

# ÍNDICE

<b>1. BANCO DE DADOS.....</b>	<b>5</b>
1.1. CONCEITO DE BANCO DE DADOS .....	5
1.2. SISTEMA DE COMPUTAÇÃO .....	6
<i>Memória de longa duração.....</i>	<i>6</i>
<i>Gerência de Banco de Dados.....</i>	<i>7</i>
1.3. SISTEMA GERENCIADOR DE BANCO DE DADOS (SGBD) .....	7
<i>Gerência e gerenciadores .....</i>	<i>7</i>
<i>Necessidade dos dados.....</i>	<i>7</i>
<i>Abordagem de Sistêmica.....</i>	<i>9</i>
<i>Abordagem Física.....</i>	<i>9</i>
<i>Tamanho .....</i>	<i>10</i>
<i>Entidade.....</i>	<i>10</i>
<i>Tamanho .....</i>	<i>10</i>
<i>Registro.....</i>	<i>10</i>
<i>Elaboração de modelo de dados.....</i>	<i>11</i>
<i>Integridade de dados.....</i>	<i>11</i>
<i>Privacidade e segurança.....</i>	<i>11</i>
<i>Agilidade e flexibilidade na obtenção de informações .....</i>	<i>11</i>
<i>Padronização .....</i>	<i>12</i>
1.4. OBJETOS .....	12
<i>O que é objeto .....</i>	<i>12</i>
<i>Exemplo de objeto e suas instâncias.....</i>	<i>12</i>
<i>Algumas regras sobre objetos.....</i>	<i>13</i>
<i>Sobre a natureza dos objetos .....</i>	<i>13</i>
1.4. ATIVIDADES .....	14
<b>2. MODELAGEM DE DADOS.....</b>	<b>21</b>
2.1. TABELA.....	21
2.1.1. <i>Definição.....</i>	<i>21</i>
2.1.2. <i>Propriedades.....</i>	<i>22</i>
2.2. DOMÍNIO.....	23
2.2.1. <i>Definição.....</i>	<i>23</i>
2.2.2. <i>Regra.....</i>	<i>23</i>
2.2.3. <i>Exposição.....</i>	<i>24</i>
2.2.4. <i>Diretriz.....</i>	<i>24</i>
2.3. CONVENÇÕES E NOMENCLATURA .....	25
2.3.1. <i>Regras .....</i>	<i>25</i>
2.3.2. <i>Diretrizes.....</i>	<i>25</i>
2.3.3. <i>Definições .....</i>	<i>25</i>
2.4. REGRAS ADICIONAIS .....	27
2.4.1. <i>Regra 1.....</i>	<i>27</i>
2.4.2. <i>Regra 2.....</i>	<i>27</i>
2.4.3. <i>Regra 3.....</i>	<i>28</i>
2.4.4. <i>Regra 4.....</i>	<i>28</i>
2.5. VALORES DUPLICADOS .....	29
2.5.1. <i>Regra.....</i>	<i>29</i>
2.6. VALORES NULOS .....	30
2.6.1. <i>Regra.....</i>	<i>30</i>
2.7. CHAVE PRIMÁRIA .....	31
2.7.1. <i>Regra 1.....</i>	<i>31</i>
2.7.2. <i>Regra 2.....</i>	<i>31</i>

2.7.3. Regra 3.....	31
2.7.4. Regra 4.....	31
2.7.5. Diretrizes.....	31
2.7.6. Regra 5.....	32
2.8. CHAVE ESTRANGEIRA.....	33
2.8.1. Regra 1.....	33
2.8.2. Regra 2.....	35
2.9. LER TABELAS.....	36
2.9.1. Exposição 1.....	36
2.9.2. Exposição 2.....	38
2.9.3. Exposição 3.....	38
2.9.4. Exposição 4.....	38
2.10. REVISÃO.....	39
2.10.1 Definição de Tabela.....	39
2.10.2. Definição de Domínio.....	39
2.10.3. Definições para Colunas.....	39
2.10.4. Definição de Valor Duplicado.....	39
2.10.5. Definição de Valor Nulo.....	39
2.10.6. Definição de Chave Primária.....	39
2.10.7. Definição de Chave Estrangeira.....	40
2.10.8. Comentário Geral.....	40
2.11. MODELO DE DADOS.....	41
2.11.1. Definição de Modelo de Dados.....	41
2.11.2. Construções.....	41
2.11.3. Operadores.....	41
2.11.4. Regras de Integridade.....	41
2.11.5. Reflexão sobre Tabelas e Entidades.....	41
2.12. ENTIDADES.....	42
2.12.1. Definição.....	42
2.12.2. Exposição.....	42
2.13. ASSOCIAÇÕES.....	43
2.13.1. Definição.....	43
2.13.2. Exposição.....	43
2.13.3. Associações “um-para-um”.....	44
2.13.4. Associações “um-para-muitos”.....	45
2.13.5. Associações “muitos-para-muitos”.....	46
2.14. TABELAS ASSOCIATIVAS.....	47
2.14.1. Definição.....	47
2.14.2. Regra.....	47
2.14.3. Exposição.....	47
2.15. ATIVIDADES.....	49
<b>ACCESS 2 E 2000.....</b>	<b>52</b>
1. INTRODUÇÃO.....	52
1.1. Banco de dados.....	52
1.2. Campo.....	52
1.3. Registro.....	52
1.4. Índices.....	52
1.5. Chave.....	52
2. INICIANDO O ACCESS 2000.....	53
3. CRIANDO UM BANCO DE DADOS.....	54
4. TABELAS.....	55
5. NOME DO CAMPO.....	56
6. TIPO DE DADOS.....	56
6.1. Texto.....	56
6.2. Memorando.....	56
6.3. Número.....	56
6.4. Data/Hora.....	56
6.5. Moeda.....	56
6.6. AutoNumeração.....	57
6.7. Sim/Não.....	57

6.8. Objeto OLE.....	57
6.9. Hyperlink.....	57
6.10. Assistente de pesquisa.....	57
7. PROPRIEDADES DO CAMPO.....	57
7.1. Tamanho do campo.....	57
8. FORMATO.....	58
8.1. Texto e Memorando.....	58
9. MÁSCARA DE ENTRADA.....	59
9.1. Casas decimais.....	60
9.2. Legenda.....	60
9.3. Valor padrão.....	60
9.4. Texto de validação.....	61
9.5. Requerido.....	61
9.6. Permitir comprimento zero.....	61
9.7. Indexado.....	61
9.8. Compactação Unicode.....	61
10. CHAVE PRIMÁRIA.....	61
11. DEFININDO UMA CHAVE PRIMÁRIA.....	62
12. INSERINDO DADOS NA TABELA.....	62
13. EDITANDO A TABELA DE DADOS.....	63
14. CLASSIFICAÇÃO CRESCENTE DECRESCENTE.....	64
15. LOCALIZAR REGISTROS.....	64
16. EXCLUIR REGISTROS.....	65
17. EXCLUIR CAMPOS.....	65
18. INSERIR CAMPOS.....	66
19. CONSULTAS.....	66
20. CRIAR UMA CONSULTA.....	66
21. SELEÇÃO DOS CAMPOS.....	67
22. CLASSIFICANDO A CONSULTA.....	68
23. EXIBIÇÃO DE CAMPOS.....	68
24. SALVAR A CONSULTA.....	69
25. CRITÉRIOS DE PESQUISA.....	69
26. CONSULTANDO VÁRIAS TABELAS.....	70
27. SQL.....	71
28. DECLARAÇÕES SQL.....	71
<i>Instrução SQL</i> .....	71
29. RELACIONAMENTOS.....	72
30. FORMULÁRIOS.....	72
31. CRIANDO UM FORMULÁRIO.....	73
32. ESTRUTURA DO FORMULÁRIO.....	76
33. ALTERAR APARÊNCIA DO TEXTO.....	77
34. RÓTULO DO FORMULÁRIO.....	78
35. EXIBIR FORMULÁRIO.....	79
36. RELATÓRIOS.....	79
37. CRIANDO UM RELATÓRIO.....	79
38. ESTRUTURA DO RELATÓRIO.....	84
39. IMPRIMINDO O RELATÓRIO.....	86
40. MACROS.....	86
41. CRIAR UMA MACRO.....	87
42. CÓDIGO VISUAL BASIC NO MICROSOFT ACCESS.....	88
43. CRIAR CÓDIGO VISUAL BASIC.....	88
<b>1. NORMAS E PADRÕES.....</b>	<b>91</b>
1.1. O POR QUÊ.....	91
1.2. CONCEITO.....	91
1.3. OBJETIVOS.....	92
1.4. PREMISSAS FINAIS.....	92
<b>2. SIGLAS, ABREVIATURAS E REGRAS DEFINIDAS.....</b>	<b>93</b>
2.1. CÓDIGOS DOS SISTEMAS.....	93
2.2. NOMES DE TABELAS.....	93

2.3. NOMES DE COLUNAS.....	94
2.4. NOMES DE OBJETOS.....	95
<b>3. NORMAS PARA PRODUTOS.....</b>	<b>98</b>
3.1. ANÁLISE.....	98
3.2. PROGRAMAÇÃO .....	98
3.3. TELAS .....	101
3.4. MODELOS DE TELAS E PROGRAMAÇÃO.....	105
3.5. RELATÓRIOS .....	112
<b>ANEXO 1 - TABELA DE MNEMÔNICOS .....</b>	<b>113</b>
<b>ANEXO 2 – CÓDIGOS FONTES.....</b>	<b>114</b>
<i>Macro Autoexec do Sistema CAU – primeira a ser executada .....</i>	<i>114</i>
<i>Script (módulo) Anexa_Tabelas() do Sistema CAU .....</i>	<i>115</i>

# 1. Banco de Dados

## 1.1. Conceito De Banco De Dados

O completo entendimento e aceitação de uma terminologia, passa por um processo muito maior do que possam pensar alguns. Vamos tomar como exemplo o Sistema Métrico Decimal.

Uma série de estudos desenvolvidos por diversos cientistas teve de ser empreendido, para que a terminologia ganhasse uma definição, que por sua vez, seria estabelecida como padrão internacionalmente reconhecido e aceito, ainda que, sujeito a revisões periódicas que pudessem forçar sua modificação.

Um Banco de Dados é constituído de um conjunto de tabelas relacionadas entre si. Estes relacionamentos são regras implementadas através de funções específicas de cada SGBD, e que disponibilizam os dados a partir destas regras conforme eles são solicitados e as regras são aplicadas (Figura 4).

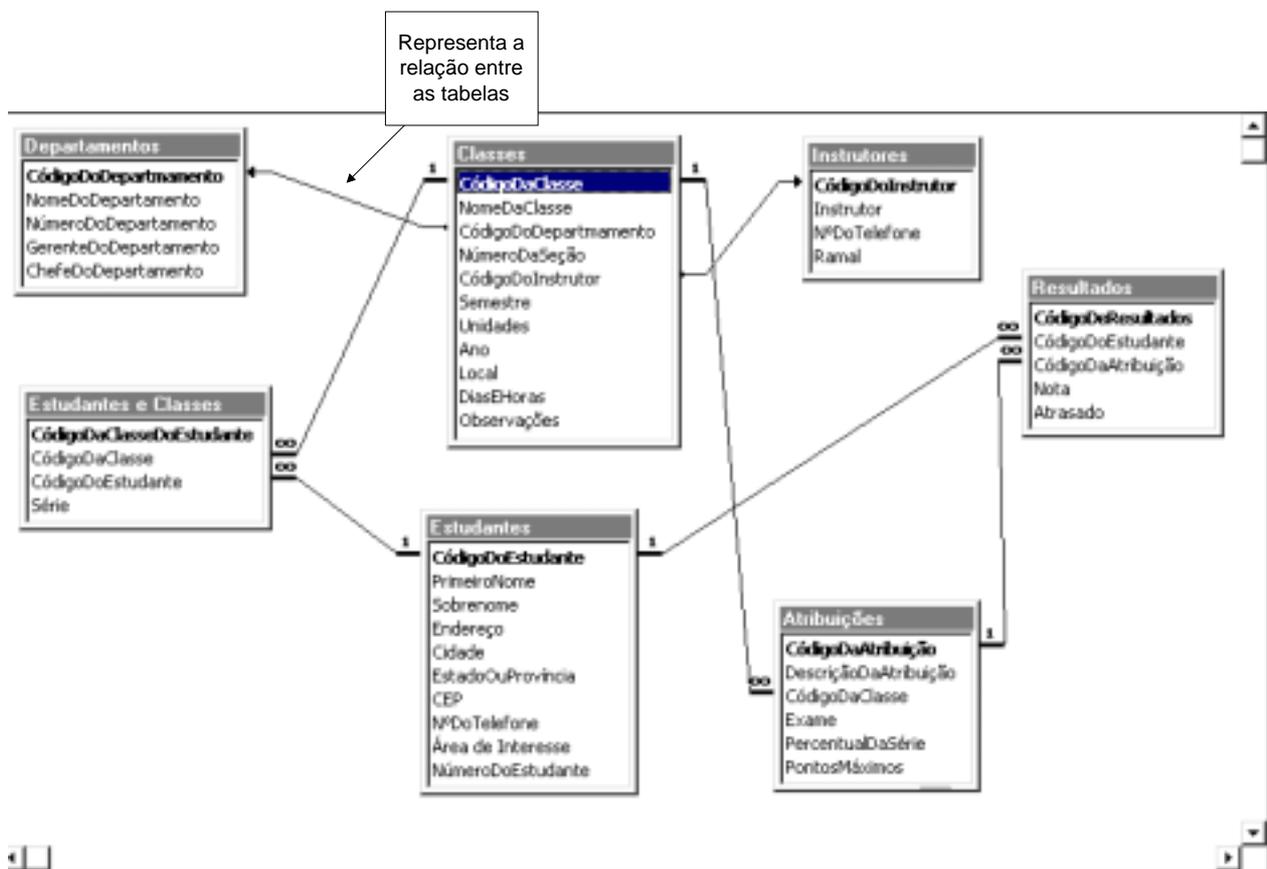


Figura 4

Examinemos agora alguns conceitos bastante aceitos sobre Banco de Dados e façamos sobre eles alguns comentários, esquemas e deduções.

## 1.2. Sistema de Computação

Na Figura 25, temos uma representação gráfica clássica de um Sistema de Computação, que é assinalada por cinco elementos básicos: (1) Entrada Padrão - Standard Input - por onde os dados são inseridos no (2) Processador Central - CPU - para que sejam processados. (3) Memória Volátil utilizada pela CPU para armazenamento temporário dos dados que serão processados e dos seus resultados antes de sua gravação na (4) Memória de Massa, que fará o armazenamento permanente dos dados processados. (5) Saída Padrão - Standard Output - por onde os dados processados serão exibidos.

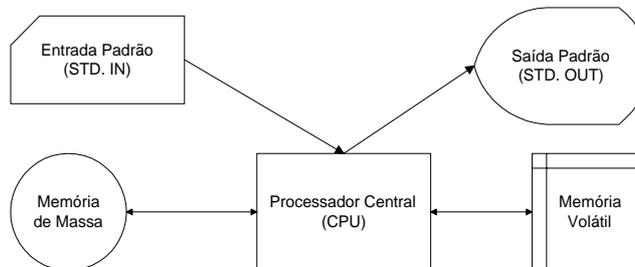


Figura 25

Para cada um destes 5 elementos temos uma referência real com equipamentos de informática do nosso dia-a-dia.

- (1) Entrada Padrão - Standard Input - Teclado
- (2) CPU - Processador Central
- (3) Memória Volátil - É a memória de trabalho do computador
- (4) Memória de Massa - Fitas, Hard Disk (winchester), Disquetes, CD-ROM.
- (5) Saída Padrão - Standard Output - Monitor de Vídeo ou Impressora.

### Memória de longa duração

São os elementos físicos representantes da Memória de Massa do Sistema de Computação, ilustrados na Figura 26 em algumas de suas formas mais comuns.

