de enquadramento das penas em tempo de execução:

- **0**: enquadra ambos os eixos ao mesmo tempo.
- **1**: enquadra apenas o eixo vertical.
- **2**: enquadra apenas o eixo horizontal.

Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    Chart.FitPen(1)  
    Chart.FitPen("Pen1", 1)  
    ' Enquadra a Pen1 somente na vertical  
End Sub
```

18.2.4 LoadData

LoadData()

Carrega os dados no E3Chart. Este método é especialmente usado para carregar os dados antes de imprimir, quando usado em um objeto Relatório. Exemplo:

```
Sub CommandButton1_Click()  
    MsgBox Screen.Item("E3Chart1").LoadData()  
End Sub
```

NOTA: O método **LoadData** será síncrono somente se a pena não estiver em modo **Automático**.

18.2.5 ResetConfig

ResetConfig([Flags])

Remove todas as configurações ajustadas em um E3Chart, devolvendo-o ao estado inicial. Exemplo:

```
Sub E3Chart1_OnStartRunning()  
    ' Ao iniciar E3Chart1, remove todas as configurações  
    ResetConfig()  
End Sub
```

NOTA: este método também possui o parâmetro opcional e não usado *Flags*, somente para fins de compatibilidade com versões anteriores.

18.2.6 ShowCursor

ShowCursor()

Ativa o modo **Pesquisa de Intervalo**. Em tempo de execução, este recurso pode ser acessado através do clique com o botão direito do mouse sobre o objeto e selecionando a opção **Pesquisar Intervalos** do menu de contexto.

18.2.7 ZoomIn

ZoomIn()

O método **ZoomIn** aumenta o zoom no E3Chart, isto é, aproxima a visualização das penas. Em tempo de execução, este recurso pode ser acessado através do clique com o botão direito do mouse sobre o objeto e selecionando a opção **Mais Zoom** do menu de contexto. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").ZoomIn()  
End Sub
```

18.2.8 ZoomOut

ZoomOut()

O método **ZoomOut** diminui o zoom no E3Chart, isto é, afasta a visualização das penas no E3Chart. Em tempo de execução, este recurso pode ser acessado através do clique com o botão direito do mouse sobre o E3Chart e selecionando a opção **Menos Zoom**.

Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").ZoomOut()  
End Sub
```

18.3 Propriedades

Esta seção contém informações sobre as propriedades do objeto E3Chart.

18.3.1 Axes

 Retorna a coleção de eixos do E3Chart. A partir daí, as propriedades da coleção de eixos podem ser modificadas.

18.3.2 BackColor

 Determina a cor de fundo do E3 Chart. Para que esta cor apareça, a propriedade **ShowBackground** deve estar configurada como True. O valor padrão desta propriedade é bege (**RGB(236, 233, 216)**).

18.3.3 CursorBegin

 Define a posição do cursor inicial, entre 0 e 1. É necessário executar o método **ShowCursor** ou habilitar a opção **Pesquisa de Intervalos** para que os cursores apareçam.

18.3.4 CursorColor

 Estabelece a cor do cursor de pesquisa de intervalo. O valor padrão é vermelho (**RGB(255, 0, 0)**).

18.3.5 CursorEnd

9 Define a posição do cursor final, entre 0 e 1. É necessário executar o método **ShowCursor** ou habilitar a opção **Pesquisa de Intervalos** para que os cursores apareçam.

18.3.6 CursorLineStyle

 Estilo de linha do cursor de pesquisa de intervalo. As opções disponíveis são:

Opções disponíveis para CursorLineStyle

OPÇÃO	DESCRIÇÃO
0 - LS_Solid	Aplica uma linha sólida no cursor de intervalo do E3Chart.
1 - LS_Dash	Aplica uma linha tracejada no cursor de intervalo do E3Chart.
2 - LS_Dot	Aplica uma linha pontilhada no cursor de intervalo do E3Chart.
3 - LS_Dashdot	Aplica uma linha traço e ponto no cursor de intervalo do E3Chart.
4 - LS_Dashdotdot	Aplica uma linha traço ponto ponto no cursor de intervalo do E3Chart.
5 - LS_Null	Aplica uma linha invisível no cursor de intervalo do E3Chart.

18.3.7 CursorLineWidth

9 Estabelece a largura do cursor de intervalo.

18.3.8 Enabled

Habilita o objeto ActiveX no projeto. O valor padrão desta propriedade é True.

18.3.9 ForeColor

 Determina a cor de frente do E3 Chart. O valor padrão desta propriedade é preto (RGB(0, 0, 0)).

18.3.10 Frame

 Retorna o quadro pai do objeto. Esta propriedade é acessível apenas em tempo de execução.

18.3.11 GridBkColor

 Determina a cor de fundo da grade do E3Chart. O valor padrão desta propriedade é branco (RGB(255, 255, 255)). Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    Old = E3Chart1.GridBkColor  
    MsgBox "Próxima"  
    E3Chart1.GridBkColor = RGB(0, 255, 0)  
    MsgBox "Voltar"  
    E3Chart1.GridBkColor = Old  
End Sub
```

18.3.12 HasFocus

Esta propriedade determina que o objeto selecionado está com o foco. Esta propriedade é acessível apenas em tempo de execução.

18.3.13 Height

 A propriedade **Height** define a altura do E3 Chart, em unidades Himetric.

18.3.14 HorAxisTitle

 Determina o título do eixo horizontal principal. Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1= Screen.Item("E3Chart1")  
    Old = E3Chart1.HorAxisTitle  
    MsgBox "Próxima"  
    E3Chart1.HorAxisTitle = "!Teste"  
    MsgBox "Voltar"  
    E3Chart1.HorAxisTitle = Old  
End Sub
```

18.3.15 HorGrid

 Determina o tipo de linha que será aplicada na grade horizontal do E3Chart.

Opções disponíveis para HorGrid

OPÇÃO	DESCRIÇÃO
0 - Solid	Aplica uma linha sólida na grade horizontal do E3Chart.
1 - Dash	Aplica uma linha tracejada na grade horizontal do E3Chart.
2 - Dot	Aplica uma linha pontilhada na grade horizontal do E3Chart (padrão).
3 - Dashdot	Aplica uma linha traço e ponto na grade horizontal do E3Chart.
4 - Dashdotdot	Aplica uma linha traço ponto ponto na grade horizontal do E3Chart.
5 - Invisible	Aplica uma linha invisível na grade horizontal do E3Chart.

Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    Old = E3Chart1.HorGrid  
    For i = 0 To 5  
        E3Chart1.HorGrid = i  
        MsgBox "E3Chart1.HorGrid =" & CStr(i)  
    Next  
    MsgBox "voltar"  
    E3Chart1.HorGrid = Old  
End Sub
```

18.3.16 HorGridColor

 Determina a cor da grade horizontal do objeto E3Chart. O valor padrão desta propriedade é cinza (**RGB(192, 192, 192)**). Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    Old = E3Chart1.HorGridColor = RGB(255, 0, 0)  
    MsgBox "Próxima"  
    E3Chart1.HorGridColor = RGB(255, 0, 0)  
    MsgBox "Próxima"  
    E3Chart1.HorGridColor = RGB(0, 0, 255)  
    MsgBox "voltar"  
    E3Chart1.HorGridColor = Old  
End Sub
```

18.3.17 HorMinorTicks

9 Determina o número de subdivisões das escalas horizontais da grade. O valor padrão desta propriedade é 1. Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    Old = E3Chart1.HorMinorTicks  
    For i = 0 To 5  
        E3Chart1.HorMinorTicks = i  
        MsgBox "Próximo valor"  
    Next  
    E3Chart1.HorMinorTicks = Old  
End Sub
```

18.3.18 HorScaleBegin

9 Determina o valor inicial aplicado na escala horizontal principal da grade. Este valor pode ser numérico para E3Charts XY, ou Data para E3Charts com escala de tempo fixa. Para E3Charts de tempo real, esta propriedade não é aplicada, sendo então utilizada a propriedade **TimeSpan**.

18.3.19 HorScaleEnd

9 Determina o valor final aplicado na escala horizontal principal da grade. Este valor pode ser numérico para E3Charts XY, ou Data para E3Charts com escala de tempo fixa. Para E3Charts de tempo real, esta propriedade não é aplicada, sendo então utilizada a propriedade **TimeSpan**. Exemplo:

```
Sub ComboBox1_Change()  
    ' Define qual consulta deseja mostrar  
    current_query_index = ListIndex  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    i = 0  
    For Each query In E3Chart1.Queries  
        If i = current_query_index Then  
            query.FieldFilter(0) = "" ' Traz tudo  
            Set current_query = query  
        Else  
            query.FieldFilter(0) = "<0" ' Não traz nada para  
            ' não ficar lento  
        End If  
        i = i + 1  
    Next  
    ' Só mostra as penas que usam a consulta atual  
    For Each pen In E3Chart1.Pens  
        pen.Visible = (pen.QueryName = current_query.Name)  
    Next  
    ' Atualiza as consultas  
    E3Chart1.Queries.UpdateData()  
    Screen.Item("E3Chart1").HorScaleBegin = Now - 0.001  
    Screen.Item("E3Chart1").HorScaleEnd = Now  
End Sub
```

18.3.20 HorScaleFormat

A Contém um texto que representa uma máscara dentro da qual os valores da escala horizontal serão mostrados. Esta máscara pode representar vários tipos de valores:

- **Geral:** Não possui formatação específica, adaptando-se automaticamente ao valor especificado.
- **Número:** Apresenta números com parte inteira e fracionária. O usuário pode optar por até 15 casas decimais, por usar ou não um separador de milhares, e por apresentar números negativos com sinal ou entre parênteses. Para números muito grandes ou muito pequenos, recomenda-se utilizar o formato Científico.
- **Data:** Apresenta valores numéricos de data e hora (quando válidos). Para representar apenas a hora, use o formato equivalente.
- **Hora:** Apresenta valores numéricos de hora e data (quando válidos). Para representar apenas a data, use o formato equivalente.
- **Porcentagem:** Multiplica o número por 100 e adiciona o símbolo de porcentagem. Admite até 15 casas decimais.
- **Científico:** Apresenta o número em notação de mantissa e expoente. Ideal para números de magnitude variada. Admite até 15 casas decimais.
- **Especial:** Permite formatar números inteiros em bases não-decimais (hexadecimal, octal ou binária, por exemplo).
- **Outro:** Permite editar diretamente o código de formatação desejado, ou selecionar um formato criado anteriormente.

A máscara destes formatos, como exemplificadas no campo **Tipo**, será exibida na Janela de Propriedades (por exemplo, d/M/yy H:mm, 0E-00, etc.).

18.3.21 HorTickUnit

9 Determina o número de subdivisões entre as marcas da grade. Quando esta propriedade estiver com valor 0, o espaçamento será automático. Exemplo:

```
Sub SubCommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    Old = E3Chart1.HorTickUnit  
    For i = 0 To 30 Step 10  
        E3Chart1.HorTickUnit = i  
        MsgBox "E3Chart1.HorTickUnit = " & CStr(i)  
    Next  
    MsgBox "Voltar"  
    E3Chart1.HorTickUnit = Old
```

18.3.22 Layer

9 Indica o número da camada onde o E3Chart está posicionado.

18.3.23 Legend

 Retorna o objeto legenda do E3Chart. A partir daí, as propriedades da legenda podem ser modificadas.

18.3.24 MouseMode

 Seleciona uma das opções de runtime do menu do E3Chart. Os valores possíveis para essa propriedade são:

- **0 - MouseModeZoom**: Coloca o mouse em modo zoom por área selecionada. Esta opção está disponível nos gráficos de escala numérica em XY e escala fixa. Equivale a selecionar a opção **Caixa de Zoom** do menu de runtime do E3Chart.
- **1 - MouseModePan**: Coloca o mouse em modo de movimentação das escalas. Equivale a selecionar a opção **Mover** do menu de runtime do E3Chart.
- **2 - MouseModePanH**: Coloca o mouse em modo de movimentação das escalas somente na direção horizontal. Equivale a selecionar a opção **Mover horizontalmente** do menu de runtime do E3Chart.
- **3 - MouseModeSearch**: Coloca o mouse em modo de pesquisa de valores nos dados das penas. Equivale a selecionar a opção **Pesquisar** do menu de runtime do E3Chart.
- **4 - MouseModeCursors**: Habilita a opção de pesquisa de intervalos de tempo. Equivale a selecionar a opção **Pesquisar Intervalos** do menu de runtime do E3Chart.

18.3.25 MouseOver

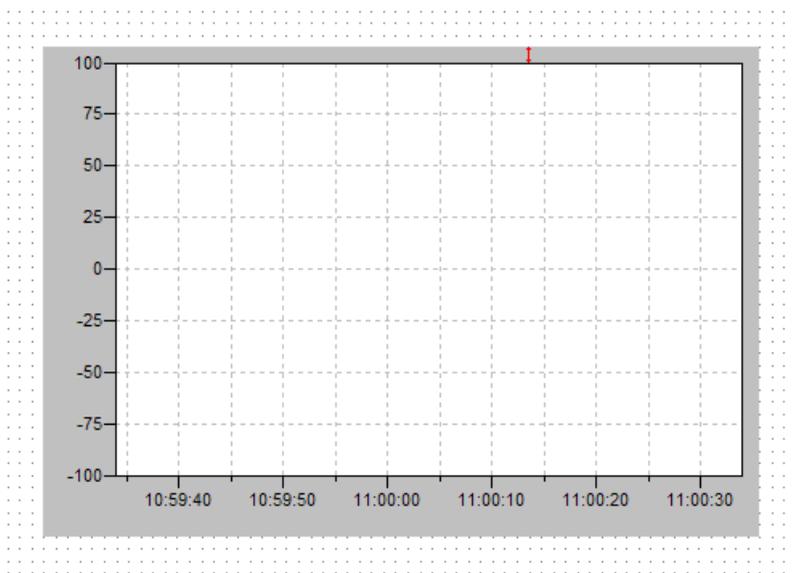
 A propriedade **MouseOver** informa se o ponteiro do mouse encontra-se em cima da Tela. Caso afirmativo, a propriedade **MouseOver** é habilitada; do contrário, será False. Esta propriedade é somente de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

18.3.26 MouseOverChild

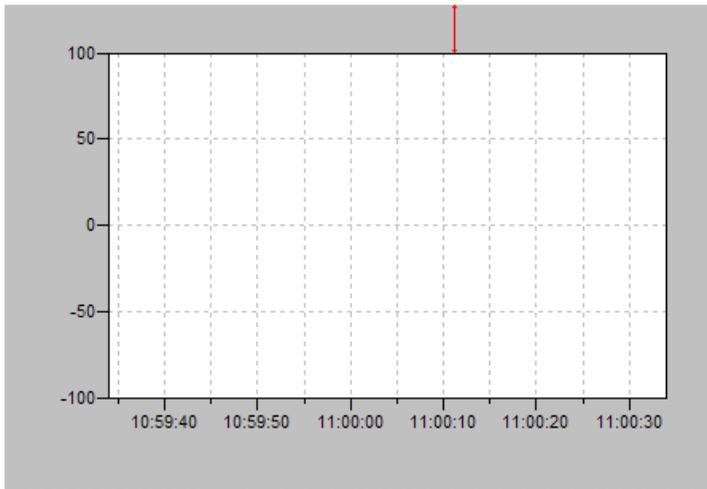
A propriedade **MouseOverChild** informa se o ponteiro do mouse encontra-se em cima de um dos objetos inseridos na Tela. Caso afirmativo, a propriedade **MouseOverChild** é habilitada. Caso contrário, será False. Esta propriedade é somente de leitura, e é acessível apenas em tempo de execução. O valor padrão desta propriedade é False.

18.3.27 Padding

9 Esta propriedade determina a distância, em pixels, entre o gráfico e a borda do objeto E3Chart, conforme pode ser observado nas figuras seguintes através das setas vermelhas. O valor padrão desta propriedade é 10. Exemplo:



Propriedade Padding igual a 10



Propriedade Padding igual a 30

18.3.28 Pens

Retorna o objeto Coleção de Penas do E3Chart. O objeto Coleção de Penas é utilizado para inserir, remover ou acessar as penas disponíveis no E3Chart. Esta propriedade é somente de leitura. Exemplo:

```
Sub CommandButton1_Click()
    For Each pen In Screen.Item("E3Chart1").Pens
        pen.Visible = True
    Next
End Sub
```

18.3.29 Queries

Retorna o objeto Coleção de Consultas dentro do E3Chart. O objeto Coleção de Consultas é utilizado para inserir, remover ou acessar as consultas disponíveis no E3 Chart. Esta propriedade é somente de leitura.

18.3.30 RefreshTime

9 Esta propriedade determina o tempo de atualização do E3Chart. Exemplo:

```
Sub E3Chart1_OnStartRunning()  
    MsgBox RefreshTime  
End Sub
```

18.3.31 ScaleFont

A Determina a fonte do texto utilizada na grade. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").ScaleFont = "Times New Roman"  
    Screen.Item("E3Chart1").ScaleFont.Size = 12  
    Screen.Item("E3Chart1").ScaleFont.Italic = true  
End Sub
```

18.3.32 Screen

 Contém uma referência para o objeto Tela onde o E3Chart está incluído.

18.3.33 ShowBackground

Habilita ou desabilita a visualização do fundo do gráfico. Se a propriedade for igual a True, o fundo do gráfico é mostrado. Caso contrário, o gráfico fica com o fundo transparente. A cor escolhida na propriedade **BackColor** não aparece se esta propriedade estiver em False (opção padrão). Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    E3Chart1.ShowBackground = Not E3Chart1.ShowBackground  
End Sub
```

18.3.34 ShowBottomScale

Se a propriedade estiver configurada para True, o eixo horizontal principal é mostrado na base da grade. Caso contrário, não é mostrado. O valor padrão é True. Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    oldBottomScale = E3Chart1.ShowBottomScale  
    MsgBox "Mostrar eixo"  
    E3Chart1.ShowBottomScale = True  
    MsgBox "Esconder eixo"  
    E3Chart1.ShowBottomScale = False  
    MsgBox "Voltar..."  
    E3Chart1.ShowBottomScale = oldBottomScale  
End Sub
```

18.3.35 ShowGridBackground

☑ Habilita ou desabilita a visualização do fundo da grade. Se a propriedade for igual a True (padrão), o fundo da grade é mostrado. Caso contrário, a grade fica com o fundo transparente. A cor escolhida na propriedade **GridBkColor** não aparece se esta propriedade estiver em False. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart1 = Screen.Item("E3Chart1")  
    Chart1.ShowGridBackground = Not Chart1.ShowGridBackground  
End Sub
```

18.3.36 ShowLeftScale

☑ Se a propriedade estiver configurada para True, o eixo vertical principal será mostrado à esquerda da grade. Caso contrário, fica invisível. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    Chart.ShowLeftScale = Not Chart.ShowLeftScale  
End Sub
```

18.3.37 ShowRightScale

☑ Se a propriedade estiver configurada para True, o eixo vertical principal da escala será mostrado à direita da grade. Caso contrário, fica invisível. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    Chart.ShowRightScale = Not Chart.ShowRightScale  
End Sub
```

18.3.38 ShowPopupMenu

☑ Habilita ou desabilita a opção de mostrar o menu de runtime do E3Chart. Se o valor for igual a True, o menu será mostrado quando o usuário clicar com o botão direito do mouse sobre o gráfico do E3Chart. Se o valor for igual a False, o menu não é mostrado. O valor padrão dessa propriedade é True.