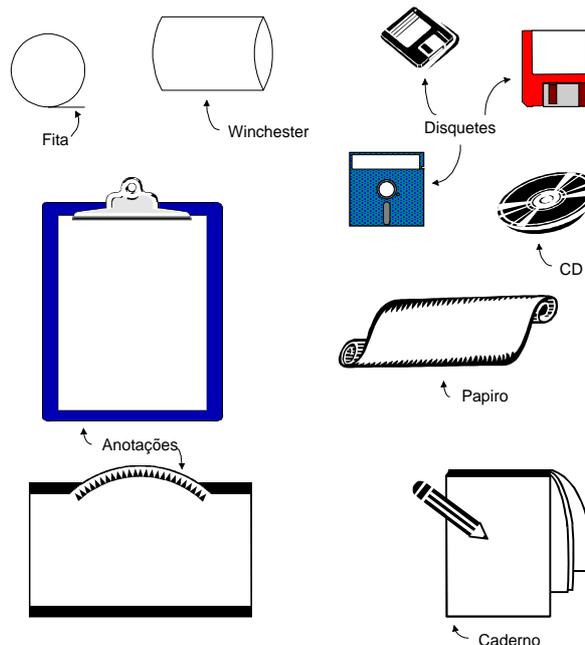


Figura 26

### Gerência de Banco de Dados

O significado da palavra gerência, claramente definido no dicionário em toda a sua abrangência, está inserido como termo técnico da Ciência da Informação. Vamos fazer uma extrapolação deste termo e dar a ele uma evidência clara e inequívoca de como é causador de reflexos fortes e decisivos na tecnologia de Banco de Dados.

### 1.3. Sistema gerenciador de banco de dados (SGBD)

Corresponde a um conjunto de programas para administrar o Banco de Dados. Estes programas preocupam-se em manter o acesso aos dados do banco de dados o mais simples possível. Geralmente um SGBD é baseado em um computador servidor de alta performance e alto grau de segurança.

Um sistema de Banco de Dados corresponde ao conjunto composto de pelo menos Banco de Dados e SGBD.

O ambiente de Banco de Dados pode ser considerado como composto pelos seguintes elementos:

- 1 – Dados de fonte
- 2 – SGBD
- 3 – Banco de Dados (armazenado)
- 4 – Administrador do Banco de Dados (e instrumentos)
- 5 – Modelo de dados
- 6 – Submodelos de dados
- 7 – Comunidade de usuários
- 8 – Normas e procedimentos de operação

### Gerência e gerenciadores

Quando um software é desenvolvido com o propósito de desempenhar a função de Gerenciador de Banco de Dados, o mínimo que se espera dele é que ele faça bem o seu papel, oferecendo ao desenvolvedor e usuários em geral ferramentas poderosas, eficientes, amigáveis e ergonômicas.

- **Poder de um SGBD**

O poder de uma ferramenta é determinado pela capacidade que ela oferece de executar tarefas das mais simples às mais complexas com segurança e simplicidade.

- **Eficiência de um SGBD**

A eficiência de uma ferramenta é determinada pela sua capacidade de desempenhar, e bem, o papel ao qual se propõe.

- **Amigabilidade de um SGBD**

A amigabilidade de uma ferramenta é determinada pela simplicidade e eficácia oferecidas em para sua operação.

- **Ergonomia de um SGBD**

A ergonomia de uma ferramenta é determinada pela facilidade que ela oferece em ser adaptada às necessidades, trejeitos e excentricidades de seu usuário.

### Necessidade dos dados

Nós vivemos a era da informação. A Terra pode até girar em torno do Sol, mas antes de mais nada o nosso mundo “gira” em torno da informação.

Não podemos colocar em dúvida que a coisa mais importante na era atual é a informação. O homem sabendo disto, criou ao longo dos anos mecanismos para torná-la mais acessível, mais pública, mais notória.

Assistimos diariamente instituições financeiras levando literalmente o acesso dos seus Bancos de Dados para dentro das empresas e dos lares de todo o mundo. Seja através de linhas telefônicas, telex, fax, terminais de computadores, Internet, o que importa é a criação de uma linha direta de acesso a informação.

Esta linha direta atende a necessidade primária do usuário/cliente. O acesso à informação.

É importante destacar que os termos informação e dado não devem ser confundidos. Para tanto vamos conceituá-los para que possamos utilizá-los e entendê-los da forma correta.

### Dado

Elemento conhecido e certo que se presta à solução de um problema qualquer como elemento componente da equação, ou em resposta de uma dúvida elementar. Por exemplo para a pergunta: Qual é sua idade? O dado que a responde é único e inequívoco, o número 30.

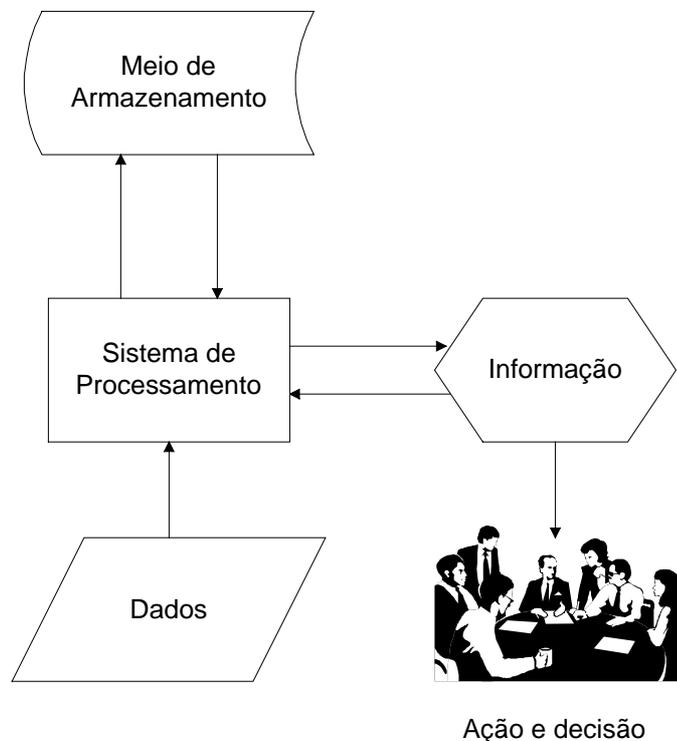
### Informação

Ato ou efeito de informar através de um conjunto de dados dispostos de maneira adequada à cada problema, dando então resposta ao problema, porém susceptível de interpretação e análise. Por exemplo para o problema: Quantos e quais são, em ordem alfabética, os funcionários que fazem aniversário no mês de outubro? A informação que responderá a este problema é mais complexa do que um simples número. Os dados que respondem este problema formam uma lista com diversos nomes ou mesmo nenhum (se não houver ninguém que faça aniversário no mês de outubro), e isto é uma informação.

### *Revisando - Dados e informação*

Dados e informação possuem diversas definições e interpretações, conforme cada autor. As considerações seguintes são colocadas segundo o interesse do Sistema de Informação, onde os Bancos de Dados assumem um papel de relevância.

Informação é dado que foi processado por um determinado sistema e obtida numa forma tal que tenha utilidade para o usuário receptor, seja para a execução de tarefas ou para a tomada de decisões (Figura 5).



**Figura 5**

### Abordagem de Sistêmica

Um conjunto de informações para um determinado usuário A pode não constituir informação para o usuário B, inclusive pode constituir um conjunto de dados para este último.

Entrada	Processamento	Saída	Visão do usuário pai do aluno	Visão do usuário professor
Notas dos alunos	Sistema de cálculo de médias	Boletim	Dar aquela moto de presente ou não	Aprovação, reprovação direta ou recuperação.

Se o sistema de processamento de dados envolve um sistema de computação eletrônica, normalmente os dados e informações obedecem a um conjunto bem definido de regras de tamanho e formato.

Neste caso, temos como entrada um conjunto de itens de dados e na saída um conjunto de itens de informação.

### Abordagem Física

As considerações de dados e informações aqui expostas são feitas com relação a um sistema de processamento de informação. No entanto, quando consideramos as informações como abstrações mentais de uma realidade qualquer, e desejamos efetuar um modelo representativo no meio de armazenamento do sistema computador, então usamos uma estrutura ou modelo de dados. Neste sentido, os dados são utilizados mais como meio de representação do que destinados a regras de transformação por sistemas de processamento.

Com o aumento da utilização de sistemas de banco de dados, muitas áreas correlatas em processamento de dados estão sofrendo revisões, com a finalidade de melhor aproveitar as vantagens desta nova tecnologia de software.

Uma das áreas que se vem transformando, refere-se à área de análise e projeto de sistemas com vistas à utilização de banco de dados.

Em contraste aos métodos tradicionais de implantação de sistemas estanques, onde predomina a produção de determinados relatórios através de um conjunto de programas, a análise e projeto de sistemas de informação com base em banco de dados tem demandado uma outra linha de análise. É de suma importância que seja bem analisado e projetado o seu núcleo de informações na forma de estrutura de dados, utilizável por diversas aplicações.

O projeto de uma estrutura ótima de dados impõe uma nova metodologia na análise das informações envolvidas. Assim, ao se projetar um Banco de Dados, o qual deverá conter as informações sobre uma determinada organização ou empreendimento, deve-se examinar as seguintes etapas:

1. Identificar as entidades mais relevantes ou representativas da organização. As entidades podem ser pessoa, lugar, coisa ou evento.
2. Identificar as propriedades mais notáveis ou de interesse das entidades escolhidas.
3. Estabelecer os atributos correspondentes às propriedades. Tais atributos consistem nas diversas classes de informação que caracterizam as propriedades das entidades. Neste plano de informação a análise é estritamente conceitual, formada na mente do analista.

A escolha de entidades e atributos não é tarefa fácil; a relevância das mesmas depende da pessoa que está analisando; diversos analistas podem denominar uma mesma entidade ou atributo por nomes diferentes, ou então diversas entidades ou diversos atributos com o mesmo nome. Determinar para cada entidade os atributos correspondentes e assinalar determinados atributos como sendo os *identificadores* da entidade, e os atributos que não são identificadores mas que *qualificam* a entidade. Por exemplo:

Atributo	Tamanho	Formato	Domínio	Tipo
Número de Matrícula	8	Numérico	-	Identificador
Nome	35	Alfabético	-	Qualificador
Sexo	1	Alfabético	M ou F	Qualificador
Data de Nascimento	8	Numérico	-	Qualificador
Estado Civil	1	Alfabético	S, C, V, D ou A	Qualificador
Endereço	45	Alfanumérico	-	Qualificador
Telefone Residencial	12	Numérico	-	Qualificador
Nome da Escola	45	Alfanumérico	-	Qualificador
Série	1	Numérico	1 até 8	Qualificador
Sala	2	Numérico	-	Qualificador
Turma	1	Alfabético	-	Qualificador
Turno	1	Alfabético	M, T ou N	Qualificador

## Entidade

Também chamada de *tabela* ou *arquivo lógico*, corresponde a um conjunto de *itens de dado* e constitui por si só a descrição de um objeto (geralmente do mundo real) que pode ser representada de forma abstrata.

Por exemplo, para descrever um aluno que estuda em uma escola precisamos dos seguintes itens de dado para compor a entidade ALUNO:

Atributo	Tamanho	Formato	Domínio
Número de Matrícula	8	Numérico	-
Nome	35	Alfabético	-
Sexo	1	Alfabético	M ou F
Data de Nascimento	8	Numérico	-
Estado Civil	1	Alfabético	S, C, V, D ou A
Endereço	45	Alfanumérico	-
Telefone Residencial	12	Numérico	-
Nome da Escola	45	Alfanumérico	-
Série	1	Numérico	1 até 8
Sala	2	Numérico	-
Turma	1	Alfabético	-
Turno	1	Alfabético	M, T ou N

## Registro

Também chamado de *record* ou *linha*, um registro consiste num conjunto de itens de dados que reunidos caracterizam uma ocorrência de informações de uma entidade.

Veja na Figura 2 um exemplo.

Considerando o conjunto temos um Registro

Código do Depto.	Departamento	Número do Departamento
1	Psicologia	120
2	Química	130
3	Matemática	140
4	Informática	150
5	Inglês	160

**Figura 2**

### Elaboração de modelo de dados

A complexidade de um Banco de Dados reside mais na sua elaboração do que em sua criação e manutenção. O processo que antecede a criação efetiva de um Banco de Dados, denomina-se modelagem de dados. É no processo da modelagem que o Analista de Sistemas, de posse de todos os levantamentos e entrevistas feitas com os usuários do sistema, define quais serão os dados necessários e como eles serão organizados em tabelas e ainda como estas tabelas serão relacionadas para atender as “necessidades do negócio”.

As “necessidades do negócio” são chamadas de Regras de Negócio. São estas regras que definem a forma pela qual os dados devem ser organizados dentro do Banco de Dados para que atendam o dia-a-dia do usuário.

### Integridade de dados

A integridade dos dados está ligada a sua confiabilidade total. Por exemplo: um Banco de Dados, cujas tabelas façam o registro de lançamentos de uma conta corrente dos clientes de um banco. Se o cliente fizer uma consulta de saldo ou solicitar um extrato de sua conta corrente, jamais poderá acontecer uma falha. A precisão deve ser de 100%. Para garantir esta integridade de dados, o SGBD deve possuir mecanismos de controle infalíveis. Imagine um depósito feito por um cliente sendo creditado para outro ou um cheque sendo devolvido por insuficiência de fundos quando na verdade tinha fundos.

### Privacidade e segurança

Tão importante quanto a integridade dos dados é a segurança que é preciso ter quanto ao acesso aos dados. A natureza das informações normalmente são sigilosas: saldo de conta corrente, investimentos, senhas de acesso, relação de clientes devedores e diversas outras. O acesso à informação só deve ser permitido àqueles que possuam sua autenticação através do nome de usuário de sua senha pessoal.

### Agilidade e flexibilidade na obtenção de informações

Não existe nada mais irritante para um usuário do que o tempo de espera para obter uma informação, não importa se no seu computador em casa, no caixa eletrônico ao consultar seu saldo ou pedir extrato, na Internet quando esta acessando aquele site interessante, os SGBDs de vem oferecer flexibilidade (permitir obter a informação de diversas formas) e agilidade permitir que as informações cheguem com rapidez).

### Padronização

Um SGBD deve oferecer uma forma padronizada de acesso aos dados, para que se ganhe independência de plataforma e desta forma agilidade para obtenção da informação. Esta padronização vai além das fronteiras do acesso e passa pela coerência que devem ter analistas, DBAs e programadores em geral ao projetar um Banco de Dados, para que ele não se transforme em uma amontoado de código que ninguém consiga entender ou consultar de maneira interativa.

Veja o capítulo Normas e Padrões para entender mais sobre esta necessidade.

## 1.4. Objetos

No universo das atividades empresariais e quaisquer outras atividades que venhamos a desempenhar ou empreender, tudo gira em torno de objetos que são utilizados corriqueiramente e dos quais, na maioria das vezes, não nos damos conta.

Por isso, e também para facilitar o entendimento de alguns elementos que serão utilizados mais adiante no estudo de Banco de Dados, veremos um conjunto de conceitos (novos para alguns, mas velhos na verdade), que devem ser do inteiro conhecimento de todos que querem entender objetos. O que faremos na verdade é simplesmente dar nomes e enunciar algumas regras que cercam o conceito de objetos.

A tecnologia da informação evoluiu muito e não podemos negligenciar a utilização destas novas técnicas, sejam para aprimorar o tratamento dos dados, sua organização ou forma de recuperação, independente do meio que seja utilizado para tal.

Para tanto o entendimento dos seus fundamentos e sua aplicabilidade serão agora discutidas, procurando iniciá-lo nesta forma de pensar a organização dos dados no dia-a-dia do seu trabalho seja ele qual for.

### O que é objeto

“Objeto é um conjunto de coisas tangíveis ou abstratas, que possuam as mesmas características e que estejam sujeitas às mesmas normas”.

As coisas tangíveis ou abstratas que podem compor um conjunto são chamadas de instâncias do objeto.

### Exemplo de objeto e suas instâncias

Objeto FERRAMENTAS DE ESCRITA MANUAL= {Lápis preto HB, Lápis de cor vermelho, Lápis de carpinteiro} formam um conjunto de coisas do mundo real. O Lápis preto HB é uma das instâncias do objeto FERRAMENTAS DE ESCRITA MANUAL.

Se montarmos uma tabela que represente o objeto FERRAMENTAS DE ESCRITA MANUAL e completarmos com as instâncias citadas teremos explícito na definição “...que possuam as mesmas características...” e teremos implícito que nenhum elemento da tabela pode estar em branco e que não podemos ter dados estranhos ao conjunto.

Com isto em mente a Tabela 1.1 é válida:

Tipo	Cor	Utilização mais comum
HB	preto	escrever
de cor	vermelho	pintar
de carpinteiro	preto	marcar madeira para corte

Tabela 1.1 Dados explícitos válidos para o objeto.

Já a Tabela 1.2 é inválida pois possui dados estranhos ao conjunto e também dados em branco:

Tipo	Cor	Utilização mais comum
HB	preto	escrever
Teclado	?	digitação
de carpinteiro	preto	marcar madeira para corte

**Tabela 1.2** Dados explícitos inválidos para o objeto.

Instância é o dado em si.

Instância do objeto EMPREGADO é a pessoa em si. É o João das Dores.

Instância do objeto CLIENTE é o cliente em si, digamos, o Banco do Brasil.

Instância do objeto PEÇA é a peça em si, digamos o Parafuso.

### Algumas regras sobre objetos

- Um objeto pode armazenar dados
- Um objeto pode armazenar operações
- Estas operações podem manipular os dados nele armazenados ou dados externos
- Um objeto pode ser composto de outros objetos
- Um objeto deve possuir um ciclo de vida definido
- Um objeto pode ser tangível (existir de fato), ou abstrato (ser fruto da imaginação)
- Um objeto deve possuir um nome que o identifique e diferencie perante os demais
- Uma mesma instância pode estar presente em n objetos
- Um objeto pode dar origem a outros objetos (gerar descendentes)
- Um objeto descendente de outro deve ter pelo menos o nome diferente
- Um objeto deve ter um objetivo definido antes de iniciar o seu ciclo de vida
- Um objeto deve ter pelo menos uma instância

### Sobre a natureza dos objetos

Objetos são tangíveis ou abstratos, e estão ligados às coisas que eles se referem ao nível de suas instâncias existirem de fato ou imaginárias (existem somente na imaginação do seu criador).

Quando pensarmos desta forma, estaremos orientando o nosso raciocínio ao que acreditamos existir, os objetos e suas instâncias, e a isto vamos chamar de Pensamento Orientado à Objetos (POO).

## 1.4. Atividades

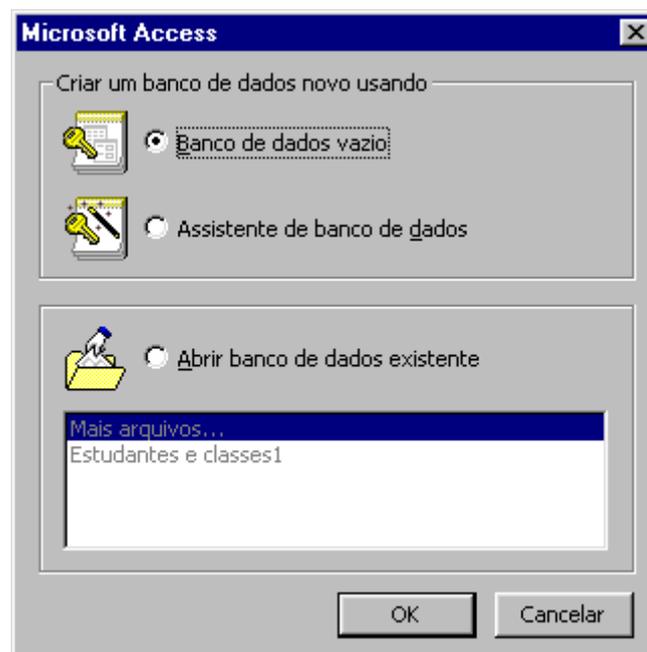
Vamos praticar os conceitos vistos até agora, desenvolvendo um pequeno conjunto de tabelas com o Access 2000. Siga os passos e as imagens básicas de cada tabela e reproduza no Access.

1º passo: abra o Access 2000 clicando no ícone da Figura 6.



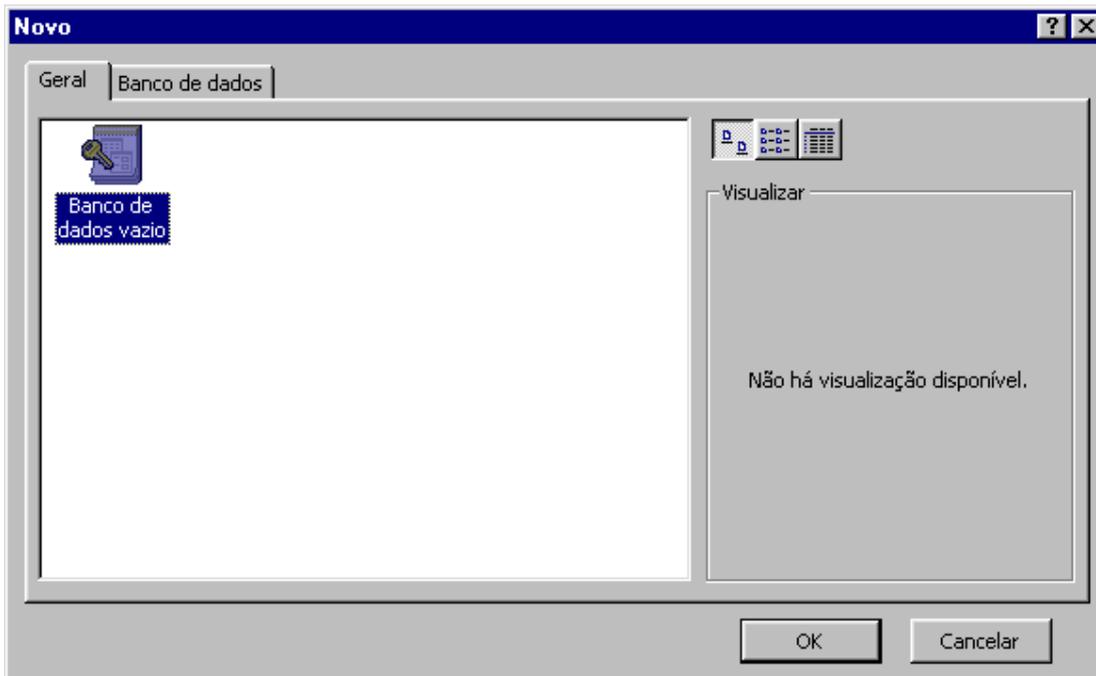
**Figura 6**

2º passo: escolha a opção banco de dados vazio e clique OK (Figura 7).



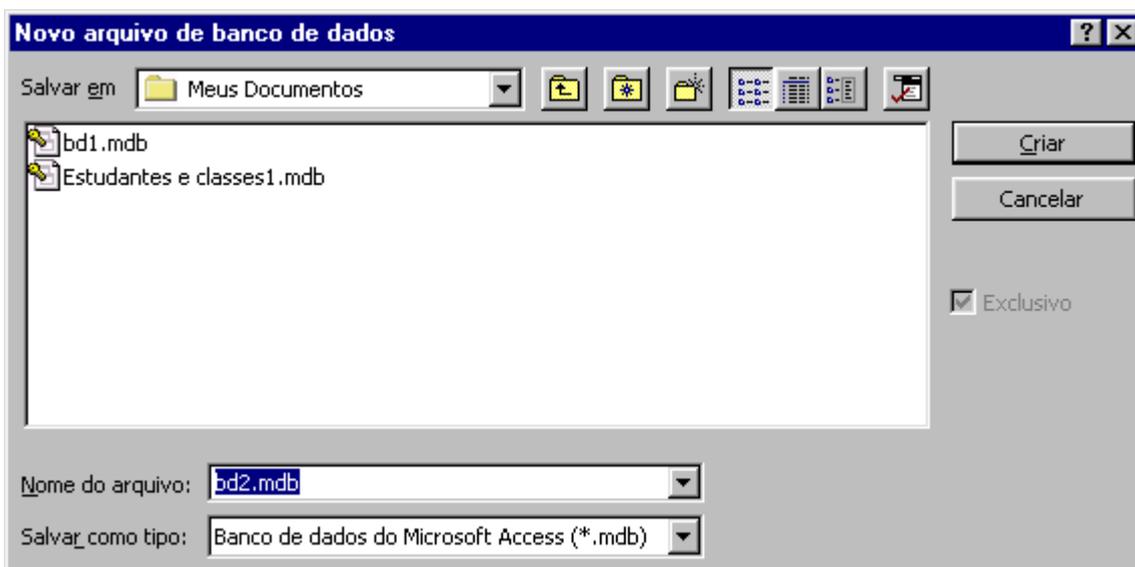
**Figura 7**

3º passo: clique OK novamente ao aparecer a janela da Figura 8.



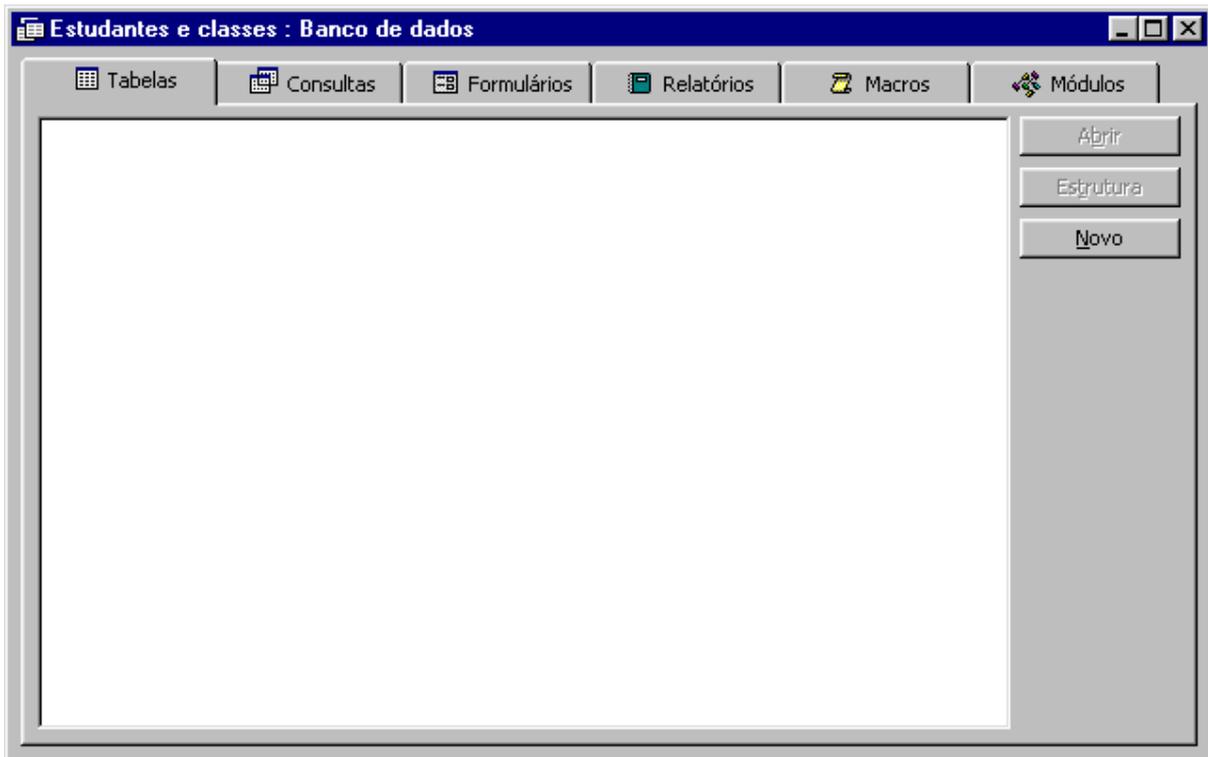
**Figura 8**

4º passo: informe o nome do novo Banco de Dados Access conforme Figura 9 e clique em Criar.



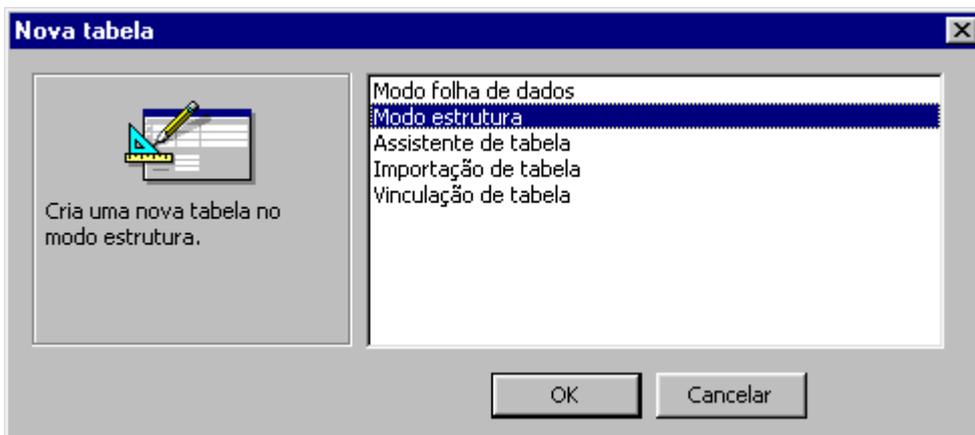
**Figura 9**

5º passo: a janela vazia da Figura 10 será exibida



**Figura 10**

6º passo: clique em Novo e escolha modo estrutura e clique OK (Figura 11).



**Figura 11**

7º passo: digite os dados para a tabela de alunos conforme Figura 12.

Name = Nome do campo

Type = Tipo de dados

Size = Tamanho do campo (detalhe na Figura 13)

---

Segunda-feira, 29 de Setembro de 1997

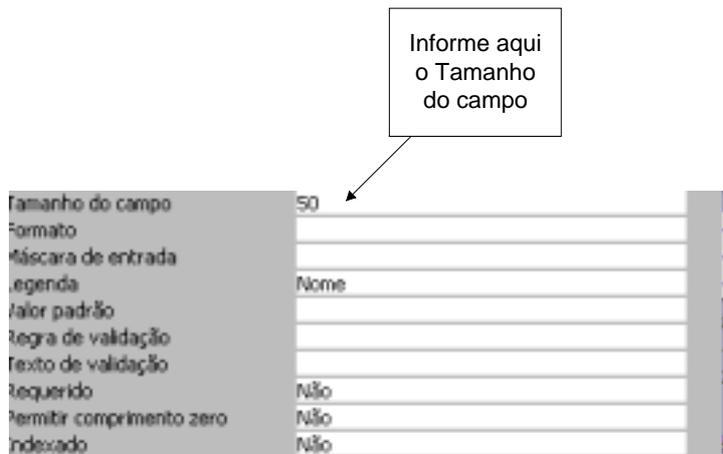
Tabela: Estudantes Página: 1

---

**Columns**

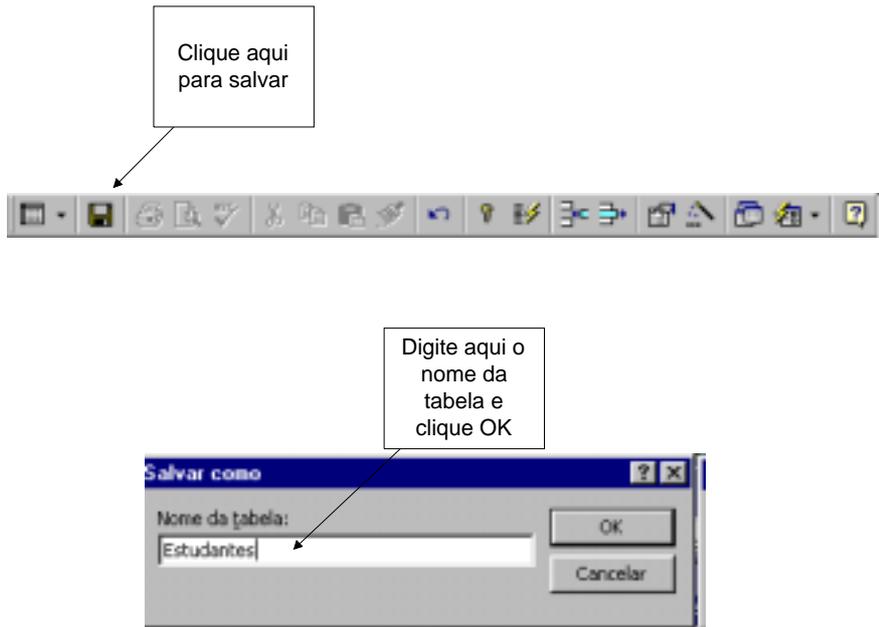
Name	Type	Size
CódigoDoEstudante	Número (Longo)	4
PrimeiroNome	Texto	50
Sobrenome	Texto	50
Endereço	Texto	255
Cidade	Texto	50
EstadoOuProvíncia	Texto	20
CEP	Texto	20
NºDoTelefone	Texto	30
Área de Interesse	Texto	50
NúmeroDoEstudante	Texto	30