18.3.39 ShowTitle

☑ Se a propriedade estiver configurada para True, o título principal do E3Chart é visível. Caso contrário, fica invisível. A propriedade **Title** contém o título que será mostrado no E3Chart. Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    oldTitle = E3Chart1.Title  
    oldShowTitle = E3Chart1.ShowTitle  
    E3Chart1.Title = "Teste!"  
    MsgBox "Mostrar"  
    E3Chart1.ShowTitle = True  
    MsgBox "Esconder"
```

```
E3Chart1.ShowTitle = False
MsgBox "Voltar"
E3Chart1.Title = oldTitle
E3Chart1.ShowTitle = oldShowTitle
End Sub
```

18.3.40 ShowTopScale

☑ Se a propriedade estiver configurada para True, o eixo horizontal principal é mostrado no topo da grade. Caso contrário, não é mostrado. O valor padrão é False.

Exemplo:

```
Sub CommandButton1_Click()
    Set Chart = Screen.Item("E3chart1")
    Chart.ShowTopScale = Not Chart.ShowTopScale
End Sub
```

18.3.41 TabStop

☑ Esta propriedade determina a utilização da tecla TAB no sistema. Se o campo estiver True, é possível utilizar a tecla; caso contrário, a tecla não pode ser utilizada.

18.3.42 TimeSpan

 Indica a escala de tempo que aparece no eixo horizontal principal do E3Chart quando esta for configurada para mostrar a escala de tempo real. O valor desta propriedade é sempre em segundos. O valor padrão desta propriedade é 60. Exemplo:

```
Sub RetanguloArr1_Click()
    MsgBox Screen.Item("E3Chart1").TimeSpan
End Sub
```

18.3.43 Tip

A Contém um texto de ajuda apresentado como dica quando o usuário coloca o ponteiro do mouse sobre o objeto.

18.3.44 Title

A Determina o título principal do E3Chart. Para que o título apareça no E3Chart, a propriedade **ShowTitle** deve ser configurada para True.

18.3.45 TitleFont

A Determina a fonte do título principal do E3Chart. Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    E3Chart1.Title = "Teste"  
    E3Chart1.ShowTitle = True  
    MsgBox "Troca fonte"  
    E3Chart1.TitleFont = "Times New Roman"  
    MsgBox "Troca tamanho"  
    E3Chart1.TitleFont.Size = 20  
End Sub
```

18.3.46 VerAxisTitle

A Determina o título do eixo vertical principal. Exemplo:

```
Sub ComboBox1_Change()  
    Screen.Item("E3Chart1").VerAxisTitle = "Titulo1"  
End Sub
```

18.3.47 VerGrid

 Determina o tipo de linha que será aplicada na grade vertical do E3Chart.

Opções disponíveis para VerGrid

OPÇÃO	DESCRIÇÃO
0 - Solid	Aplica uma linha sólida na grade vertical do E3Chart.
1 - Dash	Aplica uma linha tracejada na grade vertical do E3Chart.
2 - Dot	Aplica uma linha pontilhada na grade vertical do E3Chart (padrão).
3 - Dashdot	Aplica uma linha traço e ponto na grade vertical do E3Chart.
4 - Dashdotdot	Aplica uma linha traço ponto ponto na grade vertical do E3Chart.
5 - Invisible	Aplica uma linha invisível na grade vertical do E3Chart.

Exemplo:

```
Sub E3Chart1_OnStartRunning()  
    VerGrid = 2  
End Sub
```

18.3.48 VerGridColor

 Determina a cor da linha do eixo vertical principal da grade. O valor padrão desta propriedade é cinza (RGB(192, 192, 192)). Exemplo:

```
Sub RetanguloArr1_Click()  
    Screen.Item("E3Chart1").VerGridColor = RGB(255, 0, 0)  
End Sub
```

18.3.49 VerMinorTicks

 Determina o número de subdivisões entre as marcas do eixo vertical principal da grade. O valor padrão desta propriedade é 1. Exemplo:

```
Sub RetanguloArr1_Click()  
    Screen.Item("E3Chart1").VerMinorTicks = 3  
End Sub
```

18.3.50 VerScaleBegin

 Determina o valor no topo no eixo vertical principal da grade do E3Chart. O valor padrão desta propriedade é 100. Exemplo:

```
Sub RetanguloArr1_Click()  
    MsgBox Screen.Item("E3Chart1").VerScaleBegin  
End Sub
```

18.3.51 VerScaleEnd

 Determina o valor na base do eixo vertical principal da grade. O valor padrão desta propriedade é -100. Exemplo:

```
Sub RetanguloArr1_Click()  
    MsgBox Screen.Item("E3Chart1").VerScaleEnd  
End Sub
```

18.3.52 VerScaleFormat

Contém um texto que representa uma máscara dentro da qual os valores da escala vertical serão mostrados. Esta máscara pode representar vários tipos de valores:

- **Geral:** Não possui formatação específica, adaptando-se automaticamente ao valor especificado.
- **Número:** Apresenta números com parte inteira e fracionária. O usuário pode optar por até 15 casas decimais, por usar ou não um separador de milhares, e por apresentar números negativos com sinal ou entre parênteses. Para números muito grandes ou muito pequenos, recomenda-se utilizar o formato Científico.
- **Data:** Apresenta valores numéricos de data e hora (quando válidos). Para

representar apenas a hora, use o formato equivalente.

- **Hora:** Apresenta valores numéricos de hora e data (quando válidos). Para representar apenas a data, use o formato equivalente.
- **Porcentagem:** Multiplica o número por 100 e adiciona o símbolo de porcentagem. Admite até 15 casas decimais.
- **Científico:** Apresenta o número em notação de mantissa e expoente. Ideal para números de magnitude variada. Admite até 15 casas decimais.
- **Especial:** Permite formatar números inteiros em bases não-decimais (hexadecimal, octal e binária, por exemplo).
- **Outro:** Permite editar diretamente o código de formatação desejado, ou selecionar um formato criado anteriormente.

A máscara destes formatos, como exemplificadas no campo **Tipo**, será exibida na Janela de Propriedades (por exemplo, d/M/yy H:mm, 0E-00, etc.).

18.3.53 VerTickUnit

9 Determina o número de subdivisões entre as marcas da grade. Quando esta propriedade estiver com valor 0, o espaçamento será automático. Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    Old = E3Chart1.VerTickUnit  
    For i = 0 To 30 Step 10  
        E3Chart1.VerTickUnit = i  
        MsgBox "E3Chart1.VerTickUnit = " & CStr(i)  
    Next  
    MsgBox "Voltar"  
    E3Chart1.VerTickUnit = Old  
End Sub
```

18.3.54 Visible

9 Habilita ou desabilita a visibilidade do E3Chart. Se a propriedade estiver configurada para True, o E3Chart será visível. Caso contrário, o E3Chart não é mostrado. O valor padrão é True.

18.3.55 Width

9 Determina a largura do objeto, em unidades Himetric.

18.3.56 X

9 A propriedade **X** define a coordenada horizontal esquerda, em unidades Himetric, do objeto.

18.3.57 Y

Esta propriedade define a coordenada vertical superior, em unidades Himetric, do objeto.

18.4 Coleção de Penas

Esta seção contém informações sobre métodos e propriedades do objeto Coleção de Penas. Este objeto não possui eventos associados.

18.4.1 Métodos

Esta seção contém informações sobre os métodos do objeto Coleção de Penas.

NOTA: A coleção de penas do E3Chart deve ser acessada através da propriedade **Pens**.

18.4.1.1 AddPen

AddPen(*Pen*)

O método **AddPen** adiciona uma nova pena ao E3Chart, retornando a pena criada.
Exemplo:

```

Sub CommandButton1_Click()
    ' Cria uma pena sem nome.
    Set Pen = Screen.Item("E3Chart1").Pens.AddPen("")
    MsgBox Pen.Name
End Sub

Sub CommandButton1_DbClick()
    ' cria uma pena com o nome "Pen1".
    ' Se o nome existir, a pena é incrementada.

    Set Pen = Screen.Item("E3Chart1").Pens.AddPen("Pen1")
    MsgBox Pen.Name
End Sub

Sub CommandButton2_Click()
    ' Cria uma pena e associa a TagDemo1.
    Set Chart = Screen.Item("E3Chart1")
    Set Pen = Chart.Pens.AddPen("")
    MsgBox Pen.Name
    Pen.UsetimeStamp = True
    Pen.YLink = "Dados.TagDemo1"
    Pen.Connect()
End Sub

```

18.4.1.2 ChangePenPos

ChangePenPos(*Source, Dest*)

Modifica a ordem de desenho das penas no E3Chart. Este método possui os seguintes parâmetros:

- **Source:** determina o índice da pena a ser movida (começando em 1).
- **Dest:** determina o destino da pena (começando em 1).

Uma situação em que este método é especialmente útil é quando tem-se uma pena com o desenho de linha e outra com o desenho de área. Se a pena de área for desenhada depois da pena de linha, ela poderá esconder a última. A solução então seria inverter a ordem de desenho destas penas. Exemplo:

```

Sub CommandButton1_Click()
    ' Move a pena 1 para a posição 2.
    Screen.Item("E3Chart1").Pens.ChangePenPos(1, 2)
End Sub

```

18.4.1.3 Item

Item(*Index*)

O método **Item** retorna um objeto Pena da coleção de penas, especificado pelo índice. Este método possui o parâmetro *Index*, que pode ser do tipo numérico (se corresponder ao índice da pena) ou texto (se corresponder ao nome da pena).

Exemplo:

```

Sub CommandButton1_Click()
    ' Obtém a primeira pena.
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item(0)
End Sub

```

18.4.1.4 Remove

Remove(*Name*)

O método **Remove** apaga uma pena através do nome especificado. Este método possui o parâmetro *Name*, que determina o nome da pena a ser removida. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").Pens.Remove(1)  
End Sub
```

18.4.1.5 SetCursorPos

SetCursorPos(*X*, *Range*)

Posiciona o cursor de cada pena do E3Chart. Equivale ao comportamento de chamar o método **SetCursorPos** de cada Pena.

18.4.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Coleção de Penas.

18.4.2.1 Count

9 Contém o número total de penas inseridas no E3Chart. Esta propriedade é somente de leitura.

18.5 Penas

Esta seção contém informações sobre métodos e propriedades do objeto Pena. Este objeto não possui eventos associados.

18.5.1 Métodos

Esta seção contém informações sobre os métodos do objeto Pena.

18.5.1.1 AddPoint

AddPoint(*ValueX*, *ValueY*, [*Quality*])

Adiciona um ponto ao final do *buffer* de tempo real. O tamanho do *buffer* só é válido após a conexão da pena. Se a pena for criada no Studio, a conexão é automática, mas se for criada via script é necessário chamar o método **Connect** após sua criação. Este

método deve ser usado com penas de tempo real, e com a propriedade **UseTimeStamp** em **False**. O parâmetro opcional *Quality* indica a qualidade do ponto a ser inserido. Se este parâmetro não for informado, a qualidade do ponto será considerada boa (192). O número de pontos que podem ser adicionados a uma pena está relacionado ao tamanho do *buffer* (propriedade **BufferSize** da Pena).

18.5.1.2 Clear

Clear()

Apaga os dados do *buffer* de tempo real, sem diminuir seu tamanho. Este método não desconecta associações, nem tampouco remove dados históricos.

18.5.1.3 Connect

Connect()

O método **Connect** faz a pena conectar-se ao servidor para receber os dados de tempo real, associando as propriedades **XLink** e **YLink**. Se a pena já estiver conectada, o método fica inativo. Exemplo:

```
Sub CommandButton1_Click()  
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")  
    Pen1.Disconnect()  
    Pen1.Connect()  
End Sub
```

18.5.1.4 Disconnect

Disconnect()

O método **Disconnect** limpa os dados atuais e faz com que a pena não receba mais os dados de tempo real do seu tag associado. Se a pena já estiver desconectada o método fica inativo. Quando o método **Disconnect** é usado em uma pena mista (*DataSourceType* = 2), ele remove a parte de tempo real, permanecendo a parte histórica. Em tempo de execução, para que os dados de tempo real sejam mostrados novamente é preciso chamar o método **Connect**. Exemplo:

```
Sub CommandButton1_Click()  
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")  
    Pen1.Disconnect()  
    Pen1.Connect()  
End Sub
```

18.5.1.5 GetCursorPos

GetCursorPos(*X*, *Y*)

Retorna a posição em que o cursor interceptou a pena no E3Chart. Este método possui os parâmetros *X* e *Y*, que correspondem às coordenadas *x* e *y* do cursor. Se o método for bem sucedido retorna True, caso contrário, retorna False. Exemplo:

```
Sub CommandButton1_Click()
    For Each pen In Chart.Pens
        If pen.GetCursorPos(aa, bb) Then
            strResult = strResult & pen.name & " := " & CStr(CDate(aa)) + _
                "y " + CStr(bb) + vbNewLine
        End If
    Next
    MsgBox strResult
End Sub
```

18.5.1.6 GetPoint

GetPoint(*ValueX*, *ValueY*)

Retorna as coordenadas *X* e *Y* do ponto mais próximo ao valor de entrada em *ValueX*. O parâmetro *ValueX* informa o valor de referência para a busca do ponto, e depois recebe o valor efetivo da coordenada *X* do ponto mais próximo encontrado. O parâmetro *ValueY* retorna o valor efetivo da coordenada *Y* do ponto mais próximo encontrado. O método retorna True se encontrou um ponto, e False caso contrário.

18.5.1.7 SetCursorPos

SetCursorPos(*X*, *Range*)

Posiciona o cursor da pena do E3Chart. O parâmetro *X* indica em qual posição o cursor deve ser posicionado, similar ao comportamento de mover o cursor na pesquisa com o mouse. O cursor é movido para a posição mais próxima indicada por *X*. O parâmetro *Range* é opcional e usado como validação. O cursor só será movido se o ponto válido estiver dentro do intervalo. Qualquer valor negativo indica que o intervalo passado não deve ser usado. Exemplo:

```
' Existindo um ponto válido em (x = 1, y = 10) e outro em (x = 4, y = 20).
SetCursorPos(2) ' move o cursor para o ponto (1, 10)
SetCursorPos(4) ' move o cursor para o ponto (4, 20)

' Quando range é usado, o cursor só é movido se ponto estiver dentro do range.
' Não vai mover o cursor, porque 2 está mais de 0.5 unidade
' distante do ponto mais próximo, que é 1.
SetCursorPos (2, 0.5)
' Envia o cursor para o ponto (1, 10).
SetCursorPos (2, 4)
```

Este método retorna True se o cursor foi movido, caso contrário retorna False.

18.5.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Pena.

18.5.2.1 AutoQuery

 Determina consulta automática da pena. Se True, a pena requer o uso da consulta automática. Do contrário, a consulta automática não será utilizada. A consulta automática não poderá ser utilizada nos seguintes casos:

- Quando o SQL é customizado pelo usuário.
- Quando se está utilizando Storage.
- Quando se está utilizando várias tabelas.

18.5.2.2 AverageY

 Informa a média da pena no intervalo, caso a propriedade **EnableCalc** esteja habilitada. Caso o E3Chart esteja no modo de **Pesquisa de Intervalos**, mostra a média neste intervalo; do contrário, mostra a média no intervalo entre o início e o fim no eixo horizontal. Valores com qualidade ruim não são considerados caso a propriedade **ShowBadPoints** esteja desabilitada. Esta propriedade é somente de leitura.

18.5.2.3 BkColor

 Determina a cor de fundo usada em uma pena do tipo área. O valor padrão desta propriedade é vazio. Exemplo:

```
Sub CommandButton1_Click()  
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")  
    Pen1.BkColor = RGB(255, 0, 0)  
End Sub
```

18.5.2.4 BufferSize

 Determina o número de pontos mantidos na pena de tempo real. Após este valor, os dados mais antigos são descartados. Nas penas históricas, esta propriedade não tem efeito. Esta propriedade será considerada somente após a conexão da pena. Para maiores informações, veja o método **Connect**. O valor padrão desta propriedade é 1000, e deve ser sempre maior que 0. Exemplo:

```

Sub CommandButton1_Click()
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")
    Pen1.Disconnect()
    Pen1.BufferSize = 5000
    Pen1.Connect()
End Sub

```

18.5.2.5 Color

 Determina a cor da linha da pena no E3Chart. O valor padrão desta propriedade é vazio. Exemplo:

```

Sub CommandButton1_Click()
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")
    Pen1.Color= RGB(212, 208, 20)
End Sub

```

18.5.2.6 DataSourceType

 Determina a origem dos dados da pena. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para DataSourceType

OPÇÃO	DESCRIÇÃO
0 - Tempo Real	Indica conexão da pena a um tag atualizado em tempo real.
1 - Histórico	Indica conexão da pena aos dados vindos de uma consulta.
2 - Pena Mista	Indica conexão da pena a tags de tempo real e dados históricos simultaneamente.

Quando a propriedade **DataSourceType** for igual a 0 (Tempo Real), as propriedades **XLink** e **YLink** informam os links utilizados; ou então a propriedade **UseTimeStamp** informa que a propriedade **XLink** não será usada, estando em seu lugar o *timestamp* da propriedade **YLink**. Quando a propriedade **DataSourceType** for igual a 1 (Histórico), as propriedades **XField** e **YField** informam os campos da tabela a serem usados. A propriedade **QueryName** indica o nome da tabela usada. Quando **DataSourceType** for igual a 2 (Pena Mista), as opções 0 e 1 funcionarão simultaneamente para a pena.

NOTA: Em tempo de execução, quando a propriedade for alterada e a pena deixar de mostrar os dados de tempo real, será preciso chamar o método **Connect** para que estes dados sejam mostrados novamente.

Exemplo:

```
Sub CommandButton1_Click()
    Set E3Chart1 = Screen.Item("E3Chart1")
    MsgBox "Clique OK para criar a pena."
    Set Pen = E3Chart1.Pens.AddPen("Pen1")
    Pen.DataSourceType = 0 ' Tempo real
    Pen.YLink = "Dados.TagDemo1"
    Pen.UseTimeStamp = True ' Em X usa o timestamp
    Pen.Color = RGB(255, 0, 0)
    Pen.Docstring = "Teste"
    MsgBox "Clique OK para conectar."
    Pen.Connect() ' Começa a receber os dados
    MsgBox "Clique OK para enquadrar."
    E3Chart1.FitPen(0)
    MsgBox "Clique OK para remover a pena."
    E3Chart1.Pens.Remove(Pen.Name)
End Sub
```

18.5.2.7 DigitalData

Determina o estilo de plotagem digital. Se a propriedade estiver configurada para True, o estilo de plotagem digital assume que a variação do dado seja de forma digital, ou seja, que seu valor em relação ao último variou instantaneamente. Caso contrário, a variação é considerada linear e os pontos são unidos por um segmento de reta. O valor padrão desta propriedade é True. Exemplo:

```
Sub CommandButton1_Click()
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")
    Pen1.DigitalData = True
End Sub
```

18.5.2.8 EnableCalc

Habilita ou desabilita o cálculo da média, mínimos e máximos dentro do intervalo.

18.5.2.9 EnableHighLimit

Habilita ou desabilita a verificação do limite alto.

18.5.2.10 EnableLowLimit

Habilita ou desabilita a verificação do limite baixo.

18.5.2.11 EU

A Esta propriedade é utilizada para identificar a unidade de engenharia que o valor representa, como por exemplo graus, metros, KW/h, etc.