**Figura 12**



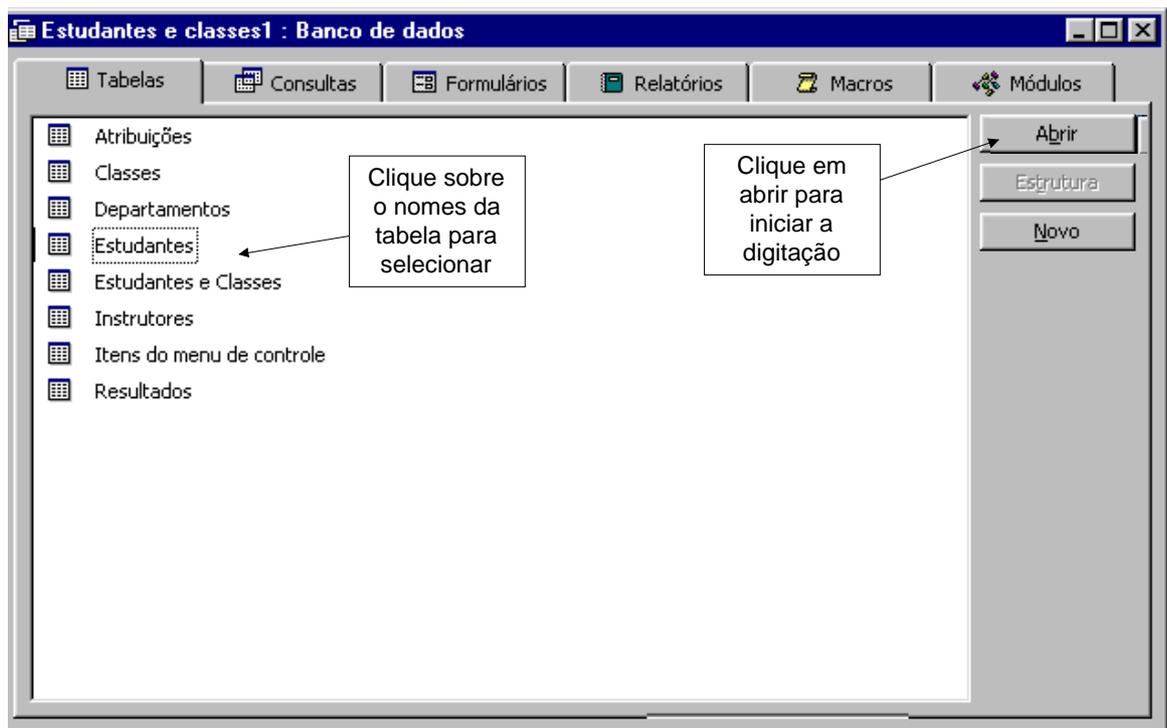
**Figura 13**

8º passo: salve a tabela com o nome Estudantes clicando em salvar (Figura 14). Depois de salvar a tabela está pronta para ser utilizada para digitação dos dados.



**Figura 14**

9º passo: digite alguns dados no modo formulário clicando em abrir (Figura 15 e Figura 16).



**Figura 15**

Digite os dados

Cód. do Estud.	Nome	Sobrenome	Endereço	Cidade	Estado	CEP	Telef
1	Nancy	Davolio	507 - 20th Ave.	Seattle	WA	98122-	(514) 55
2	Tim	Smith	908 W. Capital	Tacoma	WA	98401-	(514) 55
3	Brandon	Coake	722 Moss Bay	Kirkland	WA	98033-	(514) 55
4	Margaret	Peacock	4110 Old Redm	Redmond	WA	98052-	(514) 55
5	Matthew	Dunn	14 Garrett Hill	Seattle	WA	98105-	(514) 55
6	Helvetius	Nagy	722 DaVinci Bl	Kirkland	WA	98034-	(514) 55
7	Deborah	Peterson	305 - 14th Ave.	Seattle	WA	98128-	(514) 55

Figura 16

Repita agora os passos 6 a 9 para criar as tabelas ilustradas nas Figuras 17, 18 e 19, fazendo em seguida a entrada de dados.

Tabela: Classes Segunda-feira, 29 de Setembro de 1997  
Página: 1

**Columns**

Name	Type	Size
CódigoDaClasse	Número (Longo)	4
NomeDaClasse	Texto	50
CódigoDoDepartamento	Número (Longo)	4
NúmeroDaSeção	Número (Inteiro)	2
CódigoDoInstrutor	Número (Longo)	4
Semestre	Texto	30
Unidades	Texto	30
Ano	Número (Inteiro)	2
Local	Texto	255
DiasEHoras	Texto	20
Observações	Memorando	-

Figura 17



D:\Meus Documentos\Estudantes e classes1.mdb  
Tabela: Departamentos

Segunda-feira, 29 de Setembro de 1997  
Página: 2

---

## Columns

<u>Name</u>	<u>Type</u>	<u>Size</u>
CódigoDoDepartamento	Número (Longo)	4
NomeDoDepartamento	Texto	50
NúmeroDoDepartamento	Número (Longo)	4
GerenteDoDepartamento	Texto	30
ChefeDoDepartamento	Texto	50

**Figura 18**

---

D:\Meus Documentos\Estudantes e classes1.mdb  
Tabela: Instrutores

Segunda-feira, 29 de Setembro de 1997  
Página: 3

---

## Columns

<u>Name</u>	<u>Type</u>	<u>Size</u>
CódigoDoInstrutor	Número (Longo)	4
Instrutor	Texto	50
NºDoTelefone	Texto	30
Ramal	Texto	30

**Figura 19**



## 2. Modelagem de Dados

### 2.1. Tabela

#### 2.1.1. Definição

Uma tabela é uma representação gráfica bi-dimensional de dados, composta de colunas e linhas.

#### Exemplo

Tabela de Departamentos

Departamento	
CódigoDepartamento	NomeDepartamento
Vendas	Vendas
Compras	Compras
Pessoal	Pessoal

Esta tabela tem duas colunas e três linhas.

O nome da tabela é **departamento**.

As colunas da tabela são chamadas de CódigoDepartamento e NomeDepartamento.

A primeira linha dessa tabela apresenta o departamento de vendas, a segunda, o departamento de compras e assim por diante.

#### Exercício 1

Empregado		
CódigoEmpregado	NomeEmpregado	CódigoDepartamento
1	João	Vendas
2	José	Compras
3	Maria	Vendas
4	Roberto	Vendas

1. Qual é o nome da tabela?

R.:

2. Quantas colunas?

R.:

3. Qual é o código do empregado Roberto?

R.:

4. Usando também a tabela de departamentos do exemplo anterior. Qual é o nome do departamento em que João trabalha?

R.:

5. Quantos empregados trabalham no departamento de vendas?

R.:

6. Qual é o departamento que tem mais empregados?

R.:

7. Qual departamento que tem menos?

R.:

### Respostas

1. Empregado, 2. 3, 3. 4, 4. Vendas, 5. 3, 6. Departamento de vendas, 7. Pessoal.

### 2.1.2. Propriedades

Existem três importantes propriedades de tabelas:

#### 1. Simplicidade

Tabelas são familiares e auto-explicativas. A maioria das pessoas já usou, Ou pelo menos, já teve contato com tabulações de dados em tabelas. Não foi necessário um treinamento especializado para que você respondesse, corretamente, as questões do exercício anterior.

#### 2. Precisão

Tabelas não possuem ambiguidades. Havia uma, somente uma, resposta correta para cada uma das questões do exercício anterior.

#### 3. Flexibilidade

Tabelas não mostram somente a estrutura dos dados, mas também os exemplos.

#### Exemplo

Tabela mostrando somente a estrutura de dados:

Categoria

CódigoCategoria	NomeCategoria
-----------------	---------------

Tabela mostrando a estrutura de dados e seus exemplos:

Categoria

CódigoCategoria	NomeCategoria
1	Ótimo
2	Bom
3	Regular
4	Mau
5	Péssimo

## 2.2. Domínio

### 2.2.1. Definição

Um domínio é um tipo fundamental de informação de um sistema.

Colunas são baseadas em domínios ou geradas por eles.

Mais de uma coluna pode ser baseada no mesmo domínio.

### 2.2.2. Regra

Duas colunas são baseadas no mesmo domínio se os valores correspondentes de cada se referem a mesma pessoa, coisa ou lugar do mundo real.

#### Exemplo

Cinco colunas, quatro domínios:

Departamento		Empregado		
CódigoDpto	NomeDpto	CódigoEmpreg	NomeEmpreg	CódigoDpto
Vendas	Vendas	1	João	Vendas
Compras	Compras	2	José	Compras
Pessoal	Pessoal	3	Maria	Vendas

#### Domínios

**CódigoDpto, NomeDpto, CódigoEmpreg, NomeEmpreg**

A coluna **CódigoDpto** na tabela **Departamento** e a coluna **CódigoDpto** da tabela **Empregado** estão ambos baseadas no mesmo domínio "**CódigoDpto**", desde que os valores de ambas as colunas correspondam ao mesmo departamento.

Quatro colunas, três domínios:

Cliente		Pedido	
CódigoCliente	NomeCliente	CódigoPedido	CódigoCliente
1	Maria	1	1
2	Maria	2	1

**CódigoCliente, NomeCliente, CódigoPedido.**

A coluna **CódigoCliente** na tabela **Cliente** e **CódigoCliente** da tabela **Pedido** são ambas baseadas no mesmo domínio "**CódigoCliente**", desde que os valores de ambas as colunas correspondam ao mesmo cliente.

## Exercício 2

Liste os domínios da seguinte tabela:

### Empregado

CódigoEmpreg	NomeEmpreg	DataAdmissão	DataNascimento	SalárioEmpreg
1	João	15/01/82	15/07/53	5.000,00
2	José	15/07/83	11/09/48	4.762,90
3	Maria	22/02/80	06/07/61	3.910,90
4	Roberto	12/01/82	06/07/61	7.890,00

## Exercício 3

Liste os domínios das seguintes tabelas:

### Pedido

CódigoPedido	TipoPagamento	DataEmissão	DataRemessa	ValorTotal
1	À vista	15/01/89	25/01/89	5.000,00
2	30 dias	15/01/89	25/01/89	4.762,90
3	30 dias	16/01/89	26/01/89	3.910,90
4	À vista	16/01/89	27/01/89	7.890,00

### Peça

CódigoPedido	NomePeça	CustoPeça	PreçoPeça
1	Porca	15,00	18,00
2	Parafuso	37,70	56,00
3	Arruela	12,00	21,00
4	Chave	87,00	134,00
5	Caixa	114,90	237,00

### PedidoPorPeça

CódigoPedido	CódigoPeça	QuantidadePedida	QuantidadeRemetida
1	2	100	100
1	3	80	78
2	5	340	340
3	1	1.500	1.500
4	1	3.000	3.000

### 2.2.3. Exposição

Domínios são fundamentais para o modelo relacional, porque definem comparações válidas entre tabelas. Os domínios mostram homônimos e sinônimos, fazendo com que o escopo do sistema esteja sempre sob controle.

### 2.2.4. Diretriz

A quantidade de domínios deve ser minimizada.

## 2.3. Convenções e Nomenclatura

### 2.3.1. Regras

Nomes de tabelas devem ser únicos dentro de um sistema.

Nomes de colunas devem ser únicos dentro de uma tabela.

### 2.3.2. Diretrizes

#### 1. Selecione nomes de tabelas e colunas com muito cuidado.

Nomes de tabelas e colunas não só identificam, mas também descrevem cada tabela e cada coluna. Nomes de tabelas e colunas são referenciados por toda a documentação do sistema. Nome de tabelas e colunas aparecem em terminais e listagens do sistema

#### 2. Selecione nomes de tabelas e colunas com o usuário.

Os usuários do sistema estão, geralmente, mais familiarizados com a terminologia de suas informações do que o analista de sistemas. Os usuários do sistema vão usar estes nomes para se comunicar com sistema enquanto o sistema existir.

#### 3. Use nomes de tabelas e colunas o mais curto possível.

Nomes de tabelas e colunas são freqüentemente referenciados nas conversações, como também em toda a documentação escrita.

#### 4. Use nomes no singular.

A experiência mostra que o modo singular é mais apropriado para dar nomes a tabelas e colunas.

#### 5. Se possível, use nomes de domínios sem qualificação.

A qualificação deveria ser colocada após o nome do domínio, quando presente.

### 2.3.3. Definições

Um nome qualificado de coluna é:

NomeDoDomínio.qualificador

Um nome totalmente qualificado de coluna é:

NomeDaTabela.NomeDoDomínio.qualificador

#### Exemplo

Os nomes de domínios que normalmente precisam de qualificação são:

Data, Hora/Minuto, Quantidade, Valor.

Nomes de colunas tipicamente qualificadas:

DataNascimento, DataAdmissão, DataPedido

HoraEfetivação, HoraEntrada, HoraInício

QuantidadeMáxima, QuantidadeRemetida

ValorMédio, ValorSalário, ValorPreço

Nomes de colunas totalmente qualificados:

Empregado.DataNascimento, Pedido.DataEmissão

Empregado.HoraExtra, Empregado.HoraAlmoço

Pedido.QuantidadePedida, Pedido.QuantidadeRemetida

Pedido.ValorPreço, Pedido.ValorCotado

## Exercício 4

Complete a tabela abaixo determinando os domínios e definindo nomes qualificados para os dados informados:

### Exercício 4

Item Dados	Nome Domínio	Nome Coluna
Nome do empregado		
Código do empregado		
Data do nascimento		
Data da admissão		
Salário bruto		
Imposto federal		
Imposto estadual		
salário líquido		
Código do gerente		
Data da demissão		
Valor da dedução		



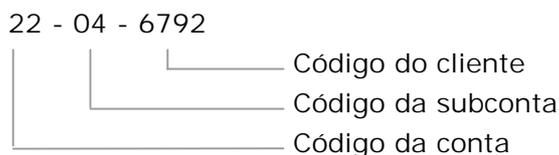
## 2.4. Regras Adicionais

### 2.4.1. Regra 1

Domínios (conseqüentemente colunas), devem ser itens que não podem ser decompostos.

#### Exemplo

O número da conta:



Deveria ser representado pela tabela abaixo:

#### Departamento

CódigoConta	CódigoSubConta	CódigoCliente
22	04	6792

#### Nota

Por conveniência, os domínios de **Data** e **Hora** são reconhecidamente exceções a esta regra. É entendido que o **Domínio** data é dividido em três domínios distintos (**Dia, Mês, Ano**) e o domínio **Hora** também é dividido em três domínios diferentes (**Hora, Minuto, Segundo**).

### 2.4.2. Regra 2

A ordem das colunas de uma tabela deve ser arbitrária.

#### Exemplo

As tabelas abaixo não são válidas porque as colunas não podem ser rearranjadas sem que os dados nelas contidos percam seu significado.

#### VendaPorEmpregado

CódigoConta	Quantidade dos últimos 6 meses					
	Venda	Venda	Venda	Venda	Venda	Venda

#### Pedido

CódigoPedido	Detalhe do Pedido 9 Ocorrências				
	Peça	Quantidade	Peça	Quantidade	Peça ...

#### Exposição

Note que se a ordem das colunas de uma tabela não for arbitrária, deverão aparecer uma ou mais das propriedades indesejáveis, descritas a seguir:

Dados implícitos;  
Número limitado de ocorrências;  
Manutenção anormal obrigando a remanejamento de dados quando o total das ocorrências for atingido;  
Anomalia de recuperação (leitura), porque a posição atual (ou a última posição) é imprevisível.

### 2.4.3. Regra 3

A ordem das linhas na tabela deve ser arbitrária.

#### Exemplos

A tabela abaixo não é válida pois suas linhas não podem ser rearranjadas sem alterar o significado dos dados.

Histórico.Salário

CódigoEmpreg	ValorSalário
1	5.000,00
1	4.000,00
1	3.000,00
2	3.000,00
2	2.000,00
2	1.000,00

### 2.4.4. Regra 4

Cada linha na tabela deve ser única.

#### Exemplo

Tabela abaixo não é válida pois suas linhas não são sempre únicas (ver empregado 3).

Histórico.Salário

CódigoEmpreg	ValorSalário
1	5.000,00
1	3.750,00
1	2.100,00
2	3.450,00
2	2.310,00
2	1.500,00
3	7.800,00
3	4,600,00
3	4,600,00

## 2.5. Valores Duplicados

### Definição

Um valor duplicado é um valor, ou um grupo de valores que é exatamente igual a um valor ou grupo de valores, em linhas da mesma coluna (colunas).

### 2.5.1. Regra

Colunas ou grupo de colunas, assinaladas com a sigla "ND" (*no duplicates*), não podem ter valores iguais.

### Exercício 5

Assinale as violações acima na seguinte tabela:

Empregado

CódigoEmpreg	NomeEmpreg	CódigoCategoria	DataNascimento	SalárioEmpreg
ND	ND			
1	João	1	53/07/15	5.000,00
2	João		48/09/11	10.000,00
2	José	2	61/07/06	
4	Roberto	2	53/12/22	7.890,00



## 2.6. Valores Nulos

### Definição

Um valor nulo é uma lacuna de uma tabela, isto é, é uma entrada que não existe.

Valores nulos significam "desconhecimento" e não devem ser confundidos com zeros ou espaços.

A equivalência de dois valores nulos é indeterminada, isto é, dois valores nulos não são necessariamente iguais.

### 2.6.1. Regra

Colunas ou grupo de colunas assinaladas com a sigla "NN" (*not null*) não podem ter valores nulos.

### Exercício 6

Indique valores nulos, na tabela abaixo, com o sinal "-" e mostre as violações das regras "ND" e "NN".

Empregado

CódigoEmpreg	NomeEmpreg	CódigoCategoria	DataNascimento	SalárioEmpreg
ND, NN	ND			NN
1		1	53/07/15	5.000,00
1	João		48/09/11	0.000,00
2		2	61/07/06	
4	Roberto	2	53/12/22	7.890,00



## 2.7. Chave Primária

### Definição

Uma chave primária é uma coluna ou um grupo de colunas que assegura a unicidade das linhas dentro de uma tabela. Uma chave primária que tenha mais de uma coluna é chamada de chave primária composta. Chaves primárias não implicam em sequência de acesso.

### 2.7.1. Regra 1

Chaves primárias são indicadas pela sigla "PK" (*primary key*), imediatamente abaixo do cabeçalho da coluna apropriada.

### Exemplo

Assumindo que **CódigoEmpreg** seja sempre único:

#### Empregado

CódigoEmpreg	NomeEmpreg	CódigoCategoria	DataNascimento	SalárioEmpreg
PK				

Obs: Nem toda chave única é PK, mas toda PK é chave única.

### Exemplo

Assumindo que a combinação de **Código.Pedido** e **Código.Peça** seja sempre única:

#### PeçaPorPedido

CódigoPedido	CódigoPeça	QuantidadePedida	QuantidadeRemetida	ValorUnitário
PK				

Mudança (alteração) de dado que é PK, significa exclusão.

### 2.7.2. Regra 2

Valores de chaves primárias não podem ser nulos ("PK" implica em "NN").

Valores de chaves primárias não podem ser nulos porque uma linha sem chave primária não pode ser distinguida de outra linha na mesma tabela.

### 2.7.3. Regra 3

Colunas com chaves primárias não podem conter valores duplicados, "PK" implica em "ND", por definição.

### 2.7.4. Regra 4

Chaves primárias não podem ser alteradas, ("NC" = *no change*) "PK" implica em "NC".

Esta regra é freqüentemente objeto de debates entre os teóricos dos banco de dados relacionais. Em certos casos parecem ser viáveis modificações em algumas chaves primárias compostas. Em termos gerais porém, é recomendável manter-se fiel a esta regra.

### 2.7.5. Diretrizes

1. Selecione chaves primárias absolutamente disciplinadas e que permaneçam únicas. Chaves primárias devem ser designadas de uma única maneira muito disciplinada e de uma única fonte. Chaves primárias atribuídas pelo sistema são ideais.

2. Selecione chaves gerais que não tenham qualquer tendência de alterações. Não é uma boa idéia incluir atributos descritivos em uma chave primária, pois ela tem tendência de mudar cada vez que o atributo mudar. Uma exposição mais detalhada a respeito desta diretriz será feita no tópico "Chaves Inteligentes".
3. Se possível, selecione chaves primárias compactas. Chaves primárias são frequentemente usadas e referenciadas no sistema. Usuário e sistemas podem ser afetados negativamente por chaves muito grandes.
4. Selecione chaves primárias numéricas. Chaves primárias numéricas são mais fáceis de acessar e gerenciar. Um número é mais facilmente disponível somando-se um a mais ao número já existente. Além disso, números são mais fáceis de serem guardados em comunicações verbais. Para uma possível exceção a esta diretriz, veja o tópico "Entidade Menores".
5. Selecione chaves primárias que são familiares ao usuário do sistema. Sempre que possível use os códigos já existentes, os identificadores etc. Esta diretriz, no entanto, não deve ser praticada em detrimento das outras diretrizes.

### 2.7.6. Regra 5

Chaves primárias que são atribuídas pelo sistema são identificadas pela sigla "SA" (*system assigned*), logo abaixo do cabeçalho da coluna correspondente.

## 2.8. Chave Estrangeira

### Definição

Uma chave estrangeira é uma coluna ou um grupo de colunas que pode ou não fazer parte da chave primária de uma tabela, mas certamente é chave primária de uma outra tabela. Uma chave estrangeira composta de mais de uma coluna é chamada de chave estrangeira composta.

### 2.8.1. Regra 1

Chaves estrangeiras são indicadas pela sigla "FK" (*foreign key*) abaixo do cabeçalho da coluna apropriada.

### Exemplo

**Empregado.CódigoDpto** é uma chave estrangeira já que não é chave primária na tabela **Empregado**, mas é baseada no mesmo domínio da chave primária da tabela **Departamento**.

#### Departamento

CódigoDpto	NomeDpto
PK	

#### Empregado

CódigoEmpreg	NomeEmpreg	CódigoDpto
PK		FK

### Exemplo

**Pedido.CódigoCliente** é uma chave estrangeira já que não é chave primária na tabela **Pedido**, mas é baseada no mesmo domínio da chave primária da tabela **Cliente**.

#### Cliente

CódigoCliente	NomeCliente
PK	

#### Pedido

CódigoPedido	CódigoCliente
PK	FK

### Exercício 7

Localize as chaves estrangeiras das seguintes tabelas.

#### Edifício

NúmeroEdifício	NomeEdifício
PK	

#### Departamento

CódigoDpto	NomeDpto	NúmeroEdifício
PK		

## Gerente

NúmeroGerente	NomeGerente
PK	

## Projeto

CódigoProjeto	NomeProjeto	NúmeroGerente
PK		

### Exercício 8

Localize as chaves estrangeiras das seguintes tabelas.

## Pedido

CódigoPedido	TipoPagamento	DataEmissão	DataRemessa	ValorTotal
PK				

## Peça

CódigoPeça	NomePeça	CustoPeça	PreçoPeça
PK			

## PeçaPorPedido

CódigoPedido	CódigoPeça	QuantidadePedida	QuantidadeRemetida
PK			

### Exercício 9

Localize as chaves estrangeiras das seguintes tabelas.

## Empregado

CódigoEmpreg	NomeEmpreg	CódigoGerente
PK		
1	João	2
2	José	
3	Maria	2
4	Roberto	2

## Peça

CódigoPeça	NomePeça	CódigoPeçaAnterior	PreçoPeça	CustoPeça
PK				

## PeçaPorComprador



CódigoPeça	CódigoComprador	QuantidadePeça
PK		

Edifício		Departamento		
NúmeroEdifício	NomeEdifício	CódigoDpto	NomeDpto	NúmeroEdifício
PK		PK		FK
1	Central	Vendas	Vendas	1
2	Anexo 1	Compras	Compras	
3	Administração	Pessoal	Pessoal	3

### 2.8.2. Regra 2

Chaves estrangeiras podem ter valores nulos, mas se houver valor, este deve existir na tabela onde o domínio da chave estrangeira é chave primária.

#### Exercício 10

Mostre as violações da regra acima nas seguintes tabelas:

Gerente		Projeto		
NúmeroGerente	NomeGerente	NúmeroProjeto	NomeProjeto	NúmeroGerente
PK		PK		FK
2	José	101	Reator	2
3	Alan	102	Motor	1
4	Lais	103	Bomba	6
5	Carlos	104	Freio	
6	Lúcia	109	Cambio	8

## 2.9. Ler Tabelas

### 2.9.1. Exposição 1

Já demonstramos que tabelas são simples, precisas e flexíveis.

Outra propriedade desejável é o fato delas serem concisas.

Cada tabela de um sistema conta-nos uma série de coisas a respeito dos itens de dados, seus inter-relacionamentos e, até mesmo (como veremos) todos os processos que podem ser realizados com os dados.

#### Exemplo

Dadas duas tabelas:

Empregado

CódigoEmpreg	NomeEmpreg	CódigoDpto
PK	NN	FK

Departamento

CódigoDpto	NomeDpto
PK	ND, NN

Nós sabemos que:

- . O sistema refere-se a departamento e empregado (dos nomes das tabelas).
- . Cada departamento tem que ter um código de departamento, porque "PK" implica em "NN".
- . Cada departamento tem um código único, porque "PK" implica em "ND".
- . Os códigos de departamentos não podem mudar, porque "PK" implica em "NC".
- . Cada departamento tem um nome ("NN").
- . Cada departamento tem um nome (cada código de departamento aparece somente em uma linha).
- . Dois departamentos não podem ter mesmo nome ("ND").
- . Cada departamento pode ter zero, um ou mais empregados ("FK").
- . Todos os empregados têm seu próprio código, porque "PK" implica em "ND".
- . Código de empregados não podem ser alterados, porque "PK" implica em "NC".
- . E assim por diante...

### Exercício 11

Dadas duas tabelas:

Departamento

CódigoDpto	NomeDpto
PK	ND, NN

Empregado

CódigoEmpreg	NomeEmpreg	CódigoDpto
PK	NN	FK

**Responda as seguintes perguntas:**

1. Dois empregados podem ter o mesmo nome?

R.:

2. Um empregado pode não ter um nome?

R.:

3. Em quantos departamentos um empregado pode trabalhar?

R.:

4. Um empregado tem que ser alocado a um departamento?

R.:

5. Dois empregados com o mesmo nome podem trabalhar no mesmo departamento?

R.:

6. Um código de departamento pode aparecer na tabela de empregado sem estar na tabela de departamento?

R.:

### **Respostas**

1. Pode. 2. Não. 3. 1. 4. Não. 5. Sim. 6. Não.

### 2.9.2. Exposição 2

Assim, podemos perceber que as tabelas nos contam uma porção de coisas a respeito dos dados do sistema de uma maneira bastante concisa.

Mas, tabelas também podem nos contar uma porção de coisas a respeito do processamento do sistema.

Processos podem ser divididos em duas categorias:

- Manutenção de dados - inclusão, alteração e exclusão.
- Recuperação de dados - seleção, projeção, junção e agrupamento.

Juntar (mostrar informações de duas ou mais tabelas).

- Mostrar todos os códigos e nomes de empregados e código e nome de departamento de um dado departamento.
- Mostrar todos os códigos e nomes de departamentos atualmente se empregados.
- E assim por diante...

Agrupar (sumarizar informações de uma ou mais tabelas).

- Contar o número de departamentos.
- Contar o número total de departamentos.
- Determinar número empregados por departamento.
- Determinar número mínimo de empregados em qualquer departamento.
- Determinar número máximo de empregados em qualquer departamento.
- Determinar número médio de empregados em todos departamento.
- E, assim por diante...

### 2.9.3. Exposição 3

Assim, podemos ver o grande número de informações que podem ser documentadas, em um par de tabelas realmente muito pequenas, sem sacrifício da simplicidade e sua completude.

Mas, existem outras implicações...

### 2.9.4. Exposição 4

Deve ter ficado claro que a lista de processos no exemplo dado anteriormente, não está, de maneira alguma, completa. E também deve ficar entendido que uma lista completa de processos de, por exemplo um sistema médio (de 35 a 80 tabelas) é alguma coisa bastante extensa.

Estas duas observações, leva-nos a considerações bastante importantes:

1. Analistas de sistemas devem-se concentrar mais em informações (dados) do sistema e não em seus processos, porque:

1. A lista dos processos é perto de interminável.
2. Os dados, em conjunto com "Standards" de operações sobre telas (incluir, selecionar etc), definem o processo.
3. Linguagens estão cada vez mais orientadas a objetos.

2. Sistemas devem ser interativos "programados pelos usuários", porque:

1. A lista dos processos é perto de interminável.
2. Os dados, em conjuntos com "Standards" de operações sobre tabelas (incluir, selecionar etc), permitem ao usuário fazer todos os processos de que ele necessita.

3. Sistemas devem ser muito bem otimizados de maneira tal que não venham a ter desvios funcionais, porque:

1. A lista dos processos é perto de interminável.
2. Os dados, em conjunto com "Standards" de operações sobre tabelas (incluir, selecionar etc), e acessos genéricos, suportam os acessos necessários ao usuário, com uma boa performance.

## 2.10. Revisão

### 2.10.1 Definição de Tabela

Uma tabela é uma representação gráfica bi-dimencional de dados, composta de colunas e linhas.

#### Regras

Nomes de tabelas tem que ser únicos no sistema.

Nomes de colunas tem que ser únicas na tabela.

Colunas, exceto **Data** e **Hora**, não devem se compostas.

A ordem das colunas devem ser arbitrária.

A ordem das linhas de uma tabela deve ser arbitrária.

### 2.10.2. Definição de Domínio

Um domínio é um tipo fundamentos de informação dentro de um sistema.

#### Regra

Duas colunas são baseadas no mesmo domínio se os valores correspondentes de cada coluna se regerem mesma pessoa, coisa ou lugar do mundo real.

### 2.10.3. Definições para Colunas

Um nome de coluna qualificada é uma coluna representada da seguinte maneira:

nome\_do\_domínio.qualificador

Um nome de coluna totalmente qualificada é uma coluna representada da seguinte maneira:

nome\_da\_tabela.nome\_do\_domínio.qualificador

### 2.10.4. Definição de Valor Duplicado

Um valor duplicado é um valor ou grupo de valores de uma tabela que é exatamente igual a valor ou grupo de valores na mesma coluna.

#### Regra

Colunas ou grupo de colunas assinaladas com a sigla "ND" não podem conter valores duplicados.

### 2.10.5. Definição de Valor Nulo

Um valor nulo é uma lacuna em uma coluna de tabela.

#### Regra

Colunas assinaladas com a sigla "NN" não podem conter valores nulos.

### 2.10.6. Definição de Chave Primária

Chave primária é uma coluna ou grupo de colunas que assegura a unicidade de linha dentro de tabelas. Uma chave primária que tem mais de uma coluna é chamada de chave primária composta.

### Regras

Chaves primárias são identificadas pela sigla "PK" sob o cabeçalho da coluna apropriada.

Valores de chave primária não podem ser nulos, porque "PK" implica em "NN".

Colunas de chaves primárias não podem conter valores duplicados, porque "PK" implica em "ND".

Chaves primárias não podem ser alterados, porque "PK" implica em "NC".

Chave primária atribuída pelo sistema é indicada pela sigla "SA" (System Assigned) sob o cabeçalho da coluna apropriada.