18.5.2.12 HighLimit

9 Determina o limite de alarme alto.

18.5.2.13 InterpolatedBeginY

9 Informa o valor do ponto interpolado onde o cursor inicial cruza com a pena. Esta propriedade é somente de leitura.

18.5.2.14 InterpolatedEndY

9 Informa o valor do ponto interpolado onde o cursor final cruza com a pena. Esta propriedade é somente de leitura.

18.5.2.15 LimitPenBkColor

 Determina a cor de fundo da pena quando em alarme.

18.5.2.16 LimitPenColor

 Determina a cor da pena quando em alarme.

18.5.2.17 LowLimit

9 Determina o limite de alarme baixo.

18.5.2.18 MaxY

9 Informa o valor máximo da pena no intervalo, caso a propriedade **EnableCalc** esteja habilitada. Caso o E3Chart esteja no modo de **Pesquisa de Intervalos**, mostra a média nesse intervalo; do contrário, mostra a média no intervalo entre o início e o fim no eixo horizontal. Valores com qualidade ruim não são considerados caso a propriedade **ShowBadPoints** esteja desabilitada. Esta propriedade é somente de leitura.

18.5.2.19 MinY

9 Informa o valor mínimo da pena no intervalo, caso a propriedade **EnableCalc** esteja habilitada. Caso o E3Chart esteja no modo de **Pesquisa de Intervalos**, mostra a média nesse intervalo; do contrário, mostra a média no intervalo entre o início e o fim no eixo horizontal. Valores com qualidade ruim não são considerados caso a propriedade **ShowBadPoints** esteja desabilitada. Esta propriedade é somente de leitura.

18.5.2.20 Name

A Determina o nome da pena. Exemplo:

```
Sub CommandButton1_Click
    Screen.Item("E3Chart1").Pens.Name = "Pen1"
End Sub
```

18.5.2.21 PenStyle

 Determina o tipo de linha da pena. As opções disponíveis são as seguintes:

Opções disponíveis para PenLineStyle

OPÇÃO	DESCRIÇÃO
0 - LsSolid	Linha sólida.
1 - LsDash	Linha tracejada.
2 - LsDot	Linha pontilhada.
3 - LsDashDot	Linha traço-ponto.
4 - LsDashDotDot	Linha traço-ponto-ponto.
5 - LsNull	Sem linha.

Exemplo:

```
Sub CommandButton1_Click()
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")
    Pen1.PenLineStyle = 4
End Sub
```

18.5.2.22 PenType

 Determina o tipo de desenho da pena no E3Chart:

- 0: Linha
- 1: Ponto
- 2: Ponto-linha
- 3: Área

Exemplo:

```
Sub CommandButton1_Click()  
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")  
    Pen1.PenType = 1  
End Sub
```

18.5.2.23 QueryName

A Determina o nome da consulta que a pena está utilizando. Esta propriedade é utilizada se a propriedade **DataSourceType** estiver configurada para 1 (Histórico).

Exemplo:

```
Sub CommandButton1_Click()  
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")  
    Pen1.QueryName = Consulta12  
End Sub
```

18.5.2.24 ScaleX e ScaleY

9 As propriedades **ScaleX** e **ScaleY** indicam a quais escalas X e Y do E3Chart a pena está associada, respectivamente. A escala configurada para **ScaleX** tem orientação horizontal, ou seja, pode estar posicionada no topo ou na base do E3Chart. A escala configurada para **ScaleY** tem orientação vertical, estando posicionada à esquerda ou à direita do objeto. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    Set Pen = Chart.Pens.AddPen("PenaDoTagDemo2")  
    Pen.YLink = "Dados.TagDemo2"  
    Pen.UseTimeStamp = True  
    ' A escala deve existir previamente.  
    Pen.ScaleY = "EscalaParaTagDemo2"  
    Pen.Connect  
End Sub
```

18.5.2.25 ScanValue

 Define o tempo de leitura esperado do tag da pena de tempo real. Este valor é levado em conta no modo de desenho analógico. Quando o valor ultrapassa o valor determinado por **ScanValue**, considera-se que o valor do tag não foi alterado no intervalo. De outra forma, quando o **ScanValue** é igual a zero, os dados da pena são sempre conectados com uma reta ligando os dois pontos como se o valor estivesse variando de forma linear. A unidade desta propriedade é milissegundos. Exemplo:

```
Sub CommandButton1_Click()  
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")  
    Pen1.ScanValue = 1000  
End Sub
```

18.5.2.26 ShowAverage

Habilita a exibição da média da pena no E3Chart. Esta propriedade só tem efeito caso a propriedade **EnableCalc** esteja habilitada.

18.5.2.27 ShowBadPoints

Quando desabilitada, os pontos com qualidade ruim não serão plotados. Se a pena estiver traçando linhas, as linhas que passam pelos pontos de qualidade ruim não serão conectadas. Para que a qualidade dos pontos seja levada em consideração na parte histórica da pena, o campo **nomecampo_quality** deve estar selecionado na Consulta do E3Chart. Quando habilitada, todos os pontos são plotados normalmente.

18.5.2.28 ShowMinMax

Habilita a exibição dos pontos mínimo e máximo da pena no E3Chart. Esta propriedade só tem efeito caso a propriedade **EnableCalc** esteja habilitada.

18.5.2.29 UseTimeStamp

 Determina que será utilizado, para o eixo horizontal, o valor do *timestamp* associado ao eixo vertical. Veja exemplo na descrição da propriedade **DataSourceType**.

18.5.2.30 Visible

☑ Determina se pena está visível no E3Chart. Se a opção estiver configurada para True, a pena é visível em tempo de execução. Caso contrário, a pena fica invisível. Exemplo:

```
Sub CommandButton1_Click()  
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")  
    Pen1.Visible = Not Pen1.Visible  
End Sub
```

18.5.2.31 Width

9 Determina a largura da linha da pena no E3Chart. Exemplo:

```
Sub CommandButton1_Click()  
    Set Pen1 = Screen.Item("E3Chart1").Pens.Item("Pen1")  
    Pen1.Width = 10  
End Sub
```

18.5.2.32 XField

A Nome do campo da consulta usado para plotar o dado em uma escala horizontal. É usada para penas históricas.

18.5.2.33 XLink

A Nome do *link* usado para plotar o dado em uma escala horizontal. Quando o valor desta propriedade é alterado, a pena é automaticamente desconectada. Após a configuração, é preciso chamar o método **Connect** para que a pena comece a receber os dados relativos a este *link*. É usada para penas de tempo real.

18.5.2.34 XMaxY

9 Informa o valor X relativo ao ponto **MaxY**. Esta propriedade é somente de leitura.

18.5.2.35 XMinY

9 Informa o valor X relativo ao ponto **MinY**. Esta propriedade é somente de leitura.

18.5.2.36 YField

A Nome do campo da consulta usado para plotar o dado em uma escala vertical. É usada para penas históricas.

18.5.2.37 YLink

A Nome do *link* usado para plotar o dado em uma escala vertical. Quando o valor desta propriedade é alterado, a pena é automaticamente desconectada. Após a configuração, é preciso chamar o método **Connect** para que a pena comece a receber os dados relativos a este *link*. É usada para penas de tempo real.

18.6 Coleção de Eixos

Esta seção contém informações sobre métodos e propriedades do objeto Coleção de Eixos. Este objeto não possui eventos associados.

18.6.1 Métodos

Esta seção contém informações sobre os métodos do objeto Coleção de Eixos.

18.6.1.1 AddAxis

AddAxis(*AxisName*)

Adiciona um novo eixo com o nome determinado no parâmetro *AxisName* e retorna o eixo criado. Caso se tente criar um eixo com um nome que já existe, será mostrada uma mensagem de erro. Querendo-se que o nome seja gerado automaticamente, deve se passar o parâmetro *AxisName* em branco. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    Set newAxis = Chart.Axes.AddAxis("")  
    newAxis.Color = RGB(255, 0, 0)  
End Sub
```

18.6.1.2 Remove

Remove(*Index*)

Remove o eixo através de seu nome ou índice, conforme especificado no parâmetro *Index*. Os eixos principais 0 e 1 não podem ser removidos. Caso se tente removê-los, aparecerá uma mensagem de erro. Exemplos:

```
Sub CommandButton1_Click()  
    ' Este exemplo remove todos os eixos adicionais
```

```

Set Chart = Screen.Item("E3Chart")
While (Chart.Axes.Count > 2)
    Chart.Axes.Remove(2)
Wend
End Sub

Sub CommandButton1_Click()
    ' Remove um eixo adicional, se existir
    Set Chart = Screen.Item("E3Chart1")
    Chart.Axes.Remove(2)
End Sub

```

18.6.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Coleção de Eixos.

18.6.2.1 Count

9 Retorna o número total de eixos do E3Chart, incluindo os dois eixos principais (horizontal e vertical). Exemplo:

```

Sub CommandButton1_Click()
    Set Chart = Screen.Item("E3Chart1")
    MsgBox Chart.axes.Count
End Sub

```

18.6.2.2 HorAxis

9 Retorna o eixo horizontal principal. Este eixo também participa da lista de eixos. Exemplo:

```

Sub CommandButton1_Click()
    Set Chart = Screen.Item("E3Chart1")
    MsgBox Chart.axes.Item(0).Name & ", " & Chart.axes.Item(1).Name
    MsgBox Chart.axes.HorAxis.Name & ", " & _
        Chart.axes.Item("NomeDoEixo").Name
End Sub

```

18.6.2.3 Item

9 Retorna o eixo através de seu nome ou índice. O índice 0 é sempre o eixo horizontal principal e o índice 1 é sempre o eixo vertical principal. Exemplo:

```

Sub CommandButton1_Click()
    Set Chart = Screen.Item("E3Chart")
    MsgBox Chart.axes.Item(0).Name & ", " & Chart.axes.Item(1).Name
    MsgBox Chart.axes.HorAxis.Name & ", " & Chart.axes.Item("NomeDoEixo").Name
End Sub

```

18.6.2.4 VerAxis

9 Retorna o eixo vertical principal. Este eixo também participa da lista de eixos. Exemplos:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    MsgBox Chart.axes.Item(0).Name & ", " & Chart.axes.Item(1).Name  
    MsgBox Chart.axes.Item("NomeDoEixo").Name & ", " & _  
        Chart.axes.VerAxis.Name  
End Sub
```

18.7 Eixos

Esta seção contém informações sobre métodos e propriedades do objeto Eixo. Este objeto não possui eventos associados.

18.7.1 Métodos

Esta seção contém informações sobre os métodos do objeto Eixo.

NOTA: **HorAxis** e **VerAxis** são propriedades da coleção de eixos que acessam os eixos padrão horizontal e vertical, respectivamente. Por exemplo, ao invés de utilizar `Chart.Axes.Item("EixoHorizontal")`, pode-se utilizar `Chart.Axes.HorAxis`. Demais eixos criados pelo usuário terão nomes próprios, escolhidos caso a caso.

18.7.1.1 GetHistoricPeriod

GetHistoricPeriod(*Begin*, *End*)

Retorna o intervalo de tempo mostrada na escala histórica. O parâmetro *Begin* indica a data inicial da escala histórica, e *End* indica a data final. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart")  
    Chart.Axes.Item("NomeDoEixo").GetHistoricPeriod min, max  
    value = CStr(dmin) & " " & CStr(dmax)  
    MsgBox "data inicial = " & CStr(min) & _  
        vbNewLine & " data final = " & CStr(max)  
End Sub
```

18.7.1.2 GetMinMax

GetMinMax(*Min*, *Max*)

Retorna nos parâmetros *Min* e *Max* os valores mínimo e máximo da escala numérica, respectivamente. Exemplo:

```
Sub CommandButton1_DBClick()  
    Set Chart = Screen.Item("E3Chart")  
    Chart.Axes.Item("NomeDoEixo").GetMinMax dmin, dmax  
    MsgBox CStr(dmin) & " " & CStr(dmax)  
End Sub
```

18.7.1.3 GetRealTimePeriod

GetRealTimePeriod(*Period*)

Retorna a unidade de tempo configurada na escala de tempo real. O parâmetro *Period* recebe o valor da escala de tempo. Exemplo:

```
Dim Unidade, Valor  
Unidade = Screen.Item("E3Chart1").Axes.Item_  
("EixoHorizontal").GetRealTimePeriod(Valor)  
MsgBox "Valor: " & CStr(Valor) & " Unidade: " & CStr(Unidade)
```

18.7.1.4 GetTickSpacing

GetTickSpacing(*TickSpacing*, *TimeUnit*)

Retorna o espaçamento entre os ticks (subdivisões da escala) e a unidade configurada. O parâmetro *TickSpacing* determina o espaçamento entre os ticks, e *TimeUnit* determina a unidade. Quando este parâmetro está em zero, significa que é automático. A unidade não é utilizada quando a escala é numérica.

As opções disponíveis no parâmetro *TimeUnit* são as mesmas demonstradas na tabela do método **GetRealTimePeriod**. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    Unitvalue_  
    = Chart.Axes.Item("NomeDoEixo").GetTickSpacing(TickSpacing)  
    MsgBox "valor" = " & CStr(TickSpacing) & _  
    " unidade " & CStr(unitvalue)  
End Sub
```

18.7.1.5 SetHistoricPeriod

SetHistoricPeriod(*Begin*, *End*)

Configura o período de tempo para a escala histórica. O parâmetro *Begin* determina o período inicial da escala, e *End* determina o período final da escala. Exemplo:

```

Sub CommandButton1_Click()
    Set Chart = Screen.Item("E3Chart2")
    Chart.Axes.Item("NomeDoEixo").ScaleType = 2
    ' Mostra o último período
    Chart.Axes.Item("NomeDoEixo").SetHistoricPeriod now - 1, now
End Sub

```

18.7.1.6 SetMinMax

SetMinMax(*Min*, *Max*)

Configura os valores mínimo e máximo da escala numérica. O valor mínimo é determinado pelo parâmetro *Min* e o máximo por *Max*. Exemplo:

```

Sub Circulo1_Click()
    Set Chart = Screen.Item("E3Chart2")
    Chart.Axes.Item("NomeDoEixo").SetMinMax -10, 500
End Sub

```

18.7.1.7 SetRealTimePeriod

SetRealTimePeriod(*Times*, *TimeUnit*)

Ajusta o intervalo de tempo na unidade definida pelo parâmetro *TimeUnit*. As opções disponíveis neste parâmetro são as mesmas demonstradas na tabela do método **GetRealTimePeriod**. O parâmetro *Times* determina o intervalo de tempo, e a unidade da escala é especificado por *TimeUnit*. O eixo fica sempre sendo atualizado neste modo (tempo real). Exemplo:

```

Sub CommandButton1_Click()
    Set Chart = Screen.Item("E3Chart5")
    ' tuSeconds = 0, tuMinutes = 1, tuHours = 2, tuDays = 3,
    ' tuWeeks = 4, tuMonths = 5, tuYears = 6
    ' 2 minutos
    Chart.Axes.Item("NomeDoEixo").SetRealTimePeriod 2, 1
    Chart.Axes.Item("NomeDoEixo").SetTickSpacing 30, 0
End Sub

```

18.7.1.8 SetTickSpacing

SetTickSpacing(*TickSpacing*, *TimeUnit*)

Configura o espaçamento entre os ticks (subdivisões da escala) usando a unidade. O espaçamento entre os ticks é determinado pelo parâmetro *TickSpacing*. O parâmetro *TimeUnit* determina a unidade. Caso a escala seja numérica, a unidade não é considerada. As opções disponíveis no parâmetro *TimeUnit* são as mesmas demonstradas na tabela do método **GetRealTimePeriod**. Exemplo:

```

Sub CommandButton1_Click()
    Set Chart = Screen.Item("E3Chart1")
    ' 10 (caso seja a escala numérica,
    ' o valor da unidade não é considerado)
    Chart.Axes.Item("NomeDoEixo").SetTickSpacing 10, 0
    Chart.Axes.Item("NomeDoEixo").SetTickSpacing 20, 0
End Sub

```

18.7.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Eixo.

NOTA: **HorAxis** e **VerAxis** são propriedades da coleção de eixos que acessam os eixos padrão horizontal e vertical, respectivamente. Por exemplo, ao invés de utilizar `Chart.Axes.Item("EixoHorizontal")`, pode-se utilizar `Chart.Axes.HorAxis`. Demais eixos criados pelo usuário terão nomes próprios, escolhidos caso a caso.

18.7.2.1 Color

 Determina a cor principal do Eixo. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3chart1")  
    Chart.Axes.HorAxis.Color = RGB(0, 255, 0)  
End Sub
```

18.7.2.2 Format

 Determina a formatação dos valores do eixo. Exemplo:

```
Sub CommandButton1_Click()  
    ' Muda a formatação  
    Set Chart = Screen.Item("E3chart1")  
    strOldFormat = Chart.Axes.Item("NomeDoEixo").Format  
    MsgBox "Clique para setar a formatação automática."  
    Chart.Axes.Item("NomeDoEixo").Format = "" ' Automatico  
    MsgBox "Clique para usar outra formatação."  
    Chart.Axes.Item("NomeDoEixo").Format = "0.0"  
    MsgBox "Clique para usar outra formatação."  
    Chart.Axes.Item("NomeDoEixo").Format = "dd/MM/yy hh:mm:ss"  
    MsgBox "Clique novamente para voltar à formatação original."  
    Chart.Axes.Item("NomeDoEixo").Format = strOldFormat  
End Sub
```

18.7.2.3 GridColor

 Determina a cor das linhas da grade. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3chart1")  
    Chart.Axes.Item("NomeDoEixo").ShowGrid = False  
    MsgBox "Clique para trocar a cor das linhas na grade."  
    Chart.Axes.Item("NomeDoEixo").GridColor = RGB(0, 0, 255)  
    Chart.Axes.Item("NomeDoEixo").ShowGrid = True  
End Sub
```

18.7.2.4 GridStyle

 Determina o estilo da linha da grade. As opções disponíveis são as seguintes:

Opções disponíveis para GridStyle

OPÇÃO	DESCRIÇÃO
0 - solid	O estilo da linha da grade é sólida.
1 - dash	O estilo da linha da grade é pontilhada.
2 - dot	O estilo da linha da grade é tracejada.
3 - dashdot	O estilo da linha da grade é pontilhada e tracejada.
4 - dashdotdot	O estilo da linha da grade é traço-ponto-ponto.
5 - invisible	Não há linhas visíveis na grade (invisível).

Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    ' Solid 0, dash 1, dot 2, dashdot 3, dashdotdot 4, invisible 5  
    For i = 0 To 5  
        MsgBox "Clique para mudar o estilo da linha no grid."  
        Chart.Axes.Item("NomeDoEixo").GridStyle = i  
    Next  
End Sub
```

18.7.2.5 Inverse

Inverte a ordem dos valores mínimo e máximo na escala numérica. Normalmente, em escalas verticais, o valor mínimo parece abaixo, e o valor máximo acima. Nas escalas horizontais, o valor mínimo aparece na esquerda, e o valor máximo na direita. Quando a propriedade **Inverse** estiver configurada para True, no entanto, esta ordem será invertida, valores máximos abaixo ou à esquerda, valores mínimos acima ou à direita. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    Chart.Axes.Item("NomeDoEixo").Inverse = Not _  
        Chart.Axes.Item("NomeDoEixo").Inverse  
End Sub
```

18.7.2.6 MinorTicks

9 Determina o número total de subdivisões entre as escalas. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    Chart.Axes.Item("NomeDoEixo").MinorTicks = _  
    Chart.Axes.Item("NomeDoEixo").MinorTicks + 1  
End Sub
```

18.7.2.7 Mirror

Indica o espelhamento do eixo. Se a propriedade estiver configurada para True, o eixo será espelhado no lado oposto do eixo original. Caso contrário, o eixo permanece na mesma posição. Exemplo:

```
Sub CommandButton1_DBClick()  
    Set Chart = Screen.Item("E3Chart1")  
    Chart.Axes.Item("NomeDoEixo").Mirror = Not _  
    Chart.Axes.Item("NomeDoEixo").Mirror  
End Sub
```

18.7.2.8 Name

A Determina o nome do eixo. Exemplo:

```
Sub CommandButton1_DBClick()  
    Set Chart = Screen.Item("E3Chart1")  
    Chart.Axes.Item("NomeDoEixo").Title = Chart.Axes.Item("NomeDoEixo").Name  
End Sub
```

18.7.2.9 Position

 Determina a posição do eixo em relação à grade do E3Chart. As opções disponíveis são as seguintes:

Opções disponíveis para Position

OPÇÃO	DESCRIÇÃO
0 - axpLeft	O eixo é posicionado à esquerda da escala.
1 - axpRight	O eixo é posicionado à direita da escala.
2 - axpTop	O eixo é posicionado no topo da escala.
3 - axpBottom	O eixo é posicionado na base da escala.

Exemplo:

```
Sub CommandButton1_Click()  
Set Chart = Screen.Item("E3Chart1")  
Set newAxis = Chart.Axes.AddAxis("")  
For i = 0 To 3  
    MsgBox "Clique OK para mudar a posição do eixo."  
    newAxis.Position = i  
Next  
MsgBox "Remover o eixo."  
Chart.Axes.Remove(newAxis.Name)  
End Sub
```

18.7.2.10 ScaleType

 Determina o tipo de escala mostrada pelo eixo. As opções disponíveis deste item são as seguintes:

Opções disponíveis para ScaleType

OPÇÃO	DESCRIÇÃO
0 - atNumberScale	Escala numérica.
1 - atLastPeriod	Mostra último período (Tempo Real).
2 - atPeriod	Intervalo de tempo (Histórica).

Exemplo:

```
Sub CommandButton1_Click()  
Set Chart = Screen.Item("E3Chart1")  
Set newAxis = Chart.Axes.AddAxis("")  
For i = 0 To 2  
    MsgBox "Clique OK para mudar o tipo de escala."  
    newAxis.ScaleType = i  
Next  
MsgBox "Remover o eixo."  
Chart.Axes.Remove(newAxis.Name)  
End Sub
```

18.7.2.11 ShowGrid

Determina a visualização das linhas da grade. Se a propriedade estiver configurada para True, as linhas da grade são mostradas. Caso contrário, as linhas são escondidas. Exemplo:

```
Sub CommandButton_Click()  
Set Chart = Screen.Item("E3Chart1")  
Chart.Axes.Item("NomeDoEixo").ShowGrid = Not _  
Chart.Axes.Item("NomeDoEixo").ShowGrid
```

```
End Sub
```

18.7.2.12 Title

A Determina o título do eixo. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item("E3Chart1")  
    Chart.Axes.Item("NomeDoEixo").Title = _  
        Chart.Axes.Item("NomeDoEixo").Name  
    MsgBox "Clique para remover o título."  
    ' Retira o título  
    Chart.Axes.Item("NomeDoEixo").Title = ""  
End Sub
```

18.7.2.13 Visible

Determina a visibilidade do eixo na grade. Se a propriedade estiver configurada para True, o eixo é visível na grade. Caso contrário, o eixo fica invisível. Exemplo:

```
Sub CommandButton1_Click()  
    Set Chart = Screen.Item(E3Chart1)  
    Chart.Axes.Item("NomeDoEixo").Visible = Not _  
        Chart.Axes.Item("NomeDoEixo").Visible  
End Sub
```

18.8 Coleção de Consultas

Esta seção contém informações sobre métodos e propriedades do objeto Coleção de Consultas. Este objeto não possui eventos associados.

18.8.1 Métodos

Esta seção contém informações sobre os métodos do objeto Coleção de Consultas.

18.8.1.1 AddQuery

AddQuery(*QueryName*, *IsInternal*)

Adiciona uma consulta na coleção de consultas do E3Chart. Este método possui os parâmetros *QueryName*, que determina o nome da consulta que está sendo adicionada, e *IsInternal*, que é opcional, obsoleto e não deve ser informado. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").Queries.AddQuery("Consulta1")  
End Sub
```

18.8.1.2 Item

Item(*Index*)

O método **Item** retorna um objeto Consulta da coleção de consultas, especificado pelo índice. Este método possui o parâmetro *Index*, que pode ser do tipo numérico (se corresponder ao índice da consulta) ou texto (se corresponder ao nome da consulta). Exemplo:

```
Sub CommandButton1_Click()  
    Set query = Screen.Item("E3Chart1").Queries.Item(0)  
End Sub
```

18.8.1.3 Remove

Remove(*Index*)

Remove o objeto Consulta especificado pelo nome ou pelo índice da Coleção de Consultas. Este método possui o parâmetro *Index*, que pode ser do tipo numérico (se corresponder ao índice da consulta) ou texto (se corresponder ao nome da consulta). Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").Queries.Remove(0)  
End Sub
```

18.8.1.4 UpdateData

UpdateData()

Atualiza os dados de todas as consultas. Exemplo:

```
Sub Texto1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    E3Chart1.Queries.UpdateData()  
End Sub
```

18.8.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Coleção de Consultas.

18.8.2.1 Count

9 Contém o número total de consultas do E3Chart. Esta propriedade é somente de leitura.

18.9 Legenda

Esta seção contém informações sobre métodos e propriedades do objeto `Legenda`. Este objeto não possui eventos associados.

18.9.1 Métodos

Na legenda, podem ser selecionadas várias colunas. Cada coluna mostra um tipo de informação e possui um nome e um valor correspondente. A seguir, tem-se a tabela da descrição das colunas possíveis na legenda.

Opções disponíveis para identificação das colunas

OPÇÃO	DESCRIÇÃO
Cor da pena	Mostra a cor da pena.
Descrição	Mostra a propriedade DocString da pena.
DifX	Mostra a diferença entre os cursores inicial e final.
DifY	Mostra a diferença entre os pontos interpolados inicial e final no eixo Y.
EscalaX	Mostra o nome da escala associada ao tagX.
EscalaY	Mostra o nome da escala associada ao tagY.
Estado	Mostra o status da pena.
FimX	Mostra a posição final do cursor.
FimY	Mostra o ponto interpolado onde o cursor final cruza a pena.
InicioX	Mostra a posição inicial do cursor.
InicioY	Mostra o ponto interpolado onde o cursor inicial cruza a pena.
MaximoY	Mostra o valor máximo da pena no intervalo.
MediaY	Mostra o valor médio da pena no intervalo.

OPÇÃO	DESCRIÇÃO
MínimoY	Mostra o valor mínimo da pena no intervalo.
Nome da pena	Mostra o nome da pena.
Nome do tag X	Mostra o nome do tagX associado à pena.
Nome do tag Y	Mostra o nome do tagY associado à pena.
Unidade	Mostra a unidade associada à pena.
Valor do tag X	Mostra o valor do tagX associado à pena. Este valor é mostrado quando o gráfico está no modo de Pesquisa.
Valor do tag Y	Mostra o valor do tagY associado à pena. Este valor é mostrado quando o gráfico está no modo de Pesquisa.

18.9.1.1 ChangeColumnPos

ChangeColumnPos(*Source*, *Dest*)

Troca a posição entre duas colunas. Possui os seguintes parâmetros:

- **Source**: índice da coluna que será movida para *Dest*.
- **Dest**: índice da coluna que será movida para *Source*.

Exemplo:

```
Sub CommandButton1_Click()
    Screen.Item("E3Chart1").Legend.ChangeColumnPos 1, 2
End Sub
```

18.9.1.2 Count

Count()

Retorna o número de colunas da legenda. Exemplo:

```
Sub CommandButton1_Click()
    MsgBox Screen.Item("E3Chart1").Legend.Count()
End Sub
```

18.9.1.3 InsertColumn

InsertColumn(*Col*, *Index*)

Insera uma nova coluna na legenda. Este método possui os seguintes parâmetros:

- **Col**: identifica a coluna a ser inserida (ver **Tabela de Identificação das Colunas**, no início do tópico **Métodos da Legenda**).
- **Index**: determina a posição em que a coluna será inserida.

Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").Legend.InsertColumn 1, 2  
End Sub  
  
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").Legend.InsertColumn "Color", 2  
End Sub
```

18.9.1.4 Item

Item(*Col*)

Retorna uma coluna da legenda pelo nome ou pelo índice. O parâmetro *Col* determina o índice ou o nome da coluna (ver **Tabela de Identificação das Colunas**, no início do tópico **Métodos da Legenda**). Exemplo:

```
Sub CommandButton1_Click()  
    MsgBox Screen.Item("E3Chart1").Legend.Item(1)  
End Sub
```

18.9.1.5 RemoveColumn

RemoveColumn(*Col*)

Remove uma coluna. Este método possui o parâmetro *Col*, que determina a coluna que será removida (ver **Tabela de Identificação das Colunas**, no início do tópico **Métodos da Legenda**). Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Chart1").Legend.RemoveColumn(1)  
End Sub
```

18.9.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto `Legenda`.

18.9.2.1 Count

 Retorna o número total de colunas da `Legenda`. Exemplo:

```
Sub CommandButton10_Click()  
    MsgBox Screen.Item("E3Chart1").Legend.Count  
End Sub
```

18.9.2.2 LegendPos

 Indica a posição da legenda no `E3Chart`.

Opções disponíveis para `LegendPos`

OPÇÃO	DESCRIÇÃO
0 - <code>IgTop</code>	Mostra a legenda no topo.
1 - <code>IgLeft</code>	Mostra a legenda à esquerda.
2 - <code>IgBottom</code>	Mostra a legenda na base.
3 - <code>IgRight</code>	Mostra a legenda à direita.

Exemplo:

```
Sub CommandButton10_Click()  
    Screen.Item("E3Chart1").Legend.LegendPos = 3  
End Sub
```

18.9.2.3 ShowAllPens

Quando a propriedade for `True`, todas as penas do `E3Chart` são mostradas na legenda. A propriedade `Visible` da pena é ignorada. Quando em `False`, somente as penas com a propriedade `Visible` igual a `True` serão mostradas. Exemplo:

```
Sub CommandButton1_Click()  
    Set E3Chart1 = Screen.Item("E3Chart1")  
    E3Chart1.Legend.ShowAllPens = Not _  
        E3Chart1.Legend.ShowAllPens  
End Sub
```

18.9.2.4 ShowHeader

☑ Determina a visibilidade do título da legenda (cabeçalho). Se a opção for configurada para True, o título da legenda é mostrado. Caso contrário, o título fica invisível. Exemplo:

```
Sub CommandButton2_Click()  
    Screen.Item("E3Chart1").Legend.ShowHeader = False  
End Sub
```

18.9.2.5 Size

9 Determina o tamanho da legenda. Este tamanho pode significar a altura ou largura, dependendo da posição da legenda. Exemplo:

```
Sub CommandButton13_Click()  
    MsgBox Screen.Item("E3Chart1").Legend.Size  
End Sub
```

18.9.2.6 Visible

☑ Determina a visibilidade da legenda. Se a opção estiver configurada como True, a Legenda é visível no E3Chart. Caso contrário, fica invisível. Exemplo:

```
Sub CommandButton13_Click()  
    Screen.Item("E3Chart1").Legend.Visible = False  
End Sub
```

18.10 Colunas da Legenda

Esta seção contém informações sobre propriedades do objeto Colunas da Legenda. Este objeto não possui eventos nem métodos associados.

18.10.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Colunas da Legenda.

18.10.1.1 Caption

A Permite definir um título para a coluna.

18.10.1.2 Column

 Retorna o identificador da coluna. Veja a **Tabela de Identificação das Colunas**.

Exemplo:

```
Sub CommandButton15_Click()  
    Set col= Screen.Item("E3Chart1").Legend.Item(1)  
    MsgBox col.Column  
End Sub
```

18.10.1.3 Format

A Configura o formatador usado na coluna. Exemplo:

```
Sub CommandButton15_Click()  
    Set col= Screen.Item("E3Chart1").Legend.Item(1)  
    MsgBox col.Format = "0.0"  
End Sub
```

18.10.1.4 Index

9 Retorna a posição da coluna na Legenda. Exemplo:

```
Sub CommandButton15_Click()  
    Set col = Screen.Item("E3Chart1").Legend.Item(1)  
    MsgBox col.Index  
End Sub
```

18.10.1.5 Name

A Determina o nome da coluna. O nome das colunas pode ser visto na **Tabela de Identificação das Colunas**. Exemplo:

```
Sub CommandButton1_Click()  
    set Chart = screen.item("E3Chart1")  
    set query = Chart.Queries.Item(0)  
    MsgBox query.Name  
    MsgBox query.TableName  
End Sub
```

18.10.1.6 TextAlign

 Retorna o alinhamento do texto da coluna. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para TextAlign

OPÇÃO	DESCRIÇÃO
0 - taLeft	Alinhamento esquerdo.

OPÇÃO	DESCRIÇÃO
1 - taRight	Alinhamento direito.
2 - taCenter	Alinhamento centralizado.

Exemplo:

```
Sub CommandButton1_Click()  
    Set col = Screen.Item("E3Chart1").Legend.Item(1)  
    col.TextAlign = 2  
End Sub
```

18.10.1.7 Width

9 Retorna a largura da coluna. Exemplo:

```
Sub CommandButton1_Click()  
    Set col = Screen.Item("E3Chart1").Legend.Item(1)  
    col.Width = 50  
End Sub
```

CAPÍTULO

19 E3Playback

Esta seção contém informações sobre métodos e propriedades do objeto E3Playback. Este objeto não possui eventos associados.

19.1 Métodos

Esta seção contém informações sobre os métodos do objeto E3Playback.

19.1.1 Pause

Pause()

Congela o relógio de *playback* no tempo atual. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Playback1").Pause()  
End Sub
```

19.1.2 Play

Play()

Inicia a reprodução dos dados no horário atual de *playback*, avançando o relógio de acordo com a velocidade de reprodução atual. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Playback1").Play()  
End Sub
```

19.1.3 Stop

Stop()

Interrompe o *playback*, os dados de tags e alarmes são removidos das telas e nenhuma nova consulta é feita ao banco de dados. Exemplo:

```
Sub CommandButton1_Click()  
    Screen.Item("E3Playback1").Stop()  
End Sub
```

19.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto E3Playback.

19.2.1 CurrentTime

 Mostra a data e a hora atual do objeto E3Playback. Esta propriedade está disponível somente em tempo de execução.

19.2.2 DBServer

A Indica o nome do objeto Banco de Dados que contém os dados históricos do aplicativo.

19.2.3 InitialScreen

A Indica a Tela que será mostrada inicialmente no E3Playback. Se essa propriedade for deixada em branco, será utilizada a tela inicial do objeto Viewer.

19.2.4 PlaybackState

 Informa o estado atual do objeto E3Playback. Esta propriedade está disponível somente em tempo de execução e as opções são as seguintes:

Opções da propriedade PlaybackState

OPÇÃO	DESCRIÇÃO
0 - Parado	O relógio de playback está parado.
1 - Reproduzindo	O relógio de playback está se movendo.
2 - Pausado	O relógio de playback está pausado.

CAPÍTULO

Fórmulas

20

Esta seção contém informações sobre métodos e propriedades do objeto Fórmula. Este objeto não possui eventos associados.

20.1 Métodos

Esta seção contém informações sobre os métodos do objeto Fórmula.

20.1.1 CreateUnit

CreateUnit(*UnitName*)

Cria uma unidade na tabela de fórmulas. Este método possui o parâmetro *UnitName* que determina o nome da unidade a ser criada. Retorna True se conseguir realizar a operação; caso contrário, retorna False. Exemplo:

```
Sub Botao1_Click()  
    Dim val  
    ' Quando clica no botão, cria uma nova Unit  
    Application.GetObject("Formula1").CreateUnit("Unit2")  
End Sub
```

20.1.2 CreateValue

CreateValue(*ValueName*)

Cria um conjunto de valores na tabela de fórmulas. Este método possui o parâmetro *ValueName*, que determina o nome do conjunto a ser criado. Retorna True se conseguir realizar a operação; caso contrário, retorna False. Exemplo:

```
Sub Botao1_Click()  
    Dim val  
    ' Quando clica no botão, cria um novo valor  
    Application.GetObject("Formula1").CreateValue("Modelo5")  
End Sub
```

20.1.3 DeleteUnit

DeleteUnit(*UnitName*)

Apaga uma unidade na tabela de fórmulas. Este método possui o parâmetro *UnitName*, que informa o nome da unidade a ser apagada. Retorna True se conseguir realizar a operação; caso contrário, retorna False. Exemplo:

```
Sub Botao1_Click()  
    Dim val  
    ' Quando clica no botão, apaga a unit  
    Application.GetObject("Formula1").DeleteUnit("Unit2")  
End Sub
```

20.1.4 DeleteValue

DeleteValue(*ValueName*)

Apaga um conjunto de valores na tabela de fórmulas. Este método possui o parâmetro *ValueName*, que informa o conjunto de valores a ser apagado. Retorna True se conseguir realizar a operação; caso contrário, retorna False. Exemplo:

```
Sub Botao1_Click()  
    Dim val  
    ' Quando clica no botão, apaga um conjunto de valores  
    Application.GetObject("Formula1").DeleteValue("Modulo5")  
End Sub
```

20.1.5 FindUnit

FindUnit(*UnitName*)

Verifica se uma determinada unidade existe no banco de dados das fórmulas. Este método possui o parâmetro *UnitName*, que determina o nome da unidade a ser encontrada. O método retorna True se conseguir realizar a operação; caso contrário, retorna False. Exemplo:

```
Sub Botao1_Click()  
    Dim val  
    ' Quando clica no botão, aparece uma caixa de mensagem  
    ' (MessageBox) com o resultado  
    MsgBox(Application.GetObject("Formula1")._  
        FindUnit("Unit2"))  
End Sub
```

20.1.6 FindValue

FindValue(*ValueName*)

Verifica se um determinado conjunto de valores existe no banco de dados de fórmulas. Este método possui o parâmetro *ValueName*, que informa o nome do conjunto a ser verificado. Retorna True se conseguir realizar a operação; caso

contrário, retorna False. Exemplo:

```
Sub Botao1_Click()  
    Dim val  
    ' Quando clica no botão aparece uma caixa de mensagem  
    ' (MessageBox) com o resultado  
    MsgBox CStr(Application.GetObject("Formula1")._  
        FindValue("Modelo5"))  
End Sub
```

20.1.7 GetUnitData

GetUnitData(单位名称, 模板名称, Val)

Coloca na variável indicada em *Val* o tag associado à unidade *单位名称* do template *模板名称*. Retorna True se a operação foi bem-sucedida, ou False, caso contrário. Exemplo:

```
Sub Botao1_Click()  
    Dim qualTag, qualFormula  
    Application.GetObject("Formula1").GetUnitData _  
        "Unidade1", "Template2", var1  
End Sub
```

20.1.8 GetValueData

GetValueData(ValorName, 模板名称, Val)

Coloca na variável *Val* o valor do conjunto de valores *ValorName* que está associado ao template *模板名称*. Retorna True se a operação foi bem-sucedida, ou False, caso contrário. Exemplo:

```
Sub Botao1_Click()  
    Dim valor, qualFormula  
    Application.GetObject("Formula1").GetValueData _  
        "Valores4", "Template2", var1  
End Sub
```

20.1.9 LoadFormulaValuesQuiet

LoadFormulaValuesQuiet(单位名称, ValorName)

Carrega um conjunto de valores para uma unidade de destino, sem apresentar nenhuma mensagem. Este método possui os parâmetros *单位名称*, que determina o nome da unidade e *ValorName*, que determina o nome do conjunto de valores. Retorna True quando é executado com sucesso, e False quando falha (o que não significa um erro de script, necessariamente). Exemplo:

```
Sub Botao1_Click()  
    Application.GetObject("Formula1").LoadFormulaValuesQuiet _  
        "Unidade3", "valor1"  
End Sub
```

NOTA: Este método também é acessado através do objeto **Viewer**.

20.1.10 RenameUnit

RenameUnit(UnitName, NewUnitName)

Renomeia uma determinada unidade existente na tabela de fórmulas. Retorna True se conseguir realizar a operação, ou False, caso contrário. Este método possui os parâmetros *UnitName*, que determina o nome da unidade a ser encontrada; e *NewUnitName*, que informa o novo nome da unidade. Exemplo:

```
Sub Botao1_Click()  
    Dim val  
    ' Quando clica no botão, renomeia uma unit  
    Application.  
        GetObject("Formula1").RenameUnit "Unit2", "Unit3"  
End Sub
```

20.1.11 RenameValue

RenameValue(ValueName, NewValueName)

Renomeia um determinado conjunto de valores existente na tabela de fórmulas. Retorna True se conseguir realizar a operação ou False, caso contrário. Este método possui os parâmetros *ValueName*, que informa o nome do conjunto de valores e *NewValueName*, que informa o novo nome do conjunto de valores. Exemplo:

```
Sub Botao1_Click()  
    Application.GetObject("Formula1").RenameValue "Modelo5", "ModeloABC"  
End Sub
```

20.1.12 SaveFormulaValues

SaveFormulaValues(UnitName, ValueName, [IgnoreErrors])

Este método salva os valores atuais dos Tags de uma unidade de origem em um conjunto de valores na tabela de fórmulas. Este método não faz a verificação dos limites, caso o *template* tenha uma restrição do tipo absoluta. O parâmetro *UnitName* é o nome da unidade de origem e o parâmetro *ValueName* é o conjunto de valores que será salvo. Retorna True se conseguir realizar a operação. Do contrário, retorna False. O parâmetro *IgnoreErrors*, quando em True, faz com que todos os valores sejam gravados, independente de haver erros de associações na Fórmula. Seu valor padrão é False, no entanto. Exemplo:

```
Sub Botao1_Click()  
    Application.GetObject("Formula1") SaveFormulaValues "Unidade1", "Valor1"  
End Sub
```

20.1.13 SetUnitData

SetUnitData(*UnitName*, *TemplateName*, *Data*)

Carrega para a tabela de fórmulas o Tag identificado para um certo modelo (*template*), numa determinada unidade. Retorna True se conseguir realizar a operação ou False, caso contrário. Este método possui os parâmetros *UnitName*, que informa o nome da unidade, *TemplateName*, que informa o nome do *template* do Tag e *Data*, que informa o nome da variável que contém o nome do Tag. Exemplo:

```
Sub Botao1_Click()  
Application.GetObject("Formula1").SetUnitData "Unidade2", "Template5", 50  
End Sub
```

20.1.14 SetValueData

SetValueData(*ValueName*, *TemplateName*, *Data*)

Altera o valor referente a um *template* definido para um determinado conjunto de valores. Este método faz a verificação dos limites, retornando True se conseguir realizar a operação ou False, caso contrário. Este método possui os parâmetros *ValueName*, que determina o nome do conjunto de valores, *TemplateName*, que determina o nome do *template* e *Data*, que determina o nome da variável que possui o valor. Exemplo:

```
Sub Botao1_Click()  
Application.GetObject("Formula1").SetValueData "Unidade2", "Template1", 100  
End Sub
```

20.2 Propriedades

Esta seção contém informações sobre as propriedades do objeto Fórmula.

20.2.1 DBServer

A Indica o nome do banco de dados onde serão gravadas as informações da fórmula, ou seja, as unidades e os conjuntos de valores. O valor padrão desta propriedade é vazio.

20.2.2 ImmediateExecute

Quando habilitada, faz com que a Fórmula escreva seus registros diretamente no Banco de Dados, sem precisar passar por suas filas de operações (arquivos .e3i e .e3o). Isso faz com que as operações feitas no Banco de Dados sejam vistas de forma mais rápida.

20.2.3 TableName

A Indica o nome das tabelas onde ficarão as informações das fórmulas. O valor padrão desta propriedade é vazio.

Bibliotecas de Usuário

Esta seção contém informações sobre os objetos XControls, XObjects e as Propriedades de um ElipseX.

21.1 XControls e XObjects

Esta seção contém informações sobre eventos e propriedades dos objetos XControls e XObjects. Estes objetos não possuem métodos associados.

21.1.1 Eventos

Esta seção contém informações sobre os eventos dos objetos XControls e XObjects.

21.1.1.1 Constructor

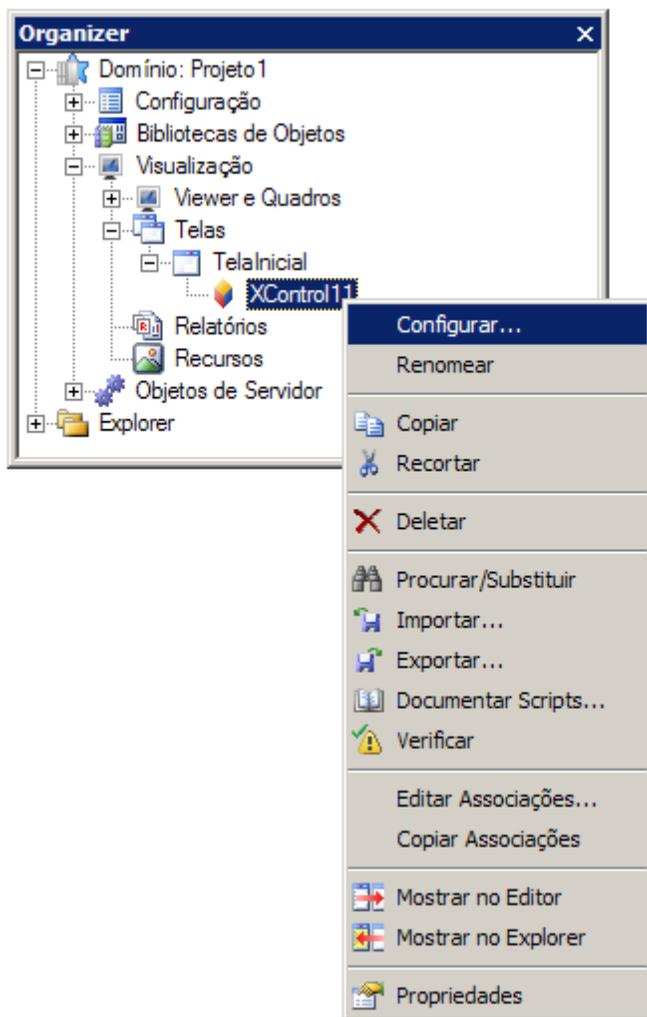
Constructor()

Disparado quando o ElipseX é iniciado. Pode-se utilizar este evento para rodar um script que inicie valores internos de um ElipseX, por exemplo.

21.1.1.2 CustomConfig

CustomConfig()

Permite configurações automatizadas nas instâncias de ElipseX. Uma opção de configuração irá aparecer no menu de contexto das instâncias de ElipseX sempre que existir um script associado ao evento **CustomConfig** da definição do ElipseX. Quando a opção é selecionada no menu, o evento é disparado. O texto que aparece na opção do menu pode ser informado na propriedade **CustomConfigText** da definição do ElipseX. Se esta propriedade estiver em branco, aparecerá o texto "Configurar".



Menu de contexto da instância do ElipseX

NOTA: O script associado ao evento **CustomConfig** roda no E3 Studio, onde os objetos não estão ativos. Portanto, o comportamento será diferente do usual.

21.1.2 Propriedades

Esta seção contém informações sobre as propriedades dos objetos XControls e XObjects.

21.1.2.1 CustomConfigText

A Indica o texto que deverá aparecer no menu para a opção de configuração da instância do ElipseX. A opção só irá aparecer se existir um script associado ao evento **CustomConfig** da definição do ElipseX. Se o valor da propriedade estiver em branco, aparecerá o texto "Configurar" na opção do menu. O valor padrão desta propriedade é vazio.

21.2 Propriedades de um ElipseX

Esta seção contém informações sobre eventos das Propriedades de um ElipseX. Estas propriedades não possuem métodos nem propriedades associadas.

21.2.1 Eventos

Esta seção contém informações sobre os eventos das Propriedades de um ElipseX.

21.2.1.1 OnPropertyChanged

OnPropertyChanged()

Ocorre quando uma propriedade de um ElipseX é modificada. Pode-se utilizar este evento para disparar scripts que executem ações de acordo com determinado status de um ElipseX.

CAPÍTULO

22 Relatórios

Esta seção contém informações sobre eventos, métodos e propriedades do objeto Relatório.

22.1 Eventos

Esta seção contém informações sobre os eventos do objeto Relatório.

22.1.1 OnAfterPrint

OnAfterPrint()

Iniciado após uma seção ter sido montada no relatório. Pode-se utilizar este evento para atualizar qualquer contador cuja utilização seja necessária depois do relatório ser completado.

22.1.2 OnBeforePrint

OnBeforePrint()

Iniciado antes da seção ter sido montada no relatório. Pode-se utilizar este evento para modificar o valor de um objeto no relatório antes que este seja impresso. Recomenda-se que os campos da consulta do relatório não sejam acessados quando este evento estiver sendo utilizado.

22.1.3 OnDataInitialize

OnDataInitialize()

Ocorre antes do evento **OnReportStart**. Este evento permite acrescentar e configurar campos à coleção **Fields** de um relatório, antes de sua geração. Exemplo:

```
Sub OnDataInitialize()  
    Fields.Add "Nome"  
    Fields.Add "Setor"  
    Fields.Add "Código"  
End Sub
```

22.1.4 OnError

OnError(Number, Description, SCode, Source, HelpFile, HelpContext, CancelDisplay)

Gerado por um erro interno no relatório. Se este evento não for tratado, o E3 irá mostrar uma mensagem de erro genérica.

Variáveis do evento OnError

NOME	DESCRIÇÃO
Number	Número inteiro identificador do erro.
Description	String com a descrição do erro.
SCode	Inteiro com o código de erro do subsistema OLE (não utilizado).
Source	String com o objeto origem do erro.
HelpFile	String com o nome e caminho do arquivo de ajuda.
HelpContext	Número de contexto do tópico de ajuda referente ao erro (inteiro).
CancelDisplay	Booleano que indica se o erro deve ser mostrado em uma MessageBox .

22.1.5 OnFetchData

OnFetchData(eof)

Disparado toda vez que um novo registro é processado. Este evento é utilizado para executar um script que modifica os valores dos campos que foram adicionados ao relatório em um script associado ao evento **OnDataInitialize**. A variável *eof* tem o valor padrão True e indica que após o script o processamento do registro atual do relatório foi encerrado.

22.1.6 OnFormat

OnFormat()

Iniciado depois que os dados são lidos e carregados no relatório, mas antes da seção ser preparada para a impressão. Este evento pode ser utilizado para modificar o layout da seção do relatório ou de qualquer outro objeto.

22.1.7 OnHyperlink

OnHyperlink(*Button*, *Link*)

Ocorre quando um link é clicado no relatório. Pode-se utilizar este evento para executar um script que redirecione um link ou para configurar um link no relatório. A variável *Button* indica qual botão foi clicado (normalmente, 1) e a variável *Link* determina qual o endereço que será seguido.

22.1.8 OnNoData

OnNoData()

Ocorre quando não há dados a serem impressos no relatório. Pode-se utilizar este evento para executar um script que mostre uma mensagem de erro na tela acusando a falta de dados para imprimir e cancele o relatório.

22.1.9 OnPageEnd

OnPageEnd()

Ocorre ao final da impressão de cada página do relatório.

22.1.10 OnPageStart

OnPageStart()

Ocorre no início da impressão de cada página do relatório.

22.1.11 OnPrintProgress

OnPrintProgress(*PageNumber*)

Ocorre enquanto uma página do relatório está sendo impressa. A variável *PageNumber* indica o número da página atual.

22.1.12 OnReportEnd

OnReportEnd()

Disparado no final da geração do relatório, após finalizar a sua impressão.

22.1.13 OnReportStart

OnReportStart()

Disparado no início da geração do relatório, antes de começar a sua impressão.

22.2 Métodos

Esta seção contém informações sobre os métodos do objeto Relatório.

22.2.1 Export

Export(*[ExportFilter]*, *[ExportFileName]*)

Imprime um relatório de acordo com o formato especificado no arquivo. Este método possui o parâmetro *strExportFilter*, que determina o filtro para o relatório, indicando o formato para a exportação. Pode assumir as seguintes opções:

- **PDF:** Exporta os dados para o formato Adobe PDF.
- **Excel:** Exporta os dados para o formato de planilha Excel.
- **HTML:** Exporta os dados para o formato HTML.
- **TEXT:** Exporta os dados para um arquivo texto.
- **RTF:** Exporta os dados para o formato Rich Text Format.
- **TIFF:** Exporta os dados para o formato Tag Image File Format.

Ao simplesmente informar o nome de um filtro, como exposto acima, os dados serão exportados utilizando-se as propriedades em comum de cada filtro. Pode-se modificar as propriedades comuns de um filtro de exportação através do método **GetExportFilter** antes de exportar os dados. O nome do arquivo deve ser informado no parâmetro *strExportFileName*. Exemplo:

```
Sub Button1_Click()  
    Set report = Application.LoadReport("[Relatório3]")  
    Select case Application._  
        SelectMenu("PDF|Excel|HTML|RTF|Texto|TIFF|Texto(CSV)")  
        Case 1  
            Report.Export "PDF", "C:\correio\reports\report.pdf"  
            MsgBox "Exportado para PDF!"  
        Case 2  
            Report.Export "EXCEL", "C:\correio\reports\report.xls"  
            MsgBox "Exportado para XLS!"  
        Case 3  
            Report.Export "HTML", "C:\correio\reports\report.html"  
            MsgBox "Exportado para HTML!"  
        Case 4  
            Report.Export "RTF", "C:\correio\reports\report.rtf"  
            MsgBox "Exportado para RTF!"  
        Case 5  
            Report.Export "TEXT", "C:\correio\reports\report.txt"  
            MsgBox "Exportado para TXT!"  
        Case 6  
            Report.Export "TIFF", "C:\correio\reports\report.tiff"  
            MsgBox "Exportado para TIFF!"  
        Case 7  
            Set reportFilter = report.GetExportFilter("TEXT")  
            reportFilter.FileName = "C:\correio\reports\report2.txt"  
            reportFilter.TextDelimiter = ","  
            report.Export reportFilter  
            MsgBox "Exportado para TXT usando filtro!"  
    End Select  
End Sub
```

22.2.2 GetExportFilter

GetExportFilter(*FilterName*)

Retorna um objeto que especifica a customização dos parâmetros de exportação. Este método possui o parâmetro *FilterName*, que determina o filtro para o relatório, indicando o tipo de formato para a exportação. Pode assumir as seguintes opções:

- **PDF**: Exporta os dados para o formato Adobe PDF.
- **Excel**: Exporta os dados para o formato de planilha Excel.
- **HTML**: Exporta os dados para o formato HTML.
- **TEXT**: Exporta os dados para um arquivo texto.
- **RTF**: Exporta os dados para o formato Rich Text Format.
- **TIFF**: Exporta os dados para o formato Tag Image File Format.

Depois de obtido o filtro, as seguintes propriedades podem ser modificadas:

Propriedades passíveis de modificação

PROPRIEDADE	FILTRO	DESCRIÇÃO
AutoRowHeight	Excel	Em True (padrão), configura a altura da linha automaticamente. Em False, configura a altura para o maior elemento na linha.
BorderSpace	Excel	Espaçamento mínimo entre as células. O valor padrão é de 59 twips.
CreateCSSFile	HTML	Se verdadeiro, gera um arquivo CSS no diretório <i>TMLOutputPath</i> .
DoubleBoundaries	Excel	Em True, indica que os elementos alinhados à direita devem repor os alinhados à esquerda na mesma coluna. Caso contrário, deixe em falso para liberar mais espaço.
ExportRange	HTML	Indica uma faixa de páginas para serem exportadas. Por exemplo, "1, 2, 3-9, 14".
FaxExport	TIFF	Objeto que permite exportar os dados no formato RFC 1314 TIFF.
FileName	Todos	Informa o nome do arquivo para o qual os dados serão exportados.
GenPageBreaks	HTML	Em True, colocará quebras de página abaixo do elemento mais inferior em cada página do relatório.
HTMLOutputPath	HTML	Caminho padrão para os arquivos HTML.
JPGQuality	PDF	Indica o nível de qualidade das imagens exportadas (de 0 a 100).
MinColumnWidth	Excel	Tamanho mínimo da coluna. O valor padrão é 1011 twips.
MinRowHeight	Excel	Tamanho mínimo da linha.
MultiSheet	Excel	Em True, cada página no relatório vai para uma planilha separada.

PROPRIEDADE	FILTRO	DESCRIÇÃO
PageDelimiter	TEXT	Configura ou retorna o caractere delimitador entre páginas.
SuppressEmptyLines	TEXT	Retira ou Insere linhas vazias, para efeito de layout.
TextDelimiter	HTML	Configura ou retorna o caractere delimitador entre os textos.
TrimEmptySpace	Excel	Em True, o espaço vertical entre os elementos será eliminado. O padrão é False.
Unicode	TEXT	Determina se o texto vai ser salvo em formato Unicode (16 bits).
WebCacheOutput	HTML	Em True, o relatório será exportado para o serviço de WebCache. Caso contrário (padrão), não será exportado.

22.2.3 Print

Print()

Imprime um relatório. Exemplo:

```
Sub Rect_Click()
    Application.LoadReport("[Relatório3]").Print()
End Sub
```

22.2.4 PrintPreview

PrintPreview()

Faz uma prévia de impressão de relatório na Tela. Se o relatório for mostrado corretamente na tela, retorna True. Caso o usuário aperte o botão **Cancel** ou aconteça algum erro, retorna False. Exemplo:

```
Sub CommandButton1_Click()
    Set report = Application.LoadReport("[Relatório1]")
    Inicio = Application.GetObject("Dados.Grafico.datai").Value
    Final = Application.GetObject("Dados.Grafico.dataf").Value
    report.Item("consulta1").SetVariableValue "Ini", Inicio
    report.Item("consulta1").SetVariableValue "Fim", Final
    report.PrintPreview()
End Sub
```

22.2.5 Query

Query()

Retorna o objeto Consulta atualmente selecionado no relatório, informado através da propriedade **SourceQuery** (pode haver mais de uma consulta no relatório).

Exemplo:

```
Sub Rect_Click()  
    Set Query = Application.LoadReport("[Relatório3]").Query()  
    Query.SetVariableValue("Chave1", "XYZ")  
End Sub
```

22.3 Propriedades

Para criar um script no relatório, utiliza-se o Editor de Scripts do Relatório, acessado através do botão **Editor de Scripts** e para visualizar o relatório, utiliza-se o botão **Gerar Relatório**, ambos localizados na Barra de Ferramentas do Relatório.

Os scripts do relatório utilizam alguns procedimentos, dependendo do objeto ou da seção que se deseja incluir o código.

Por exemplo:

```
Report.Sections("PageHeader").Controls("E3Chart1")._GridBkColor= RGB(255, 0, 255).
```

onde:

- **PageHeader**: é o nome da seção onde o objeto está incluso no relatório.
- **E3Chart1**: é o nome do objeto que está dentro da seção especificada, no caso o PageHeader.
- **GridBkColor**: é o nome da propriedade do objeto, neste caso o E3Chart.
- **RGB (255, 0, 255)**: é o parâmetro ou ação da propriedade. Neste caso, mudar a cor de fundo do gráfico para rosa.

Sendo assim, para fazer-se um script no relatório, usa-se o seguinte conceito:

```
Report.Sections("NomeDaSeção").Controls("NomeDoObjeto")._NomeDaPropriedade=parâmetros_da_propriedade
```

NOTA: O objeto Relatório encapsula um objeto ActiveReports (ou AR), que é o relatório propriamente dito.

22.3.1 Caption

A Contém o título do relatório que aparece na barra de título da janela de visualização (*preview*). O valor padrão desta propriedade é vazio.

22.4 Layout

Esta seção contém informações sobre propriedades do objeto Layout do Relatório. Este objeto não possui eventos nem métodos associados.

22.4.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Layout do Relatório.

NOTA: As propriedades aqui descritas são parte do objeto ActiveReport (ou AR) que é encapsulado no objeto Relatório. Estas propriedades são válidas apenas dentro do escopo do AR e não podem ser acessadas fora desse objeto.

22.4.1.1 `_PageBottomMargin`

9 Determina a margem inferior do relatório, em twips (1 twip = 1/1440 polegada). O valor padrão desta propriedade é 1440 (1 pol ou 2,54 cm).

22.4.1.2 `_PageLeftMargin`

9 Determina a margem esquerda do relatório, em twips (1 twip = 1/1440 polegada). O valor padrão desta propriedade é 1440 (1 pol ou 2,54 cm).

22.4.1.3 `_PageRightMargin`

9 Determina a margem direita do relatório, em twips (1 twip = 1/1440 polegada). O valor padrão desta propriedade é 1440 (1 pol ou 2,54 cm).

22.4.1.4 `_PageTopMargin`

9 Determina a margem superior do relatório, em twips (1 twip = 1/1440 polegada). O valor padrão desta propriedade é 1440 (1 pol ou 2,54 cm).

22.4.1.5 `AllowSplitters`

Permite que a tela de visualização do relatório seja dividida em duas partes. Essa propriedade só está disponível em tempo de execução. Se a propriedade estiver configurada para `False` (valor padrão), a barra para divisão não aparece na tela.

22.4.1.6 `documentName`

A Determina o nome do documento para o relatório. Este nome aparece no gerenciador de impressão e pode ser usado para identificar mais facilmente o relatório. O valor padrão desta propriedade é "ActiveReports Document".

22.4.1.7 `MaxPages`

9 Estabelece o número máximo de páginas para o relatório. Quando este número é alcançado, o E3 pára de processar o documento. O valor padrão desta propriedade é 10.

22.4.1.8 `ParentReport`

A Esta propriedade é uma variável de uso interno do sistema e contém uma referência para o objeto relatório. A propriedade é somente de leitura, e válida apenas para os eventos `OnDataInitialize` e `OnReportEnd`.

22.4.1.9 `PrintWidth`

9 Determina a largura da área de impressão do relatório, em twips. Se o tamanho do relatório for mudado em tempo de execução, a largura de impressão também deverá ser ajustada, para garantir que o relatório irá ocupar toda a área de impressão. O tamanho da área de impressão deve também incluir a largura das margens, para que o relatório não fique maior que o tamanho do papel. Se isto acontecer, o erro será assinalado por uma linha pontilhada vermelha impressa em cada página do relatório.

22.4.1.10 RulerVisible

Quando configurado como True, indica que serão mostradas uma régua vertical e outra horizontal na janela de visualização do relatório. Do contrário, as régua ficarão invisíveis.

22.4.1.11 ScriptDebuggerEnabled

Habilita ou desabilita o depurador do ActiveReports (JIT), para depurar os scripts associados aos relatórios. Este depurador não está disponível em todo o E3, apenas nos relatórios.

22.4.1.12 ScriptLanguage

A Indica a linguagem usada para interpretar os scripts associados a um relatório. A linguagem padrão é VBScript, mas JScript também pode ser utilizada.

22.4.1.13 ShowParameterUI

Habilita ou desabilita os parâmetros da caixa de diálogo da Consulta, que aparece quando o relatório está sendo executado. Se a propriedade estiver configurada para True, os parâmetros da caixa de diálogo da consulta serão mostrados. Caso contrário, estes parâmetros não são mostrados.

22.4.1.14 Status

 Retorna o status do relatório. As opções disponíveis para esta propriedades são as seguintes:

Opções disponíveis para Status

OPÇÃO	DESCRIÇÃO
0 - DDStatIdle	Indica que o relatório está fechado.
1 - DDStartRunning	Indica que o relatório está sendo executado.
2 - DDStartCompleted	Indica que o relatório está completo.
3 - DDStartCanceled	Indica que o relatório foi cancelado.

22.4.1.15 TOCEnabled

Habilita ou desabilita o índice do relatório. Se a propriedade estiver configurada para True, o índice do relatório é habilitado; do contrário, o relatório não terá índice. O valor padrão desta propriedade é True.

22.4.1.16 TOCVisible

Determina a visibilidade do índice do relatório. Se a propriedade estiver configurada para True, o índice do relatório é mostrado; do contrário, o índice fica invisível. O valor padrão desta propriedade é True.

22.4.1.17 ToolbarVisible

Habilita ou desabilita a Barra de Ferramentas da janela de prévia de impressão do relatório. Se a propriedade estiver configurada para True, a barra é habilitada; do contrário, não haverá barra de ferramentas nesta janela.

22.4.1.18 UserData

A Configura ou retorna as informações específicas do usuário. Esta propriedade é similar à propriedade **Tag** do Visual Basic, mas será exportada e salva em um arquivo .rpx. Pode ser usada para salvar e carregar qualquer informação personalizada necessária ao desenho do relatório.

22.4.1.19 Version

A Retorna o número da versão do produto.

22.4.1.20 WaterMark

 Adiciona uma imagem de fundo ao relatório (marca d'água). As marcas d'água são textos ou figuras que aparecem sob o texto do documento. Normalmente, elas tornam o documento visualmente mais interessante.



Exemplo de marca d'água

O valor padrão desta propriedade é vazio (sem imagem).

22.4.1.21 WaterMarkAlignment

 Determina o alinhamento marca d'água no relatório. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para WaterMarkAlignment

OPÇÃO	DESCRIÇÃO
0 - ddPATopLeft	Alinha a imagem ao topo e à esquerda.
1 - ddPATopRight	Alinha a imagem ao topo e à direita.
2 - ddPACenter	Alinha a imagem ao centro (padrão).
3 - ddPABottomLeft	Alinha a imagem à base e à esquerda.
4 - ddPABottonRight	Alinha a imagem à base e à direita.

22.4.1.22 WaterMarkPrintOnPages

9 Indica o número das páginas do relatório que receberão a marca d'água. A sintaxe utilizada pode incluir uma única página, uma extensão de páginas, ou mesmo uma combinação de ambos. Exemplo: 1, 5-8, 9, 10-15.

22.4.1.23 WaterMarkSizeMode

9 Configura o efeito do tamanho da marca d'água na página do relatório. As opções disponíveis são as seguintes:

Opções disponíveis para WaterMarkSizeMode

OPÇÃO	DESCRIÇÃO
0 - ddSMClip	A marca d'água é mostrada no relatório no seu tamanho original.
1 - ddSMStretch	A marca d'água irá preencher toda a página do relatório.
2 - ddSMZoom	A marca d'água será aumentada até ficar do tamanho da página do relatório.

22.5 Seção

Esta seção contém informações sobre propriedades comuns do objeto Seção do Relatório. Este objeto não possui eventos nem métodos associados.

22.5.1 Propriedades Comuns

Esta seção contém informações sobre as propriedades comuns do objeto Seção do Relatório.

22.5.1.1 BackColor

 Especifica a cor de fundo da seção do relatório. O efeito desta propriedade só será visível se a propriedade **BackStyle** estiver habilitada para a opção 1 - **ddbKNormal**. O valor padrão é branco (**RGB(255, 255, 255)**).

22.5.1.2 BackStyle

 Especifica o estilo de fundo da seção do relatório. As opções disponíveis para esta propriedade são as seguintes:

- **0 - ddBKTransparent:** fundo transparente.
- **1 - ddBKNormal:** fundo normal.

22.5.1.3 CanGrow

Determina a aplicação de *stretch* ao texto da página do relatório. Caso a largura ou a altura da página sejam aumentadas, o texto acompanha esta variação. Se a propriedade estiver configurada para True, o texto acompanha as variações de altura e de largura do objeto. Caso contrário, este permanece com as configurações iniciais. O valor padrão desta propriedade é True.

22.5.1.4 CanShrink

Determina a aplicação do efeito de encolhimento ao texto da página do relatório. Caso a largura ou a altura da página sejam diminuídas, o texto acompanha esta variação. Se a propriedade estiver configurada para True, o texto acompanha as variações de altura e largura do objeto. Caso contrário, este permanece com as configurações iniciais. O valor padrão desta propriedade é True.

22.5.1.5 height

 Determina a altura da seção da página do relatório. O valor padrão desta propriedade é 360.

22.5.1.6 IsRepeating

Determina a repetição da seção na última página do relatório. Se a propriedade estiver configurada para True, a seção será repetida na última página. Caso contrário, não haverá esta repetição.

22.5.1.7 Name

A Indica o nome da seção do relatório.

22.5.1.8 Type

 Retorna o tipo da seção. As opções disponíveis são:

Opções disponíveis para Type

OPÇÃO	DESCRIÇÃO
0 - ReportHeader	Seção tipo ReportHeader (cabeçalho de relatório).
1 - ReportFooter	Seção tipo ReportFooter (rodapé de relatório).
2 - PageHeader	Seção tipo PageHeader (cabeçalho de página).
3 - PageFooter	Seção tipo PageFooter (rodapé de página).
4 - GroupHeader	Seção tipo GroupHeader (cabeçalho de grupo).
5 - GroupFooter	Seção tipo GroupFooter (rodapé de grupo).
6 - Detail	Seção tipo Detail (área para conteúdo do relatório).

22.5.1.9 Visible

Habilita ou desabilita a visibilidade da seção no relatório. Se a propriedade estiver configurada para True, a seção é visível no relatório. Caso contrário, não é visível. O valor padrão desta propriedade é True.

22.5.2 GroupHeader

Esta seção contém informações sobre propriedades do objeto GroupHeader do Relatório. Este objeto não possui eventos nem métodos associados.

22.5.2.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto **GroupHeader** do Relatório.

22.5.2.1.1 ColumnLayout

Determina se o **GroupHeader** irá utilizar o mesmo layout das colunas configuradas na seção **Detail**. Se a propriedade estiver configurada para **True**, o número de colunas na seção **Detail** será o mesmo do **GroupHeader** ou **GroupFooter** associados. Caso contrário, ele permanece com as configurações padrão.

22.5.2.1.2 DataField

A Retorna os dados dos campos do relatório. Define um campo obrigatório para um grupo dentro do conteúdo da seção **Detail**. Este valor é ajustado ao nome de todos os campos na fonte de dados do relatório ou ao nome de um campo customizado que está inserido na coleção de campos. Quando esta propriedade é ajustada, o relatório cria um grupo novo cada vez que o valor do campo muda nos registros de dados do detalhe.

22.5.2.1.3 GrpKeepTogether

 Determina se a seção **GroupHeader** será impressa como um bloco único na mesma página do relatório. As opções disponíveis são as seguintes:

Opções disponíveis para GrpKeepTogether

OPÇÃO	DESCRIÇÃO
0 - GrpNone	A página pode ser quebrada imediatamente depois de um GroupHeader .
1 - GrpFirstDetails	O GroupHeader será impresso com a primeira seção Detail da mesma página ou coluna do relatório.
2 - GrpAll	O GroupHeader , Detail e Footer de grupo serão impressos juntos na mesma página do relatório.

O valor padrão desta propriedade é **0 - GrpNone**.

22.5.2.1.4 KeepTogether

 Determina se as seções do relatório serão impressas como um único bloco, na mesma página. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para KeepTogether

OPÇÃO	DESCRIÇÃO
0 - ddGrpNone	Há uma quebra de página depois do relatório.
1 - ddGrpFirstDetail	O relatório imprimirá a seção Detail na mesma página ou coluna.

22.5.2.1.5 NewColumn

 Insere uma nova quebra de coluna antes ou depois da impressão da seção no relatório. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para NewColumn

OPÇÃO	DESCRIÇÃO
0 - ddNPNone	Não há quebra de página na seção.
1 - ddNBefore	Inicia a impressão da seção em uma nova página.
2 - ddNPAfter	Inicia uma nova página depois de imprimir a seção.
3 - ddNPBeforeAfter	Inicia a impressão em uma nova página e uma nova página depois de imprimir a seção.

22.5.2.1.6 NewPage

 Insere uma quebra de página no relatório. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para NewPage

OPÇÃO	DESCRIÇÃO
0 - ddNPNone	Não há quebra de página na seção (padrão).

OPÇÃO	DESCRIÇÃO
1 - ddNBefore	Inicia a impressão numa nova página.
2 - ddNPAfter	Inicia uma nova página depois da impressão da seção.
3 - ddNPBeforeAfter	Inicia a impressão numa nova página e inicia uma nova página depois de imprimir a seção.

22.5.2.1.7 Repeat

 Determina se o **GroupHeader** será impresso novamente depois de ser associado à seção **Detail** quando há múltiplas páginas, colunas ou quebras de páginas no relatório. As opções disponíveis são as seguintes:

Opções disponíveis para Repeat

OPÇÃO	DESCRIÇÃO
0 - ddRepeatNone	Não há reimpressão do grupo de cabeçalho (padrão).
1 - ddRepeatOnPage	Imprime o grupo de cabeçalhos no topo da página conforme as especificações da seção Detail .
2 - ddRepeatOnColumn	Imprime o grupo de cabeçalhos no topo da coluna da página do relatório conforme as especificações da seção Detail .
3 - ddRepeatAll	Imprime o grupo de cabeçalhos e os demais objetos no topo da página do relatório conforme as especificações da seção Detail .

22.5.2.1.8 UnderlayNext

Determina se a seção deve imprimir uma seção depois da outra, consecutivamente. Se a propriedade estiver configurada para True, a seção seguinte começará a impressão a partir da coordenada superior da seção na página do relatório. Caso contrário, este recurso não é utilizado. O valor padrão desta propriedade é False.

22.5.3 Detail

Esta seção contém informações sobre propriedades do objeto Detail do Relatório. Este objeto não possui eventos nem métodos associados.

22.5.3.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Detail do Relatório.

22.5.3.1.1 ColumnCount

 Determina o número de colunas na seção **Detail** do relatório. A largura de cada coluna deve ser igual a área imprimível do relatório, dividida pelo número de colunas. O valor padrão desta propriedade é 1.

22.5.3.1.2 ColumnDirection

 A propriedade **ColumnDirection** determina a direção da impressão das colunas da seção **Detail**. As opções disponíveis são as seguintes:

Opções disponíveis para ColumnDirection

OPÇÃO	DESCRIÇÃO
0 - ddCDDownAcross	Imprime cada coluna da seção Detail de cima para baixo e depois passa para a próxima coluna à direita.
1 - ddCDAcrossDown	Imprime cada coluna da seção Detail da direita para a esquerda e assim por diante.

O layout é determinado conforme a opção configurada:

Opção 0- ddCDDownAcross



Opção 1- ddCDAcrossDown



22.5.3.1.3 ColumnSpacing

Determina o espaçamento da coluna na seção **Detail**. O valor padrão desta propriedade é 0.

22.5.3.1.4 KeepTogether

 Determina se as seções do relatório serão impressas como um único bloco, na mesma página. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para KeepTogether

OPÇÃO	DESCRIÇÃO
0 - ddGrpNone	Há uma quebra de página depois do relatório.
1 - ddGrpFirstDetail	O relatório imprimirá a seção Detail na mesma página ou coluna.

22.5.3.1.5 NewColumn

 Insere uma nova quebra de coluna antes ou depois da impressão da seção no relatório. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para NewColumn

OPÇÃO	DESCRIÇÃO
0 - ddNPNone	Não há quebra de página na seção.
1 - ddNBefore	Inicia a impressão da seção em uma nova página.
2 - ddNPAfter	Inicia uma nova página depois de imprimir a seção.
3 - ddNPBeforeAfter	Inicia a impressão em uma nova página e uma nova página depois de imprimir a seção.

22.5.3.1.6 NewPage

 Insere uma quebra de página no relatório. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para NewPage

OPÇÃO	DESCRIÇÃO
0 - ddNPNone	Não há quebra de página na seção.

OPÇÃO	DESCRIÇÃO
1 - ddNBefore	Inicia a impressão em uma nova página.
2 - ddNPAfter	Inicia uma nova página depois da impressão da seção.
3 - ddNPBeforeAfter	Inicia a impressão em uma nova página e inicia uma nova página depois de imprimir a seção.

22.5.4 GroupFooter

Esta seção contém informações sobre propriedades do objeto GroupFooter do Relatório. Este objeto não possui eventos nem métodos associados.

22.5.4.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto GroupFooter do Relatório.

22.5.4.1.1 ColumnLayout

Determina se o **GroupFooter** irá utilizar o mesmo layout das colunas configuradas na seção **Detail**. Se a propriedade estiver configurada para True, o número de colunas na seção **Detail** irá refletir no **GroupHeader** ou no **GroupFooter** associados. Caso contrário, permanece com as configurações padrão. O valor padrão desta propriedade é True.

22.5.4.1.2 KeepTogether

 Determina se as seções do relatório serão impressas como um único bloco, na mesma página. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para KeepTogether

OPÇÃO	DESCRIÇÃO
0 - ddGrpNone	Há uma quebra de página depois do relatório.
1 - ddGrpFirstDetail	O relatório imprimirá a seção Detail na mesma página ou coluna.

22.5.4.1.3 NewColumn

 Insere uma nova quebra de coluna antes ou depois da impressão da seção no relatório. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para NewColumn

OPÇÃO	DESCRIÇÃO
0 - ddNPNone	Não há quebra de página na seção.
1 - ddNBefore	Inicia a impressão da seção em uma nova página.
2 - ddNPAfter	Inicia uma nova página depois de imprimir a seção.
3 - ddNPBeforeAfter	Inicia a impressão em uma nova página e uma nova página depois de imprimir a seção.

22.5.4.1.4 NewPage

 Insere uma quebra de página no relatório. As opções disponíveis para esta propriedade são as seguintes:

Opções disponíveis para NewPage

OPÇÃO	DESCRIÇÃO
0 - ddNPNone	Não há quebra de página na seção (padrão).
1 - ddNBefore	Inicia a impressão em uma nova página.
2 - ddNPAfter	Inicia uma nova página depois da impressão da seção.
3 - ddNPBeforeAfter	Inicia a impressão em uma nova página e inicia uma nova página depois de imprimir a seção.

22.5.4.1.5 PrintAtBottom

☑ Determina se o **GroupFooter** ou o **ReportFooter** serão impressos no pé da página. Se a propriedade estiver configurada para True e o relatório contiver um **PageFooter**, o **GroupFooter** e o **ReportFooter** do relatório serão impressos acima da seção **PageFooter**. Configurando mais de uma seção para imprimir o pé da página do relatório, fará com que as seções seguintes do rodapé sejam impressas em páginas separadas.

22.6 Objetos

Esta seção contém informações sobre propriedades dos objetos do Relatório. Estes objetos não possuem eventos nem métodos associados.

22.6.1 Propriedades Comuns

Esta seção contém informações sobre as propriedades comuns aos objetos do Relatório.

22.6.1.1 BackColor

🌈 Especifica a cor de fundo do objeto no relatório. O efeito desta propriedade só será visível se a propriedade **BackStyle** estiver habilitada para a opção **1 - ddBKNormal**. O valor padrão desta propriedade é branco (**RGB(255, 255, 255)**).

NOTA: esta propriedade não é válida para os objetos Linha e Quebra de Página.

22.6.1.2 BackStyle

📄 Especifica o estilo do fundo dos objetos do relatório. As opções disponíveis para esta propriedade são as seguintes:

- **0 - ddBKTransparent:** transparente (mostra a cor definida pela propriedade **BackColor** da seção).
- **1 - ddBKNormal:** normal (mostra a cor definida pela propriedade **BackColor** do objeto).

O valor padrão é **0 - ddBKTransparent**.

NOTA: esta propriedade não é válida para os objetos Linha, Código de Barras, Quebra de Página e Tabela.

22.6.1.3 height

9 Esta propriedade determina a altura do objeto no relatório.

NOTA: esta propriedade não é válida para os objetos Linha e Quebra de Página.

22.6.1.4 left

9 Retorna o valor do posicionamento esquerdo do objeto no relatório.

NOTA: esta propriedade não é válida para os objetos Linha e Quebra de Página.

22.6.1.5 Name

A Determina o nome do objeto. O valor padrão desta propriedade é vazio.

22.6.1.6 Tag

A Retorna o tipo do tag associado ao objeto, isto é, **Boolean**, **String**, **Integer**, etc.

22.6.1.7 Top

9 Retorna o valor do topo do objeto.

NOTA: esta propriedade não é válida para o objeto Linha.

22.6.1.8 Visible

 Determina a visibilidade do objeto no relatório. Se a opção estiver configurada para True, o objeto é visível no relatório. Caso contrário, o objeto está oculto. O valor padrão desta propriedade é True.

NOTA: esta propriedade não é válida para o objeto Quebra de Página.

22.6.1.9 Width

 Esta propriedade determina a largura do objeto no relatório.

NOTA: esta propriedade não é válida para os objetos Linha e Quebra de Página.

22.6.2 Código de Barras