### 2.10.7. Definição de Chave Estrangeira

Uma chave estrangeira é uma coluna ou um grupo de colunas que podem ou não ser chave primária de uma tabela, mas certamente é chave primária de uma outra tabela. Uma chave estrangeira composta de mais de uma coluna é chamada de chave estrangeira composta.

### Regras

Chaves estrangeiras são identificadas pela sigla "FK" sob o cabeçalho da coluna apropriada.

Valores de chave estrangeira podem ser nulos ou existir na tabela onde a chave estrangeira é chave primária.

### 2.10.8. Comentário Geral

Nós escolhemos tabelas como ferramenta de modelagem de dados porque elas são:

    Simples;

    Precisas;

    Flexíveis;

    Concisas.

Analistas de sistemas devem concentrar-se em dados do sistema e não em seus processos.

Sistemas devem ser interativos e "programáveis pelo usuário".

Sistemas devem ser muito bem otimizados de maneira a não ter desvios.

### 2.11. Modelo de Dados

#### 2.11.1. Definição de Modelo de Dados

Modelo de dados é uma coleção de construções, operadores e regras de integridade que juntos, suportam uma representação dinâmica dos objetos e eventos do mundo real.

#### 2.11.2. Construções

Construções são os blocos básicos de um modelo de dados. Modelos de dados relacionais usam um tipo único de construção - tabelas.

#### 2.11.3. Operadores

Operadores são os modos com os quais os dados, no modelo de dados, são mantidos e pesquisados.

Operadores tipicamente relacionais são "incluir", "alterar", "excluir", "selecionar", "projetar", "juntar", "agrupar", e assim por diante.

#### 2.11.4. Regras de Integridade

Regras de integridade têm por finalidade manter ordem e consistência em um modelo de dados. "no-nulls", "no-duplicates", "chaves primárias" e "chaves estrangeiras" são exemplos das regras de integridade.

#### 2.11.5. Reflexão sobre Tabelas e Entidades

É importante que comecemos a pensar em **tabelas**, não como arquivos, ou inputs, ou maneiras de obter relatórios, mas como uma réplica em miniatura de pessoas, lugares ou coisas que são do interesse da comunidade dos usuários.

A primeira linha da tabela de empregados, por exemplo, não estão os dados a respeito de **Joao**, está o **João**, ele mesmo!

Da mesma maneira nós devemos parar de pensar sobre operações em tabelas como "listas de programas" ou "alterações em arquivos", e começar pensar como o usuário vê este modelo e como ele o mantém sincronizado com seu mundo real.

A rotina que conta as linhas da tabela de **pedidos**, por exemplo, não é um programa que emitirá o relatório, é equivalente (no mundo modelo) a alguém contando os pedidos fisicamente (no mundo real).

E devemos pensar a respeito de regras de integridade como coisas básicas do que "edições". Regras de integridade são, no mundo modelo, reflexões de leis físicas que são operadas no mundo real. Por exemplo, clientes tem que existir antes de haverem pedidos para eles, assim no mundo modelo, clientes não podem aparecer na tabela de pedidos, antes de existir na tabela de clientes. Nada no mundo real, jamais acontecerá no dia 32 de setembro, portanto a informação 32/09 será sempre uma entrada inválida.

### 2.12. Entidades

#### 2.12.1. Definição

Uma entidade é uma pessoa, um lugar ou uma coisa de interesse da comunidade dos usuários, sobre o qual o sistema deve manter, correlacionar e dar informações.

#### 2.12.2. Exposição

Entidades são "substantivos" (nomes).

Entidades estão dentro do escopo do sistema.

Entidades têm existência própria e não são dependentes ou subordinados a qualquer outra coisa.

Entidades podem ser tangíveis (tais como edifícios ou empregados), intangíveis (tais como departamentos ou contas) e semi-tangíveis (tais como pedidos, faturas).

As características das entidades (tais como nome do empregado) e outras informações sobre as entidades (o total de clientes ou média de pedidos do mês de maio) não são considerados entidades.

#### Exercício 12

Ache as entidades na seguinte descrição de um usuário:

"Você sabe, faz cinco anos desde que mamãe e eu começamos este pequeno negócio de loja de produtos naturais, e agora veja, temos cinco delas! Em três estados!"

Bem, você pode imaginar que controlar estas coisas está se tornando um grande problema. Parece que alguns itens sempre estão faltando em uma loja, enquanto estão saindo pela janela em outra!

E os empregados! Antes éramos somente mamãe e eu. Agora temos seis outros e eu nem mesmo me lembro quem trabalha onde!".

#### Exercício 13

Localize as entidades deste extrato de entrevista com um usuário:

"Você vê, nós temos vendedores no campo, tentando vender nossos produtos a pessoas de sua região. O problema é que algumas de nossas novas contas são realmente empresas especializadas e alguns dos vendedores não estão qualificados para atendê-los."

Assim, nós precisamos de algum modo classificar os clientes e saber quais empregados estão treinados nestas especialidades, assim nós podemos mandar ao cliente alguém que não se faça de bobo, e não nos faça de bobo também!".

### 2.13. Associações

#### 2.13.1. Definição

Uma associação é um relacionamento entre duas ou mais entidades (ou outras associações), do interesse da comunidade de usuários, sobre o que o sistema deve manter, correlacionar e dar informações.

#### 2.13.2. Exposição

Associações ocorrem tipicamente entre uma e outra entidade (**clientes** e **pedidos**, por exemplo, ou **peças** e **pedidos**), mas podem envolver qualquer número de entidades e inter-relacionamentos (veja o tópico "Associações complexas" para mais exposições deste assunto).

Associações estão no escopo do sistema.

Associações ocorrem em três diferentes maneiras (explicada no tópico a seguir).

#### Exercício 15

Ache as associações na seguinte descrição de um usuário:

"Você sabe, faz cinco anos desce que mamãe e eu começamos este pequeno negócio de loja de produtos naturais, e agora veja, temos cinco delas! Em três estados!"

Bem, você pode imaginar que controlar estas coisas está se tornando um grande problema. Parece que alguns itens sempre estão faltando em uma loja, enquanto estão saindo pela janela em outra!

E os empregados! Antes éramos somente mamãe e eu. Agora temos seis outros e eu nem mesmo me lembro quem trabalha onde!".

#### Exercício 16

Localize as associações de um resumo de uma entrevista com um usuário (assuma que "produtos" está fora do escopo do sistema).

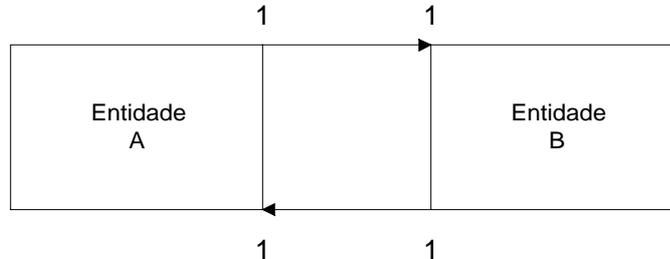
"Você vê, nós temos vendedores no campo, tentando vender nossos produtos a pessoas de sua região. O problema é que algumas de nossas novas contas são realmente empresas especializadas e alguns dos vendedores não estão qualificados para atendê-los."

Assim, nós precisamos de algum modo classificar os clientes e saber quais empregados estão treinados nestas especialidades, assim nós podemos mandar ao cliente alguém que não se faça de bobo, e não nos faça de bobo também!"

### 2.13.3. Associações “um-para-um”

Uma associação “um-para-um” ocorre quando duas entidades (digamos “A” e “B”), estão relacionadas da seguinte maneira:

Cada ocorrência da entidade “A” está relacionada a, no máximo, uma ocorrência da entidade “B” e cada ocorrência da entidade “B” está relacionada a, no máximo, uma ocorrência da entidade “A”.

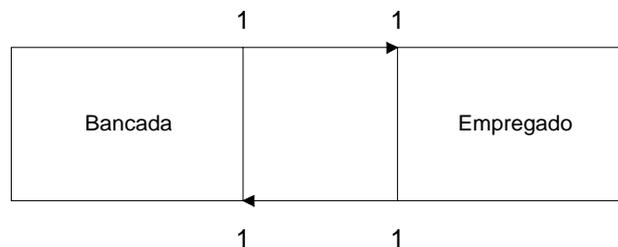


Associações “um-para-um” são modeladas colocando a chave primária da tabela da entidade “A” como coluna de chave estrangeira, ND, na tabela da entidade “B”.

Associações do tipo “um-para-um” são relativamente raras.

#### Exemplo

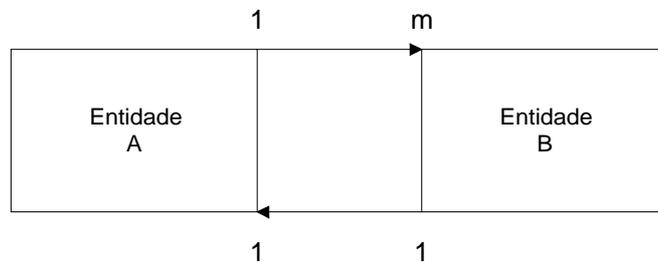
O usuário do sistema é um gerente de uma oficina muito pequena. A oficina tem oito bancadas, numeradas de 1 a 8. Cinco técnicos estão empregados atualmente, cada um tendo um número de empregado, e cada um está designado para sua própria bancada.



1	-----	1
2	-----	2
3	-----	3
4	-----	4
5	-----	5
6		
7		
8		

### 2.13.4. Associações “um-para-muitos”

Uma associação “um-para-muitos” ocorre quando duas entidades (digamos “A” e “B”), estão relacionadas da seguinte maneira: cada ocorrência da entidade “A” está relacionada a zero, uma ou mais ocorrências da entidade “B”, mas cada ocorrência da entidade “B” está relacionada com, no máximo, uma ocorrência da entidade “A”.

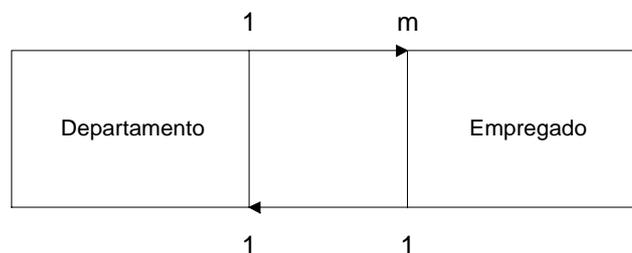


Associações “um-para-muitos” são modeladas, colocando a chave primária da tabela da entidade “A” como coluna de chave estrangeira na tabela da entidade “B”.

Associações do tipo “um-para-muitos” são bastante comuns.

#### Exemplo

Já vimos, em exemplos anteriores, um bom número de associações “um-para-muitos” (1:m): **Departamento** e **Empregado** por exemplo:



Desenhando as tabelas das entidades **Departamento** e **Empregado**:

Departamento	Empregado
CódigoDpto	NúmeroEmpreg
PK	PK
Vendas	1
Compras	2
Pessoal	3
	4

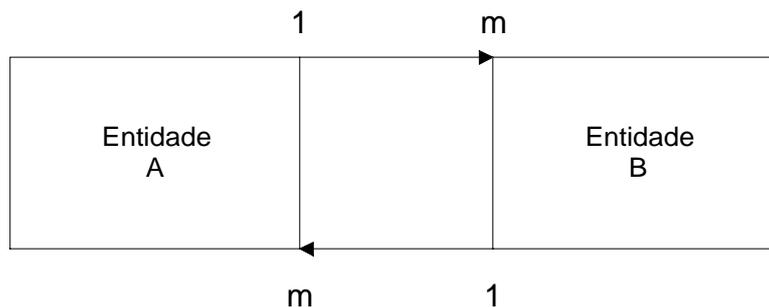
E a chave primária de **Departamento** como chave estrangeira na tabela de **Empregado**:

Departamento	Empregado	
CódigoDpto	NúmeroEmpreg	CódigoDpto
PK	PK	FK
Vendas	1	Vendas
Compras	2	Compras
Pessoal	3	Vendas
	4	Vendas

Note que associações “um-para-muitos” (1:m), são muito parecidas com associações “um-para-um” (1:1), na forma das tabelas. A diferença fundamental é que não há aplicação da regra que obriga “ND” na chave estrangeira.

### 2.13.5. Associações “muitos-para-muitos”

Uma associação “muitos-para-muitos” ocorre quando duas entidades (digamos “A” e “B”), estão relacionadas da seguinte maneira: cada ocorrência da entidade “A” está relacionada a zero, uma ou mais ocorrências da entidade “B”, e cada ocorrência da entidade “B” está relacionada com zero, uma ou mais ocorrências da entidade “A”.



Associações “muitos-para-muitos” são modeladas, definindo-se uma nova tabela com uma chave primária composta. Os componentes da nova chave primária são as chaves primárias das tabelas das entidades “A” e “B”. Ambos os componentes da nova chave primária são também chaves estrangeiras individuais.

Associações do tipo “muitos-para-muitos” são bastante comuns.

#### Exemplo

O modelo de dados para a associação “muitos-para-muitos” anterior:

A	B
ChaveA	ChaveB
PK	PK
A1	B1
A2	B2

APorB	
ChaveA	ChaveB
PK	PK
FK	FK
A1	B1
A1	B2
A2	B1

Já que as associações “muitos-para-muitos” são simétricas e, desde que a ordem das colunas em tabelas devem ser arbitrária, a tabela acima poderia ter sido desenhada ao contrário, ou seja, **B/A**, com a **Chave\_B** em primeiro lugar.

## 2.14. Tabelas Associativas

### 2.14.1. Definição

Uma tabela **associativa** é uma tabela não primária porque sua chave primária é composta de chaves estrangeiras das tabelas que definem esta associação.

### 2.14.2. Regra

Tabelas associativas sempre representam associações “muitos-para-muitos” (m:m), ou seja, associações “m:m” são representadas por tabelas associativas.

### 2.14.3. Exposição

As linhas de uma tabela associativa não representam pessoas, lugares ou coisas.

Por isso deve ficar assegurado que as tabelas primárias devem ser apresentadas em primeiro lugar, antes que sejam definidas as associações entre elas.

#### Exemplo

A tabela **A/B** definida anteriormente é uma tabela associativa:

APorB	
Chave A	Chave B
PK	
FK	FK
A1	B1
A1	B2
A2	B1

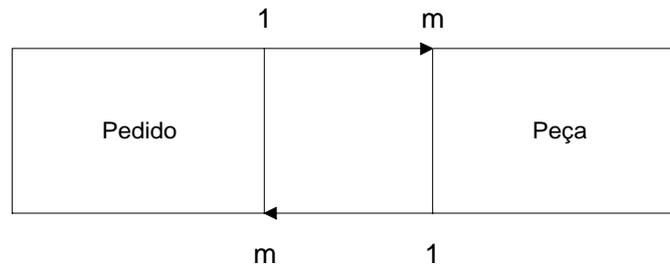
Ela é uma tabela não-primária porque existe mais de uma coluna como sua chave primária e é uma tabela associativa porque cada componente da chave primária é chave estrangeira.

As linhas desta tabela não representam os “A’s” nem os “B’s”, mas o relacionamento entre “A’s” e “B’s”.

Note que associações "1:1" e "1:m" são modeladas dentro das próprias tabelas das entidades, enquanto associações "m:m" requerem uma tabela adicional (associativa). Não deixe que esta "aparente" assimetria do modelo relacional perturbe você!

### Exemplo

Nós já vimos, em exemplos anteriores, um bom número de associações "muitos-para-muitos" (m:m). O exemplo mais conhecido é provavelmente a associação entre pedidos e peças:



Desenhando as duas tabelas das entidades **Pedido** e **Peça**:

Pedido	Peça
NúmeroPedido	NúmeroPeça
PK	PK
1	1
2	2
	4

E as duas chaves primárias compoendo a chave primária composta da tabela associativa:

PeçaPorPedido	
Número Pedido	Número Peça
PK	PK
FK	FK
1	1
1	2
2	1

## 2.15. Atividades

Após o processo de entrevista com os usuários do departamento de compras da empresa Fabrica & Vende Roupas Ltda., o DA (Administrador de Dados) identificou os seguintes requisitos de dados para atender aos trabalhos daquele departamento:

O processo de compra pode ser iniciado por uma solicitação vinda do departamento de produto (matéria prima), departamento de serviços gerais (material de escritório, limpeza, segurança e afins), departamento de processamento de dados (suprimentos, acessórios e equipamentos de informática). Esta solicitação indica o produto desejado, quantidade, marca e modelo. A qualquer momento o gerente quer acesso imediato às seguintes informações: compras de um período qualquer separado por departamento solicitante, tipo de produto, valor mínimo máximo e solicitante, utilizando qualquer combinação dos argumentos citados.