Esta seção contém informações sobre propriedades do objeto Código de Barras do Relatório. Este objeto não possui eventos nem métodos associados.

22.6.2.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Código de Barras do Relatório.

22.6.2.1.1 Alignment

 Determina o alinhamento do texto do objeto no relatório. As opções disponíveis são as seguintes:

- **0 - ddtxLeft:** alinha o texto à esquerda do objeto.
- **1 - ddtxRight:** alinha o texto à direita do objeto.
- **2 - ddtxCenter:** alinha o texto ao centro do objeto.

22.6.2.1.2 BarWidth

9 Determina a largura das barras do Código de Barras. Configurando a largura para 1, a barra do objeto é expandida em até 15 pontos, e assim por diante. Quanto maior o número configurado na propriedade, maior a largura da barra do código de barras. O valor padrão desta propriedade é 1.

22.6.2.1.3 Caption

A Contém o texto do objeto propriamente dito. O valor padrão desta propriedade é vazio.

22.6.2.1.4 CaptionPosition

 Especifica a posição do texto da propriedade **Caption** no objeto. As opções disponíveis para esta propriedade são as seguintes:

- **0 - ddbcCaptionNone:** o texto da propriedade **Caption** não aparece no código de barras.
- **1 - ddbcCaptionAbove:** o texto da propriedade **Caption** aparece acima do código de barras.
- **2 - ddbcCaptionBelow:** o texto da propriedade **Caption** aparece abaixo do código de barras.

O valor padrão desta propriedade é **0 - ddbcCaptionNone**.

22.6.2.1.5 DataField

A Configura ou retorna o dado associado ao objeto. Este dado associado pode ser um campo da tabela no Banco de Dados fornecido pelo objeto Consulta; uma expressão matemática com campos da Consulta e funções do VBScript (neste caso, o campo deve ser precedido do símbolo de igual); ou um tag ou propriedade do E3. Neste caso o valor atual da variável será mostrado ao realizar a impressão. O valor padrão desta propriedade é vazio.

NOTA: O servidor deve estar sendo executado para que o valor da variável possa ser capturado.

22.6.2.1.6 Direction

 Determina a orientação espacial dos códigos de barra. As opções disponíveis nesta propriedade são as seguintes:

Opções disponíveis para Direction

OPÇÃO	DESCRIÇÃO
0 - ddbcLeftToRight	O código de barras é direcionado da esquerda para a direita.
1 - ddbcRightToLeft	O código de barras é direcionado da direita para a esquerda.
2 - ddbcTopToBottom	O código de barras é direcionado de baixo para cima.
3 - ddbcBottomToTop	O código de barras é direcionado de cima para baixo.

O valor padrão desta propriedade é **0 - ddbcLeftToRight**.

22.6.2.1.7 EnableChecksum

Habilita ou desabilita a leitura do valor de CheckSum (Caractere de objeto do código de barras). Se a propriedade estiver habilitada para False, somente códigos com CheckSum serão afetados.

22.6.2.1.8 Font

A Determina a fonte do objeto no texto estabelecido pela propriedade **Caption**. O valor padrão desta propriedade é Arial.

NOTA: Esta propriedade não é usada em scripts ou em associações, sendo editada somente via E3 Studio.

22.6.2.1.9 Forecolor

 Especifica a cor de preenchimento de frente do objeto. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

22.6.2.1.10 Style

 Determina o estilo do código de barras. As opções disponíveis são as seguintes:

Opções disponíveis para Style

OPÇÃO	DESCRIÇÃO
0 - ddbcNone	Estilo de código de barras padrão.
1 - ddbcAnsi39	Estilo de código de barras ANSI 3 de 9 (Código 39). Utilize letras, números, -, *, \$, /, +, %, etc.
2 - ddbcAnsi39x	Estilo de código de barras ANSI estendido 3 de 9 (Código estendido 39). Utilize caracteres completos ASCII.
3 - ddbcCode_2_of_5	Estilo de código de barras 2 de 5. Utilize somente números.
4 - ddbcCode25intlv	Estilo de código de barras intercalado 2 de 5. Utilize somente números.
5 - ddbcCode25mat	Estilo de código de barras Matriz 25.
6 - ddbcCode39	Estilo de código de barras Código 39, utilize letras, números, -, *, \$, /, +, %, etc.
7 - ddbcCode39x	Estilo de código de barras Código 39 estendido. Utilize caracteres completos ASCII.
8 - ddbcCode_128_a	Estilo de código de barras 128 A. Utilize números, pontuações, ou letras.
9 - ddbcCode_128_b	Estilo de código de barras 128 B. Utilize strings, números, pontuações, ou letras.
10 - ddbcCode_128_c	Estilo de código de barras 128 C. Utilize somente

OPÇÃO	DESCRIÇÃO
	números.
11 - ddbcCode_128auto	Estilo de código de barras 128 Automático. Utilize caracteres completos ASCII. Seleciona automaticamente códigos entre 128 A, B e C para setar no código de barras o valor menor.
12 - ddbcCode_93	Estilo de código de barras Código 93. Utilize letras, números, -,*,\$, /,+%, etc.
13 - ddbcCode_93x	Estilo de código de barras Código 93 estendido. Utilize caracteres completos ASCII.
14 - ddbcMSI	Estilo de código de barras Código MSI. Utilize somente números.
15 - ddbcPostNet	Estilo de código de barras PostNet. Utilize somente números com uma verificação digital.
16 - ddbcCodabar	Estilo de código de barras. Utilize A, B, C, D, +, -, :, /, ou números.
17 - ddbcEAN_8	Estilo de código de barras EAN-8. Utilize somente números (7 números e verificação digital).
18 - ddbcEAN_13	Estilo de código de barras EAN-13. Utilize somente números (12 números e verificação digital).
19 - ddbcUPC_A	Estilo de código de barras UPC-A. Utilize somente números (11 números e verificação digital).
20 - ddbcUPC_EO	Estilo de código de barras UPC-E1. Utilize somente números. Usado para símbolos de compressão-zero UPC. Na propriedade Caption , pode-se entrar com 6 dígitos do código UPC-E ou com 11 dígitos. Se um código com 11 dígitos é inserido, o código de barras converte para 6 dígitos UPC-E, se possível. Senão, converte-se de 11 para 6 dígitos UPC-E e nada é mostrado.
21 - ddbcUPC_E1	Estilo de código de barras UPC-E1. Utilize somente números. A largura da entrada de dados do UPC E1 é de 6 caracteres numéricos.

OPÇÃO	DESCRIÇÃO
22 - ddbcRM4SCC	Estilo de código de barras Royal Mail RM4SCC. Utilize somente letras e números (com verificação digital). Este código de barras é utilizado no Reino Unido.
23 - ddbcUCCEAN128	Estilo de código de barras UCC/EAN_128. Utilize caracteres completos ASCII. A versão especial do código 128 é usado na aplicação HIBC.

22.6.3 Elipse, Retângulo e Retângulo Arredondado

Esta seção contém informações sobre propriedades dos objetos Elipse, Retângulo e Retângulo Arredondado do Relatório. Estes objetos não possuem eventos nem métodos associados.

22.6.3.1 Propriedades

Esta seção contém informações sobre as propriedades dos objetos Elipse, Retângulo e Retângulo Arredondado do Relatório.

22.6.3.1.1 LineColor

 Especifica a cor da linha do objeto. O valor padrão desta propriedade é preto (**RGB (0, 0, 0)**).

22.6.3.1.2 LineStyle

 Determina o estilo da linha do objeto. As opções disponíveis desta propriedade são as seguintes:

Opções disponíveis para LineStyle

OPÇÃO	DESCRIÇÃO
0 - ddLSTransparent	A linha do objeto fica transparente.
1 - ddLSSolid	A linha do objeto aparece sólida.
2 - ddLSDash	A linha do objeto fica tracejada.

OPÇÃO	DESCRIÇÃO
3 - ddLSDot	A linha do objeto fica pontilhada.
4 - ddLSDashDot	A linha do objeto fica tracejada.
5 - ddLSDashDotDot	A linha do objeto fica pontilhada, tracejada e pontilhada.

O valor padrão desta propriedade é **1 - ddLSSolid**.

22.6.3.1.3 LineWeight

9 Especifica a largura da linha do objeto. Configurando a largura para 1, a linha do objeto é expandida em até 15 pontos, se for 2, a linha é expandida em até 30 pontos e assim por diante. Quanto maior o número configurado na propriedade, maior a largura do objeto. O valor padrão desta propriedade é 1.

22.6.3.1.4 Shape



Permite alterar a forma do objeto. As opções disponíveis são as seguintes:

Opções disponíveis para Shape

OPÇÃO	DESCRIÇÃO
0 - ddSHRectangle	Formato Retangular.
1 - ddSHELLipse	Formato Eliptico ou Circular.
2 - ddSHRoundRect	Formato de Retângulo Arredondado.

22.6.4 Figura

Esta seção contém informações sobre propriedades do objeto Figura do Relatório. Este objeto não possui eventos nem métodos associados.

22.6.4.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Figura do Relatório.

22.6.4.1.1 DataField

A Configura ou retorna o dado associado ao objeto. Este dado associado pode ser um campo da tabela no Banco de Dados fornecido pelo objeto Consulta; uma expressão matemática com campos da Consulta e funções do VBScript (neste caso, o campo deve ser precedido do símbolo de igual); ou um tag ou propriedade do E3. Neste caso o valor atual da variável será mostrado ao realizar a impressão. O valor padrão desta propriedade é vazio.

NOTA: O servidor deve estar sendo executado para que o valor da variável possa ser capturado.

22.6.4.1.2 Forecolor

 Especifica a cor do preenchimento do fundo do objeto. Nos scripts, utilize a função do VBScript **RGB** para formar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

22.6.4.1.3 hyperLink

A Especifica o *link* atribuído ao texto. Para utilizar este recurso, use o evento **OnHyperLink**. O valor padrão desta propriedade é vazio.

22.6.4.1.4 LineColor

 Especifica a cor da linha do objeto. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

22.6.4.1.5 LineStyle

 Determina o estilo da linha do objeto. As opções disponíveis desta propriedade são as seguintes:

Opções disponíveis para LineStyle

OPÇÃO	DESCRIÇÃO
0 - ddLSTransparent	A linha fica transparente no objeto.
1 - ddLSSolid	A linha aparece sólida no objeto.
2 - ddLSDash	A linha fica tracejada no objeto.
3 - ddLSDot	A linha fica pontilhada no objeto.
4 - ddLSDashDot	A linha fica tracejada e pontilhada no objeto.
5 - ddLSDashDotDot	A linha fica pontilhada, tracejada e pontilhada no objeto.

O valor padrão desta propriedade é **1 - ddLSSolid**.

22.6.4.1.6 LineWeight

9 Especifica a largura da linha do objeto. Configurando a largura para 1, a linha do objeto é expandida em até 15 ponto, e assim por diante. Quanto maior o número configurado na propriedade, maior a largura do objeto. O valor padrão desta propriedade é 1.

22.6.4.1.7 Picture

A Especifica o arquivo de figura para o objeto. As extensões permitidas são .bmp, .gif, .jpg, .cur, .ico, .emf e .wmf. O valor padrão desta propriedade é vazio.

22.6.4.1.8 PictureAlignment

 Determina o alinhamento da figura no objeto. As opções disponíveis são as seguintes:

Opções disponíveis para PictureAlignment

OPÇÃO	DESCRIÇÃO
0 - ddPATopLeft	Alinha a figura no topo esquerdo do objeto.
1 - ddPATopRight	Alinha a figura no topo direito do objeto.
2 - ddPACenter	Alinha a figura no centro do objeto.
3 - ddPABottomLeft	Alinha a figura à esquerda do objeto.
4 - ddPABottomRight	Alinha a figura à direita do objeto.

O valor padrão desta propriedade é **2 - ddPACenter**.

22.6.4.1.9 SizeMode

 Especifica o tamanho do objeto. As opções disponíveis são as seguintes:

Opções disponíveis para SizeMode

OPÇÃO	DESCRIÇÃO
0 - ddsMClip	Mostra o objeto no seu tamanho atual.
1 - ddsMStretch	Ajusta o objeto de acordo com a sua área.
2 - ddsMZoom	Ajusta a altura ou a largura da imagem do objeto dentro da área especificada, sem distorcê-la.

22.6.5 SetPoint

Esta seção contém informações sobre propriedades do objeto SetPoint do Relatório. Este objeto não possui eventos nem métodos associados.

22.6.5.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto SetPoint do Relatório.

22.6.5.1.1 Alignment

 Determina o alinhamento do texto no objeto. As opções disponíveis para esta propriedade são:

- **0 - Left:** alinhamento à esquerda (padrão).
- **1 - Right:** alinhamento à direita.
- **2 - Center:** alinhamento centralizado.

22.6.5.1.2 CanGrow

Determina a aplicação do efeito de *stretch* ao texto do objeto. Caso a largura ou altura do objeto sejam aumentadas, o texto acompanha esta variação. Se a propriedade estiver configurada para True, o texto acompanha as variações de altura e largura do objeto. Caso contrário, este permanece com as configurações iniciais. O valor padrão desta propriedade é True.

22.6.5.1.3 CanShrink

Determina a aplicação do efeito de encolhimento ao texto do objeto. Caso a largura ou altura do objeto sejam diminuídas, o texto acompanha esta variação. Se a propriedade estiver configurada para True, o texto acompanha as variações de altura e largura do objeto. Caso contrário, este permanece com as configurações iniciais. O valor padrão desta propriedade é True.

22.6.5.1.4 ClassName

A Retorna a classe do objeto. Esta propriedade é somente de leitura.

22.6.5.1.5 DataField

A Configura ou retorna o dado associado ao objeto. Este dado associado pode ser um campo da tabela no Banco de Dados fornecido pelo objeto Consulta; uma expressão matemática com campos da Consulta e funções do VBScript (neste caso, o campo deve ser precedido do símbolo de igual); ou um tag ou propriedade do E3. Neste caso o valor atual da variável será mostrado ao realizar a impressão. O valor padrão desta propriedade é vazio.

NOTA: O servidor deve estar sendo executado para que o valor da variável possa ser capturado.

22.6.5.1.6 Font

A Esta propriedade determina a fonte do texto no objeto. O valor padrão desta propriedade é vazio. Esta propriedade não pode ser utilizada em scripts ou em associações e é configurada somente via Studio.

22.6.5.1.7 ForeColor

 A propriedade **ForeColor** especifica a cor de fundo do objeto. Nos scripts, utilize a função do VBScript **RGB** para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

22.6.5.1.8 hyperLink

A A propriedade **hyperLink** determina o *link* que será atribuído ao texto. Para utilizar este recurso, use o evento **OnHyperLink**.

22.6.5.1.9 Multiline

A propriedade **Multiline** indica se o texto terá múltiplas linhas (True) ou se será uma caixa de texto simples (False). Isto pode ser visualizado quando o objeto Viewer está em execução. O valor padrão desta propriedade é False.

22.6.5.1.10 OutputFormat

A Configura ou retorna a formatação do texto da propriedade **Text**, utilizado nas configurações da função **Format** do Visual Basic.

22.6.5.1.11 Style

A Retorna o estilo do texto configurado no objeto. Esta propriedade é somente de leitura.

22.6.5.1.12 SummaryDistinctField

A Determina o nome do campo que será usado pela função selecionada na propriedade **SummaryFunc**. Esta propriedade só é válida se a função definida em **SummaryFunc** for do grupo de funções **Distinct Summary**, o qual abrange as funções de números 9 até 15, e quando a propriedade **SummaryType** tem valor diferente de 0.

22.6.5.1.13 SummaryFunc

 Determina o tipo de função que será usada para processar os valores do campo especificado na propriedade **DataField**, como listado na tabela a seguir. Esta propriedade só é válida quando a propriedade **SummaryType** tem valor diferente de 0.

Opções disponíveis para SummaryFunc

OPÇÃO	DESCRIÇÃO
0 - Sum	Calcula a soma de todos os valores dentro do intervalo de totais especificado (grupo, página ou relatório).
1 - Avg	Calcula a média de todos os valores dentro do intervalo de totais especificado (grupo, página ou relatório).
2 - Count	Conta o número de valores dentro do intervalo de totais especificado (grupo, página ou relatório).
3 - Min	Mostra o menor valor (valor mínimo) dentro do intervalo de totais especificado (grupo, página ou relatório).
4 - Max	Mostra o maior valor (valor máximo) dentro do intervalo de totais especificado (grupo, página ou relatório).

OPÇÃO	DESCRIÇÃO
5 - Var	Calcula a variância dos valores dentro do intervalo de totais especificado (grupo, página ou relatório).
6 - VarP	Calcula a variância populacional dos valores dentro do intervalo de totais especificado (grupo, página ou relatório).
7 - Dev	Calcula o desvio padrão dos valores dentro do intervalo de totais especificado (grupo, página ou relatório).
8 - DevP	Calcula o desvio padrão populacional dos valores dentro do intervalo de totais especificado (grupo, página ou relatório).
9 - DSum	Calcula a soma de todos os valores distintos dentro do intervalo de totais especificado (grupo, página ou relatório).
10 - DAvg	Calcula a média baseada nos valores distintos dentro do intervalo de totais especificado (grupo, página ou relatório).
11 - DCount	Conta o número dos valores distintos dentro do intervalo de totais especificado (grupo, página ou relatório).
12 - DVar	Calcula a variância dos valores distintos dentro do intervalo de totais especificado (grupo, página ou relatório).
13 - DVarP	Calcula a variância populacional dos valores distintos dentro do intervalo de totais especificado (grupo, página ou relatório).
14 - DDev	Calcula o desvio padrão dos valores dentro do intervalo de totais especificado (grupo, página ou relatório).
15 - DDevP	Calcula o desvio padrão populacional dos valores distintos dentro do intervalo de totais especificado (grupo, página ou relatório).

22.6.5.1.14 SummaryGroup

A Esta propriedade só é válida quando a propriedade **SummaryType** for igual a **3 - SubTotal**. **SummaryGroup** indica qual o nome da seção **GroupHeader** que será usada para controlar os subtotais, ou seja, a cada mudança de valor no **GroupHeader**, o somatório é reiniciado.

NOTA: Quando esta propriedade é utilizada, as propriedades **CanShrink** e **CanGrow**

ficam desabilitadas.

22.6.5.1.15 SummaryRunning

 Determina se será feito o total de somatórios acumulados, de acordo com as opções:

- **0 - None:** não faz o total.
- **1 - Group:** calcula os totais acumulados para cada intervalo de totais especificado.
- **2 - All:** calcula o total acumulado para todos os valores do relatório, independente do agrupamento.

Esta propriedade só é válida quando a propriedade **SummaryType** tem valor diferente de 0.

22.6.5.1.16 SummaryType

 Determina o tipo ou nível de totalização a ser gerada. As opções disponíveis são as seguintes:

Opções disponíveis para SummaryType

OPÇÃO	DESCRIÇÃO
0 - None	Sem geração de somatório.
1 - GrandTotal	Especifica que será totalizado todo o conteúdo do relatório.
2 - PageTotal	Especifica que será gerado um subtotal por página.
3 - SubTotal	Especifica que será gerado um subtotal para cada grupo, definido na propriedade SummaryGroup .
4 - PageCount	Especifica o contador de página.

22.6.5.1.17 Text

A Determina o texto que será atribuído ao objeto.

22.6.5.1.18 VerticalAlignment

 Determina o alinhamento vertical do objeto, a saber:

- **0 - Top**: alinhamento pelo topo.
- **1 - Middle**: alinhamento ao centro.
- **2 - Bottom**: alinhamento pela base.

22.6.5.1.19 WordWrap

Habilita ou desabilita uma quebra de linha no texto, caso a área disponível para o texto ultrapasse os limites determinados no objeto. Para que esta propriedade funcione, a propriedade **Multiline** deve ser obrigatoriamente igual a True. Se for False, a configuração `white-space: nowrap` irá aparecer na propriedade **Style**.

22.6.6 Texto

Esta seção contém informações sobre propriedades do objeto Texto do Relatório. Este objeto não possui eventos nem métodos associados.

22.6.6.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Texto do Relatório.

22.6.6.1.1 Alignment

 A propriedade **Alignment** determina o alinhamento do texto, a saber:

- **0 - ddtxLeft**: alinhamento à esquerda.
- **1 - ddtxRight**: alinhamento à direita.
- **2 - ddtxCenter**: alinhamento centralizado (valor padrão).

22.6.6.1.2 Angle

9 A propriedade **Angle** indica o ângulo do texto. O valor da propriedade deve ser especificado em décimos de graus, ou seja, para que o texto seja mostrado em um ângulo de 45 graus, o valor deve ser igual à 450. O valor padrão desta propriedade é 0 (posicionamento horizontal).

22.6.6.1.3 Caption

A A propriedade **Caption** contém o texto do objeto propriamente dito. O valor padrão desta propriedade é vazio.

22.6.6.1.4 ClassName

A A propriedade **ClassName** permite especificar uma classe CSS global (indicada em uma folha de estilos CSS externa) a ser aplicada ao texto. Uma classe CSS é um padrão de formatação que determina tipo e tamanho de letra ou alinhamento e espaçamento do parágrafo, entre outras características. Através de CSS, pode-se aplicar um padrão de formatação pré-definido a um texto, agilizando e unificando a apresentação dos textos. Para aplicar um estilo específico pode-se utilizar a propriedade **Style**. O valor padrão dessa propriedade é o estilo Normal.

22.6.6.1.5 Font

A Esta propriedade indica o nome da fonte (tipo das letras) do texto. O valor padrão desta propriedade é vazio (o E3 irá usar o padrão do sistema).

NOTA: Esta propriedade não pode ser utilizada em scripts ou associações, é configurada somente via Studio e só pode ser modificada em tempo de configuração.

22.6.6.1.6 ForeColor

■ A propriedade **Forecolor** especifica a cor de preenchimento do fundo do objeto. Nos scripts, utilize a função **RGB** do VBScript para montar a cor a ser associada a esta propriedade. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

22.6.6.1.7 hyperLink

A A propriedade **hyperLink** determina um *link* que será atribuído ao objeto. Para utilizar este recurso, use o evento **OnHyperLink**. O valor padrão desta propriedade é vazio.

22.6.6.1.8 Multiline

A propriedade **Multiline** indica se o texto terá múltiplas linhas (True) ou se será uma caixa de texto simples (False). Isto pode ser visualizado quando o objeto Viewer está em execução. O valor padrão desta propriedade é False.

22.6.6.1.9 Style

A A propriedade **Style** permite especificar um estilo CSS para o texto, em substituição ao estilo global. O valor da propriedade deve ser um **String** CSS válido, caso contrário a propriedade será ignorada. O valor padrão desta propriedade é vazio (o E3 irá usar o padrão do sistema). Exemplo:

```
Sub Report1_OnBeforePrint
  Label1.Style = "font-family: Times; font-weight: bold;_
  text-align: center; color: RGB(255, 255, 0)"
End Sub
```

22.6.6.1.10 VerticalAlignment

 A propriedade **VerticalAlignment** determina o alinhamento vertical do texto, a saber.

- **0 - ddTxTop**: alinhamento pelo topo (padrão).
- **1 - ddTxMiddle**: alinhamento centralizado.
- **2 - ddTxBottom**: alinhamento pela base.

22.6.6.1.11 WordWrap

Habilita ou desabilita uma quebra de linha no texto, caso a área disponível para o texto ultrapasse os limites determinados no objeto. Para que esta propriedade funcione, a propriedade **Multiline** deve ser obrigatoriamente igual a True.

22.6.7 Linha

Esta seção contém informações sobre propriedades do objeto Linha do Relatório. Este objeto não possui eventos nem métodos associados.

22.6.7.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Linha do Relatório.

22.6.7.1.1 LineColor

 A propriedade **LineColor** especifica a cor da linha do objeto. O valor padrão desta propriedade é preto (**RGB(0, 0, 0)**).

22.6.7.1.2 LineStyle

 Esta propriedade determina o estilo da linha do objeto. O valor padrão desta propriedade é **1 - ddLSSolid**. As outras opções disponíveis desta propriedade são as seguintes:

Opções disponíveis para LineStyle

OPÇÃO	DESCRIÇÃO
0 - ddLSTransparent	A linha fica transparente no objeto.
1 - ddLSSolid	A linha aparece sólida no objeto.
2 - ddLSDash	A linha fica tracejada no objeto.
3 - ddLSDot	A linha fica pontilhada no objeto.
4 - ddLSDashDot	A linha fica tracejada e pontilhada no objeto.
5 - ddLSDashDotDot	A linha fica pontilhada, tracejada e pontilhada no objeto.

22.6.7.1.3 LineWeight

9 A propriedade **LineWeight** especifica a largura da linha do objeto. Configurando a largura para 1, a linha do objeto é expandida em até 15 pontos, se a largura for 2, a linha do objeto é expandida em até 30 pontos e assim por diante. Quanto maior o número configurado na propriedade, maior a largura do objeto. O valor padrão desta propriedade é 1.

22.6.7.1.4 X1

9 A propriedade **X1** habilita ou desabilita a posição do ponto inicial da linha no eixo X.

22.6.7.1.5 X2

9 A propriedade **X2** determina a posição do ponto final da linha no eixo X. O valor padrão desta propriedade é vazio.

22.6.7.1.6 Y1

9 A propriedade **Y1** determina a posição do ponto inicial da linha no eixo Y. O valor padrão desta propriedade é vazio.

22.6.7.1.7 Y2

9 A propriedade **Y2** determina a posição do ponto final da linha no eixo Y. O valor padrão desta propriedade é vazio.

22.6.8 Quebra de Página

Esta seção contém informações sobre propriedades do objeto Quebra de Página do Relatório. Este objeto não possui eventos nem métodos associados.

22.6.8.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Quebra de Página do Relatório.

22.6.8.1.1 Enabled

☑ A propriedade **Enabled** habilita ou desabilita o objeto no relatório. Se a opção estiver configurada para True, o objeto é habilitado no relatório. Caso contrário, o objeto fica desabilitado. O valor padrão desta propriedade é True.

22.6.9 Tabela

Esta seção contém informações sobre propriedades do objeto Tabela do Relatório. Este objeto não possui eventos nem métodos associados.

22.6.9.1 Propriedades

Esta seção contém informações sobre as propriedades do objeto Tabela do Relatório.

22.6.9.1.1 CanGrow

☑ Determina a aplicação do efeito de *stretch* ao texto do objeto. Caso a largura ou altura do objeto sejam aumentadas, o texto acompanha esta variação. Se a propriedade estiver configurada para True, o texto acompanha as variações de altura e largura do objeto. Caso contrário, este permanece com as configurações iniciais. O valor padrão desta propriedade é True.

22.6.9.1.2 CanShrink

☑ Determina a aplicação do efeito de encolhimento ao texto do objeto. Caso a largura ou altura do objeto sejam diminuídas, o texto acompanha esta variação. Se a propriedade estiver configurada para True, o texto acompanha as variações de altura e largura do objeto. Caso contrário, este permanece com as configurações iniciais. O valor padrão desta propriedade é True.

22.6.9.1.3 CloseBorder

☑ Esta propriedade habilita ou desabilita a visualização da linha da borda da base da tabela, caso a mesma ultrapasse mais de uma página no relatório.

22.6.9.1.4 left

9 A propriedade **left** retorna o valor do posicionamento esquerdo do objeto no relatório. O valor padrão desta propriedade é vazio.

22.6.10 E3Chart

Esta seção contém informações sobre propriedades do objeto E3Chart do Relatório.

22.6.10.1 Propriedades

Exemplo de utilização das propriedades do E3Chart em um relatório

Os scripts abaixo devem ser criados na seção **PageHeader** do relatório, utilizando o evento **OnBeforePrint**. Exemplos:

```
Sub OnBeforePrint
    ' Usando um E3Chart em um relatório
    Set chart = _
        Report.Sections("PageHeader").Controls("E3Chart1")
    chart.LoadData()
    chart.FitAll
End Sub

Sub OnBeforePrint
    ' Este script faz a cópia das configurações do objeto
    ' E3Chart chartfrom para o E3Chart chart
    ' que será utilizado no relatório.
    Set chartfrom = _
        Application.GetFrame().Screen.Item("E3Chart1")
    Set chart = _
        Report.Sections("PageHeader").Controls("E3Chart2")
    chart.CopyConfig(chartfrom)
    chart.LoadData()
    chart.FitAll
End Sub
```

NOTA: A descrição das propriedades do E3Chart no relatório são as mesmas explicadas no tópico **Propriedades** do E3Chart.

CAPÍTULO

23

Dúvidas Mais Frequentes

Como fazer para que uma Tela janelada mostre a barra de título com botões de minimizar, maximizar e fechar?

Para isto, deve-se utilizar o método **SetFrameOptions** do Divisor. O parâmetro *lflags* especifica características da janela. O valor 127 define uma janela com os botões **Minimizar**, **Maximizar** e **Fechar** visíveis.

Como abrir uma Tela modal?

Para abrir uma Tela modal, utilize o método **DoModal** do Viewer. Por exemplo, **Application.DoModal "Tela1", "Título1", 0, 0, 400, 200, 0, 1** abre uma Tela de nome **Tela1**, com o título **Título1**, na posição 0, 0, com 400 pixels de largura e 200 pixels de altura, passa o valor 0 como parâmetro para a Tela e habilita a barra de título da janela.

Como faço para copiar valores de uma linha do E3Browser para um tag?

Primeiro, selecione a linha (ou o registro) desejada no E3Browser. Depois, utilize o método **GetColumnValue** do E3Browser. O parâmetro *IndexCampo* é o índice da coluna a ser copiada (iniciando em 0).

Como impedir que o usuário digite um string num SetPoint?

Verifique se o valor digitado é numérico no evento **Validate**. Por exemplo:

```
Sub Text1_Validate(Cancel, NewValue)
    If NOT IsNumeric(NewValue) Then
        MsgBox "The value must be numeric."
        Cancel = True
    End If
End Sub
```

Como abrir um calendário para selecionar uma data e hora ao clicar num SetPoint?

Usando o método **ShowDatePicker** no evento **Click** do SetPoint. Por exemplo:

```
Sub Text1_Click()  
    Dim datevalue  
    If Application.ShowDatePicker(datevalue) Then  
        Value = datevalue  
    End If  
End Sub
```

Como reconhecer todos os alarmes de uma área?

Para reconhecer todos os alarmes de uma área por script, é possível utilizar o método **Application.GetObject("ServidorAlarmes1").AckArea(*area*, *user*)**, onde:

- *Area* é o nome da área de alarmes da qual se quer reconhecer o alarme.
- *User* é o nome do operador logado, que pode ser o item Application.User.

Para reconhecer todos os alarmes ativos, é possível utilizar o método **Application.GetObject("ServidorAlarmes1").AckAllAlarms(*user*)**.

Como executar uma ação ao clicar numa tecla específica do mouse ou do teclado?

Usando os eventos **KeyDown** ou **KeyUp** da tela. Estes eventos são disparados quando uma tecla é pressionada ou solta e retornam dois parâmetros. Um é o código em ASCII da tecla que foi pressionada, o outro indica a condição das teclas SHIFT e CTRL no momento em que a tecla foi pressionada. A ideia é comparar o parâmetro de retorno do evento com o código em ASCII do caractere esperado.

Como criar um script WhileRunning?

Criando um evento associado a alguma propriedade que tenha sempre o mesmo valor. Por exemplo, a propriedade **Visible** de um objeto de tela. Enquanto o objeto estiver visível (**Visible** igual a True), o script será executado. Porém é recomendado evitar o uso de scripts **WhileRunning**, pois estes podem prejudicar a performance da aplicação. Na maioria dos casos, eles podem ser substituídos por associações.

Como criar um script OnValueChanged?

Criando um evento associado à propriedade **Value** do tag, que seja executado quando a propriedade alterar o seu valor.

NOTA: Atenção para não usar métodos do Viewer no Server, por exemplo um **MsgBox**. Se for o caso, o evento pode ser criado na tela ou até no próprio objeto Viewer ao invés de ser criado no tag.

Como criar tags e objetos de tela durante a execução?

Usando o método **AddObject**. Por exemplo, o script a seguir cria tags de comunicação no driver `Driver1`.

```
Set obj = Application.GetObject("Driver1")
For i = 1 To 100
    Set tag = obj.AddObject("IOTag", false)
    tag.Name = "IOTag" & CStr(i)
    tag.Activate
Next
```

Como mostrar uma mensagem na tela ao alterar o valor de um tag?

Criando um evento na tela associado à propriedade **Value** do tag, que seja executado quando a propriedade alterar o seu valor. Neste evento, utilize o método **MsgBox** para mostrar a mensagem.

Como fazer uma consulta com filtro por data antes de montar um relatório?

Para fazer isto, é necessário configurar o objeto Consulta (ver capítulo Consulta), que acompanha o relatório, e criar as variáveis necessárias na coluna **Filtro**. No evento de chamada de relatório, use um script semelhante ao seguinte:

```
Set relatorio = Application.LoadReport("[Relatório1]")
Set consulta = Relatorio.Query()
consulta.SetVariableValue "Variavel1", Valor1
consulta.SetVariableValue "Variavel2", Valor2
Relatorio.PrintPreview()
```

Onde:

- **[Relatório1]** é o nome do relatório a ser chamado.
- **Variavel1**, **Variavel2** são as variáveis criadas no filtro do campo **E3TimeStamp**.
- **Valor1**, **Valor2** são as datas a serem consultadas.

Para verificar outros tipos de filtros, consulte o capítulo **Consultas** ou a documentação disponível no **Elipse Knowledgebase**.

Como depurar erros de scripts no servidor e no Viewer?

Caso o evento seja executado no Viewer, utilize o método **MsgBox**. Já se o evento for executado no servidor, utilize o método **Trace**.



Matriz

Rua 24 de Outubro, 353 - 10 andar
90510-002 Porto Alegre RS
Fone: (51) 3346-4699
Fax: (51) 3222-6226
E-mail: elipse@elipse.com.br

Filial SP

Rua dos Pinheiros, 870 - Conj. 141/142
05422-001 São Paulo - SP
Fone: (11) 3061-2828
Fax: (11) 3061-2828
E-mail: elipse-sp@elipse.com.br

USA

40190 Jarvis Gray Lane
Avon - NC - USA 27915
Fone: (252) 995-6885
Fax: (252) 995-5686
E-mail: info@elipse-software.com

Alemanha

D-67714 Waldfischbach Deutschland
Fone: +49 (0) 6333 4439
Fax: +49 (0) 6333 2790045
E-mail: elipse@elipse.de

Filial MG

Av. do Contorno, 6594- 17º andar Sala 01
30110-044 Belo Horizonte - MG
Fone: (31) 3555-3366
Fax: (31) 3555-3399
E-mail: elipse-mg@elipse.com.br

Filial PR

Av. Sete de Setembro, 4698/1705
80240-000 Curitiba - PR
Fone: (41) 3342-0120
Fax: (41) 3342-0120
E-mail: elipse-pr@elipse.com.br

Taiwan

15F.-1, No.56, Minsheng 1 st Rd.
800 Kaohsiung City Taiwan
Fone: +886-7-2221582
Fax: +886-7-2225909
E-mail: evan@elipse.com.br

Consulte nosso website para informações
sobre o representante do seu estado.

www.elipse.com.br
elipse@elipse.com.br