Cada comprador precisa ter acesso a um catálogo de produtos com seus respectivos fornecedores. A partir disto o comprador irá montar um lista de pedido que terá os produtos solicitados, sua cotação anterior e atual com os fornecedores habituais e as condições negociadas para valor, quantidade e forma de pagamento. O pedido deve ser mantido como histórico, ficando pendente até que o fornecedor faça a entrega.

Cada produto feito para o departamento de compras pode indicar só o produto, sendo marca e modelo opcionais. Quando estas últimas não forem especificadas o comprador irá comprar o de menor preço.

A partir destas informações o DA identificou e desenhou as seguintes tabelas:

### 1 - DEPARTAMENTOS

PK	Código do Departamento
	Nome do Departamento
FK	Setor do Departamento

### 2 - SETORES

PK	Código do Setor
	Nome do Setor

### 3 - PRODUTOS

PK	Código do Produto
	Nome do Produto
FK	Tipo do Produto
FK	Marca do Produto
	Modelo do Produto

### 4 - TIPOS DE PRODUTOS

PK	Código do Tipo
	Nome do Tipo

### 5 - MARCAS

PK	Código da Marca
	Nome da Marca

## 6 - PEDIDOS DOS DEPARTAMENTOS

PK	Código do Pedido
FK	Código do Departamento
	Data da Solicitação
	Hora da Solicitação

## 7 - ITEM DOS PEDIDOS DOS DEPARTAMENTOS

PK (FK)	Código do Pedido
PK	Número do Item
FK	Código do Produto
	Marca do Produto
	Modelo do Produto
	Quantidade do Produto

## 8 - FORNECEDORES

PK	Código do Fornecedor
	Nome do Fornecedor
	Telefone de Contato

## 9 - PRODUTO POR FORNECEDOR

PK (FK)	Código do Produto
PK (FK)	Código do Fornecedor

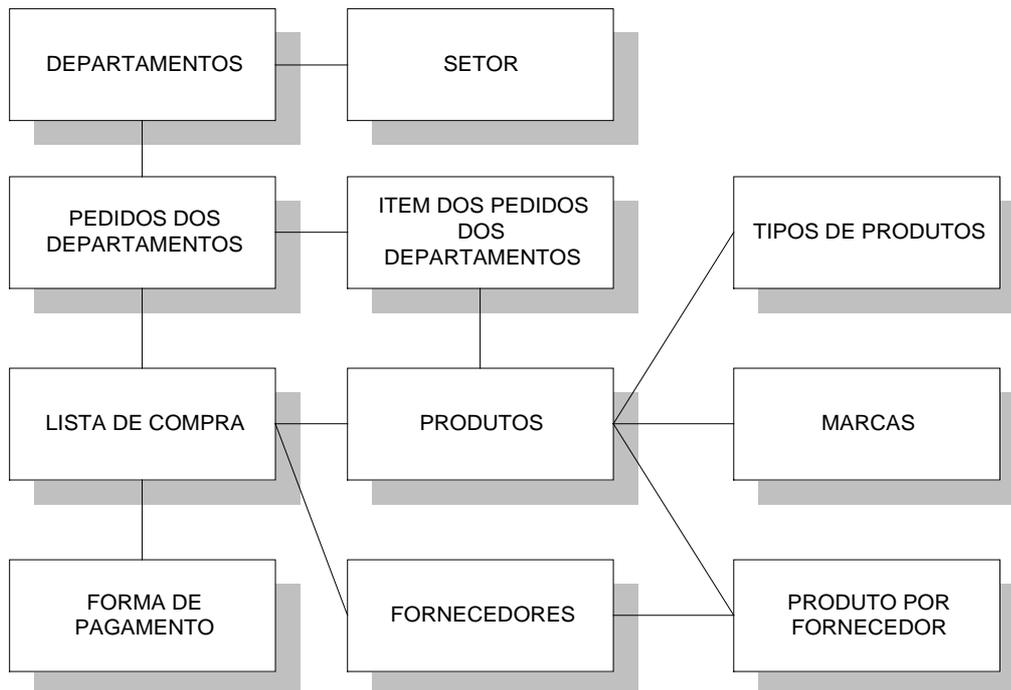
## 10 - LISTA DE COMPRA

PK (FK)	Código do Pedido
PK (FK)	Código do Produto
PK (FK)	Código do Fornecedor
	Quantidade Comprada
	Valor Total
FK	Forma de Pagamento

## 11 - FORMA DE PAGAMENTO

PK	Código da Forma
	Descrição da Forma

Em seguida o DA desenhou o diagrama da Figura 42 para visualizar melhor como fica o modelo de dados.



**Figura 42**

Cada traço no desenho define uma relação entre entidades.

Faça:

- Determine o tipo de cada uma destas relações.
- Escreva a condição determinante do relacionamento para cada relação apresentada.
- Crie Estas tabelas no Access, complementando com tipo de dado e tamanho de cada atributo das tabelas.
- Justifique sua escolha para os tipos e tamanhos utilizados.
- Implemente os relacionamentos no Access.
- Entre com dados diretamente nas tabelas criadas no Access.

# Access 2 e 2000

## 1. Introdução

O Access é um poderoso sistema de gerenciamento de banco de dados para o sistema operacional Windows.

O Access utiliza tabelas para armazenar os dados dos usuários. Portanto o primeiro passo a ser tomado é a criação de tabelas em seu banco de dados. Em seguida criar relacionamentos entre as tabelas e as planilhas de consultas de dados. Para interagir com o seu banco de dados utilizando uma interface gráfica mais aprimorada criamos os formulários. Para imprimir os dados do Banco de dados utilizamos os relatórios. Para funções que se repetem constantemente criamos Macros e código Visual Basic.

O Access permite que se construa um banco de dados para uma grande gama de atividades de controle de dados. Podemos utilizar o Access para criar desde um simples cadastro de clientes a um complexo controle de estoque, passando por aplicações de folha de pagamento, contas a pagar/receber, sistema de vendas, e não menos importante manipular dados na Internet com total interação com sua audiência (visitantes).

### 1.1. Banco de dados

Banco de dados é um elemento cuja finalidade é armazenagem de dados.

Um banco de dados possui uma série de informações relacionadas a um determinado assunto. Um exemplo, cadastro de clientes, onde está agrupado todos os dados dos clientes, nome, endereço, telefone, tudo isso num único lugar e de fácil consulta.

### 1.2. Campo

O campo é a menor unidade de informação existente em um arquivo de banco de dados. Isso significa que toda informação entrada em um arquivo de banco de dados é quebrada em porções menores, que recairão em campos específicos.

### 1.3. Registro

Um registro é um conjunto de campos. É a entidade que identifica uma entrada única em um banco de dados.

### 1.4. Índices

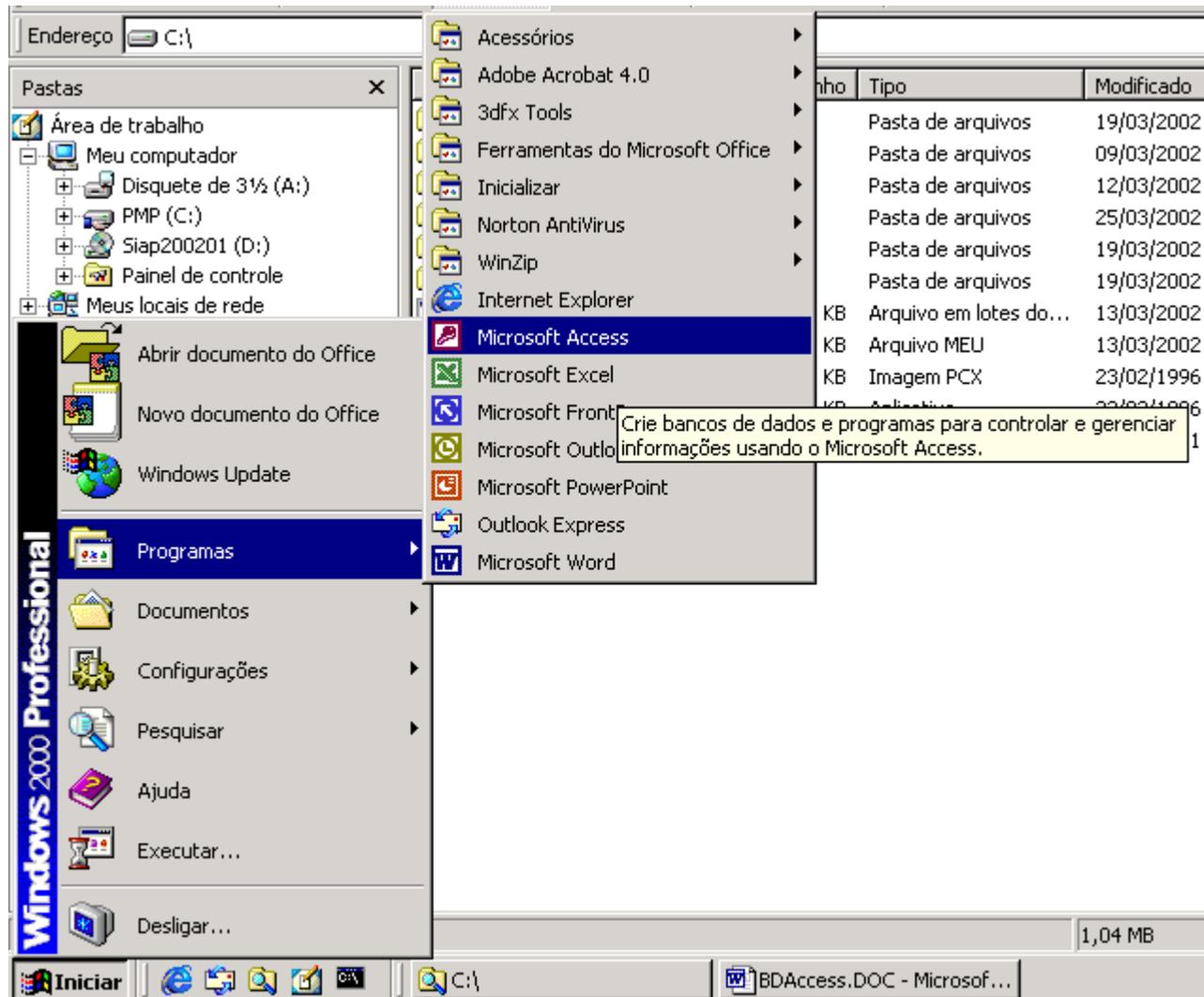
Os índices de um banco de dados permitem que as instruções sejam encontradas com uma maior rapidez. De certa forma, eles se assemelham aos índices encontrados nos livros, ou seja, direcionam até a posição onde se encontra a informação desejada.

### 1.5. Chave

Uma chave é uma entidade que permite a distinção dos registros de uma tabela. Uma chave pode ser formada por um campo ou um conjunto de campos. No último caso, elas são chamadas de chaves compostas. Uma chave primária determina necessariamente a ordem dos registros dentro da tabela.

## 2. Iniciando o Access 2000

- Para iniciar o access, clique no botão Iniciar.
- Posicione o cursor do mouse sobre Programas.
- Clique em Microsoft Access



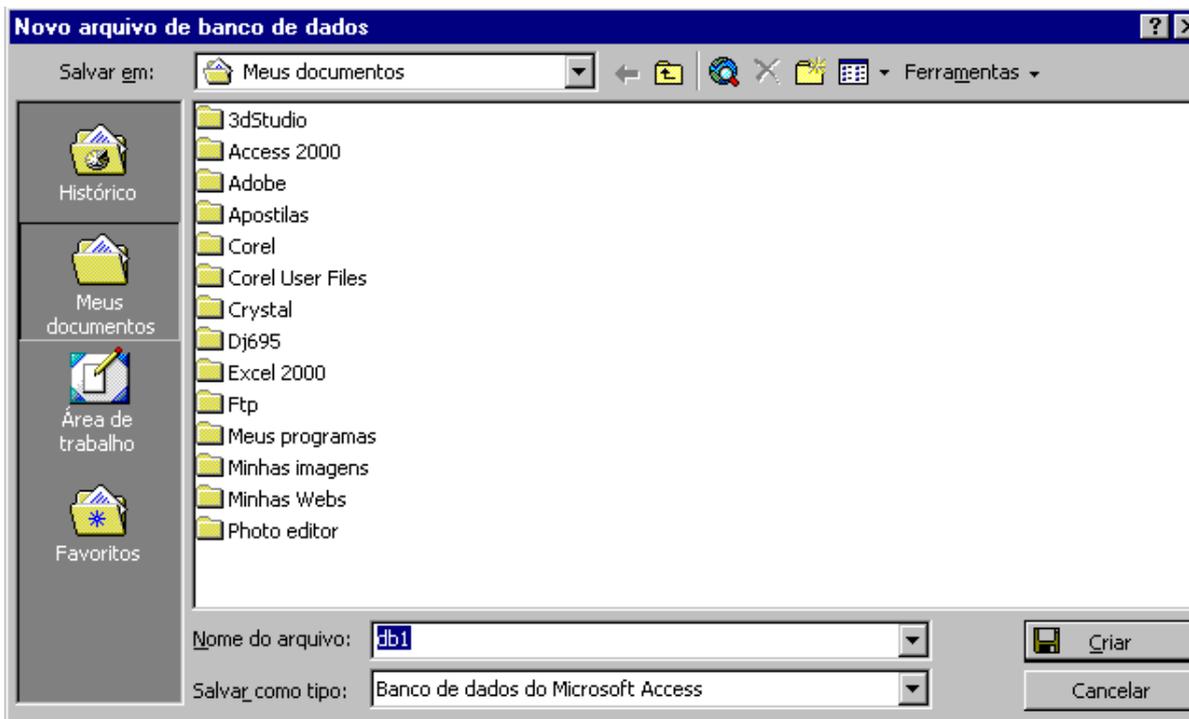
### 3. Criando um Banco de Dados

Para criar um banco de dados, utilize a opção “Banco de dados vazio do Access”.



Clique em “OK”.

Na próxima tela forneça um nome para o banco de dados e escolha a pasta onde seu banco de dados será armazenado.



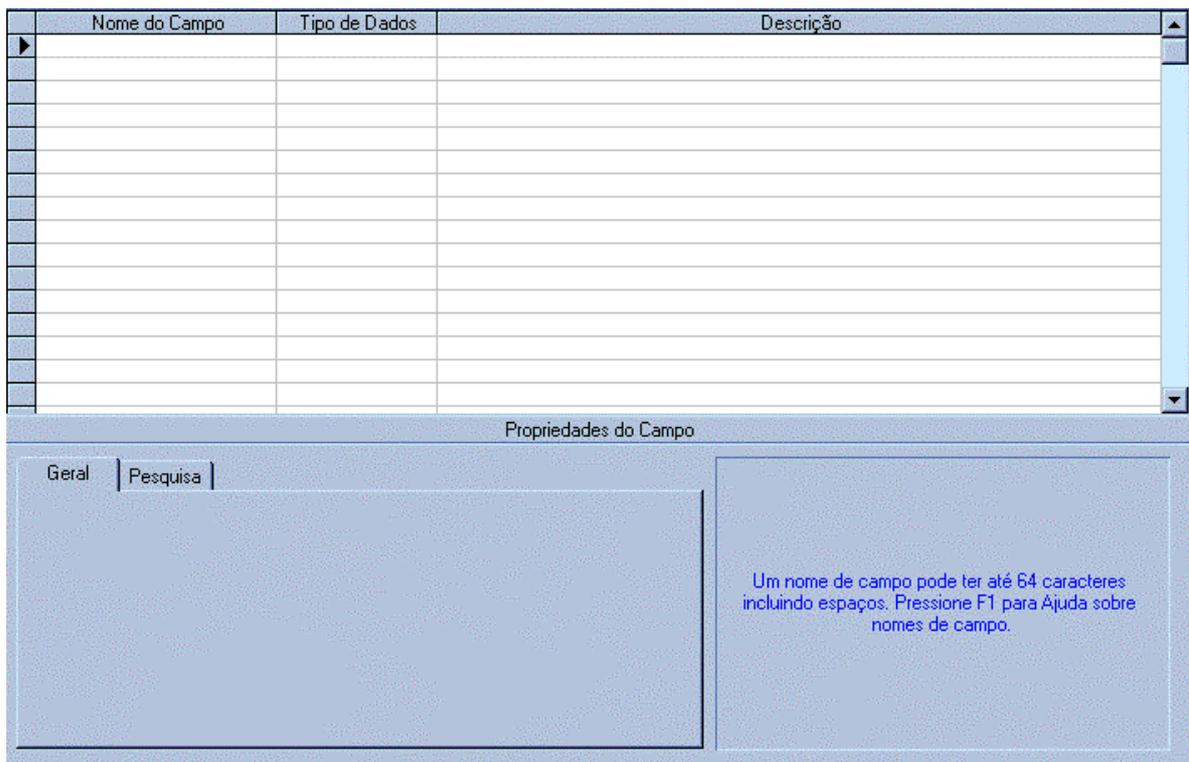
Clique em Criar, e seu banco de dados será criado. O próximo passo é criar uma tabela.

## 4. Tabelas

Para criar uma nova tabela de um clique duplo em "Criar tabela no modo estrutura"



A tabela possui três colunas



São elas:

**Nome do campo:** Identifica a natureza da informação. Ex: Nome, Endereço, Telefone, etc.

**Tipo de dados:** Identifica a característica do dado armazenado no campo. Ex: Texto, Data/Hora, Moeda, Número, etc.

**Descrição:** O preenchimento nesta coluna é opcional, porém sempre é bom digitar um texto explicativo nesta coluna, que será mostrado na "Barra de status" do formulário na hora do preenchimento dos dados.

Na tabela, o primeiro passo é definir a estrutura, informando quais serão os nomes dos campos, o tipo de dado que será digitado, e outras informações adicionais.

## 5. Nome do Campo

Para definir o nome do campo, utilize até 64 caracteres (letras e números), incluindo espaços. Não são permitidos sinais como ponto(.), exclamação(!) e colchetes([ ]).

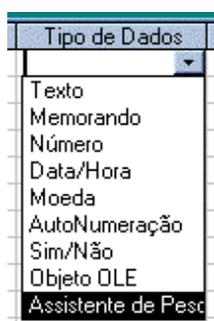
Exemplos de nomes de campo:

-  Nome do cliente: NomeCliente
-  Telefone comercial: FoneComercial
-  Data de nascimento: DtNascimento

## 6. Tipo de Dados

Após definir o Nome do campo, você deve informar o tipo de dados do mesmo. O tipo escolhido dependerá da natureza da informação que você deseja.

Veja a seguir, os tipos de dados permitidos.



### 6.1. Texto

Armazena caracteres alfanuméricos (texto e número). É o tipo indicado para armazenar informações que não requerem cálculo. Ex: nome de pessoas ou endereços.

### 6.2. Memorando

Armazena caracteres, no entanto, ele é utilizado para aguardar grandes blocos de texto que são divididos em parágrafos.

### 6.3. Número

Armazena valores numéricos inteiros ou fracionários. Este tipo deverá ser escolhido como padrão quando desejarmos efetuar cálculos com valores de campos.

### 6.4. Data/Hora

Faz com que o banco de dados trate os caracteres como datas ou horas, dependendo da forma como serão digitados.

### 6.5. Moeda

O tipo de dados Moeda tem todas características de um dado do tipo numérico, porém com acréscimo do símbolo monetário. Exemplo: "R\$"

### 6.6. AutoNumeração

Utilizado para contagem de registros. A cada registro inserido na tabela ele é automaticamente incrementado em um valor.

### 6.7. Sim/Não

Permite a entrada de valores do tipo Sim, Não, Verdadeiro ou Falso.

Qualquer outro tipo de valor que você digite no campo do tipo Sim/Não causará um erro, e o Access soará um beep alertando sobre o erro. Valores como Verdadeiro e Falso, o Access os converte para Sim ou Não automaticamente.

Quando digitar um valor "Não" certifique que digitou com o acento (til), caso contrário o Access tratará o valor como incorreto.

### 6.8. Objeto OLE

Esse tipo de dado é uma grande vantagem do ambiente gráfico Windows, pois você pode definir que o dado seja uma imagem gráfica. Uma das utilizações seria ter associada à ficha dos funcionários uma foto do mesmo.

### 6.9. Hyperlink

É um tipo de campo que irá armazenar um texto que permite saltar para um arquivo, um local em um arquivo, uma página HTML na Web.

Exemplo: <http://fp13vs23.digiweb.com>

### 6.10. Assistente de pesquisa

Cria um campo que permite que você escolha um valor a partir de uma tabela ou de uma lista de valores, utilizando uma caixa de combinação. A escolha dessa opção na lista de tipos de dados inicia um assistente para definir isso para você.

## 7. Propriedades do Campo

Ao definir o Tipo de dados, aparecerão na parte inferior da janela as propriedades do campo. Você deverá definir as propriedades específicas para cada tipo de dados.

### 7.1. Tamanho do campo

Define o tamanho máximo de caracteres que será permitido em um campo.

**Texto:** Pode ser entre 1 a 255 caracteres.

**Número:**

Definição	Descrição
Byte	Números de 1 a 255 (não fracionários)
Inteiro	Números de -32.768 a 32.767 (não fracionários)
Inteiro Longo	Números de -2.147.483.648 a 2.147.483.647 (não fracionários)
Simplex	Números com seis dígitos de precisão podendo valer até $3,402823 \times 10^{38}$
Duplo	Números com 10 dígitos de precisão, de $-1,79769313486232 \times 10^{308}$ até $1,79769313486232 \times 10^{308}$

## 8. Formato

Especifica o formato para exibição de números, datas, horas e texto.

### 8.1. Texto e Memorando

Você pode utilizar símbolos para especificar se caractere de texto é permitido e se todos os caracteres são exibidos em maiúsculas ou minúsculas.

Símbolo	Descrição
@	Um caractere de texto é requerido
&	Um caractere de texto não é requerido
<	Transforma todos os caracteres em letras minúsculas
>	Transforma todos os caracteres em letras maiúsculas

Exemplos:

Símbolo	Dados	Exibição
@@@-@@-@@@@	123456789	123-45-6789
@@@@@@@@@@	123-45-6789	123-45-6789
	123456789	123456789
>	Leonardo	LEONARDO
>	LEONARDO	LEONARDO
>	Leonardo	LEONARDO
<	Leonardo	leonardo
<	LEONARDO	leonardo
<	Leonardo	leonardo

Número e moeda:

Formato	Descrição
Número geral	(Padrão) Exibe o número exatamente como foi inserido
Moeda	Utiliza p separador de milhares; exibe números negativos entre parênteses.
Fixo	Exibe pelo menos um dígito
Standard	Utiliza o separador de milhares
Por cento	Multiplica o valor por 100; anexa um sinal de porcentagem.
Científico	Utiliza a notação científica padrão

Exemplos:

Formato	Número digitado	Número formatado
Número geral	1234567	1234,567
	-1234,567	-1234,567
	\$123,45	\$123,45
Moeda	1234,789	\$1.234,79
	-1234,789	-1234,79
	3,56645	3,57
Standard	1234,789	1.234,79
Por cento	2	200%
	0,35	35%
	3456,789	3,46E+03

	-3456,789	-3,46E+03
--	-----------	-----------

**Data/Hora:**

Formato	Descrição
Data geral	(Padrão) Se o valor for somente data, a hora não será exibida; se o valor for somente hora, a data não será exibida. Exemplos: 3/3/99 07:45:00, 4/4/99 e 06:55:00
Data completa	Exibe a data por extenso. Exemplo: Domingo, 2 de abril de 2000
Data média	Exemplo: 6-abr-99
Data abreviada	Exemplo: 4/4/99
Hora completa	Exemplo: 8:22:21
Hora média	Exemplo: 09:15
Hora abreviada	Exemplo: 21:30

**Sim/Não:**

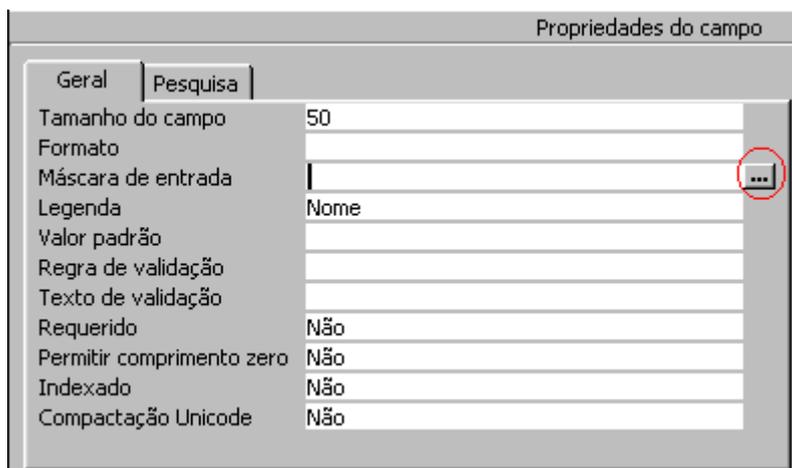
Formato	Descrição
Sim/Não	(Padrão) Sim=-1, Não=0
Verdadeiro/Falso	Verdadeiro=-1, Falso=0
Ativado/Desativado	Ativado=-1, Desativado=0

## 9. Máscara de Entrada

É um formato que você especifica para controlar a entrada de dados no campo. Uma máscara de entrada consiste em caracteres exibidos (tais como parênteses, pontos ou traços) e caracteres de máscara que especificam onde os dados devem entrar, que tipo de dados e quantos caracteres são permitidos. Para criar uma máscara de entrada, selecione a tabela desejada e clique no botão Estrutura.



Em seguida clique no botão para construir na caixa de propriedade Máscara de entrada.



### 9.1. Casas decimais

Determina o número de casas decimais que o Access utilizará para exibir os números.

### 9.2. Legenda

É o título do campo quando usado em um formulário. As legendas são tipicamente utilizadas para fornecer informações ao usuário. Se você não definir uma legenda, o nome do campo é usado como título.

### 9.3. Valor padrão

É uma maneira para se garantir que os dados inseridos no banco de dados estão corretos. Você especifica regras com uma expressão. Por exemplo, para um campo numérico aceitar apenas valores acima ou igual a 50, utiliza a seguinte regra de validação:  $\geq 50$ . Ao utilizar regra de validação, você deverá:

-  Usar aspas em expressões que contenham texto.
-  Usar o sinal "#" em expressões que contenham data.

Utiliza os seguintes operadores:

>	Maior
<	Menor
=	Igual
>=	Maior ou igual
<=	Menor ou igual
<>	Diferente

Exemplos:

Regra de validação	Texto de validação
<>0	Insira um valor diferente de zero
Em("SP", "MG")	Estado deve ser SP ou MG
>=Data()	Nenhuma da ta antes de hoje
Entre 1 e 5	Valores entre 1 e 5
Como "L??"	3 caracteres começando com L
<#3/3/99#	Apenas datas antes 3/3/1999

Você também pode criar uma regra de validação que compare os valores em dois campos. Por exemplo, você pode querer ter certeza de que a data de entrega para envio de um pedido será dentro de 15 dias a partir da data do pedido, exemplo:

[Data de entrega] <= [Data do pedido] + 15

### 9.4. Texto de validação

É a mensagem de erro que aparece quando o valor é proibido pela regra de validação.

### 9.5. Requerido

Especifica se um valor é ou não requerido em um campo. Se esta propriedade é definida como "Sim", você é obrigado a digitar um valor no campo.

Definição	Descrição
Sim	O campo requer um valor
Não	(Padrão) O campo não requer um valor

### 9.6. Permitir comprimento zero

Permite definir se será aceita uma entrada de dados em branco (nenhum caractere). Para entrar com uma sequência vazia em um campo, digite duas aspas (" ").

### 9.7. Indexado

O índice é um recurso do Access que aumenta a velocidade da busca e classificação em uma tabela. Pode-se criar tantos índices quantos forem desejados. Os índices são criados quando você salva a estrutura da tabela são automaticamente atualizados quando você adiciona ou altera registros.

Definição	Descrição
Não	(Padrão) Sem índice
Sim (Duplicação autorizada)	O índice permite valores duplicados
Sim (Duplicação não autorizada)	O índice não permite duplicação, ou seja, os valores serão únicos.

### 9.8. Compactação Unicode

Um esquema de codificação que armazene cada caractere em um byte limita você a uma única página de código — um conjunto numerado contendo no máximo 256 caracteres. Entretanto, o Unicode pode oferecer suporte à no máximo 65.536 caracteres, porque representa cada caractere como dois bytes. Como resultado, os dados em um campo **Texto**, **Memorando** ou **Hyperlink** exigem mais espaço de armazenamento do que nas versões anteriores do Access.

## 10. Chave Primária

Após definir os nomes e propriedades dos campos, você deve definir a chave primária, que é uma marca única que tem o objetivo de identificar cada registro em uma tabela. Toda tabela deve possuir uma chave primária.

Definindo uma chave primária em um campo, o Access atribuirá as seguintes características à sua tabela:



Um índice será criado automaticamente com base na chave primária, acelerando assim as operações de consultas, pesquisas, etc.

- Os dados serão mostrados na folha de dados na ordem da chave primária.
- Não será permitidos a entrada de registros com dados duplicados nos campos que definem a chave primária.

## 11. Definindo uma Chave Primária

Para definir uma chave primária, tome o seguinte procedimento:

- 1) Selecione a tabela desejada e clique no botão ( Estrutura).
- 2) Clique no campo que deseja como chave primária.

	Nome do campo	Tipo de dados	Descrição
▶	Nome	Texto	
	E-mail	Texto	E-mail do cliente

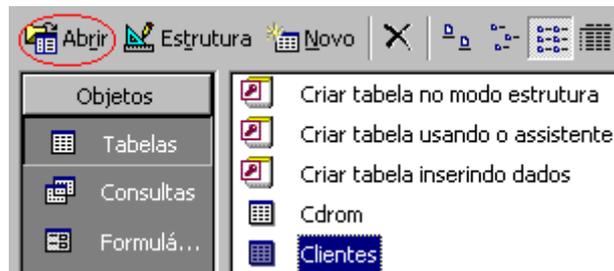
- 3) Clique no botão ( Chave primária). Um símbolo aparece no lado esquerdo do campo chave primária.

	Nome do campo	Tipo de dados	Descrição
	Nome	Texto	
	E-mail	Texto	E-mail do cliente

- 4) Clique no botão ( Salvar para salvar as alterações.

## 12. Inserindo Dados na Tabela

Selecione a tabela que deseja inserir dados. Clique no botão ( Abrir) Abrir.



O modo Folha de dados é semelhante a uma planilha eletrônica. Os nomes dos campos aparecem em um cabeçalho.

Nome	Endereço	Cidade	CEP	Produto/Compra
------	----------	--------	-----	----------------

Digite o dado correspondente no seu campo de origem. Para mudar de campo, clique com o mouse no campo desejado, ou pressione a tecla [TAB].

Inicialmente, as larguras das colunas obedecem a um tamanho padrão. Para alterar a largura de uma coluna, posicione o ponteiro do mouse no cabeçalho das colunas, na linha que separa o nome dos campos. O Ponteiro se transforma em uma seta de duas pontas; arraste-o em seguida.

Endereço	Cidade
Caixa Postal 10507,	Niterói
Rua 24, Quadra 13,	São Luís
Rua João 25, Casa	Minas Gerais

Ao pressionar a tecla [TAB] para passar ao próximo campo, aparece o símbolo de um lápis indicando que este registro ainda não foi salvo.

 Leonardo

O registro será salvo após digitar os dados do último campo. O pressionar a tecla [TAB], aparece um triângulo indicando que aquele registro foi salvo.

	Leonardo da Sil	Rua João 25, Casa	Minas Gerais	66666-000	Livro Segurança na Inter
---	-----------------	-------------------	--------------	-----------	--------------------------

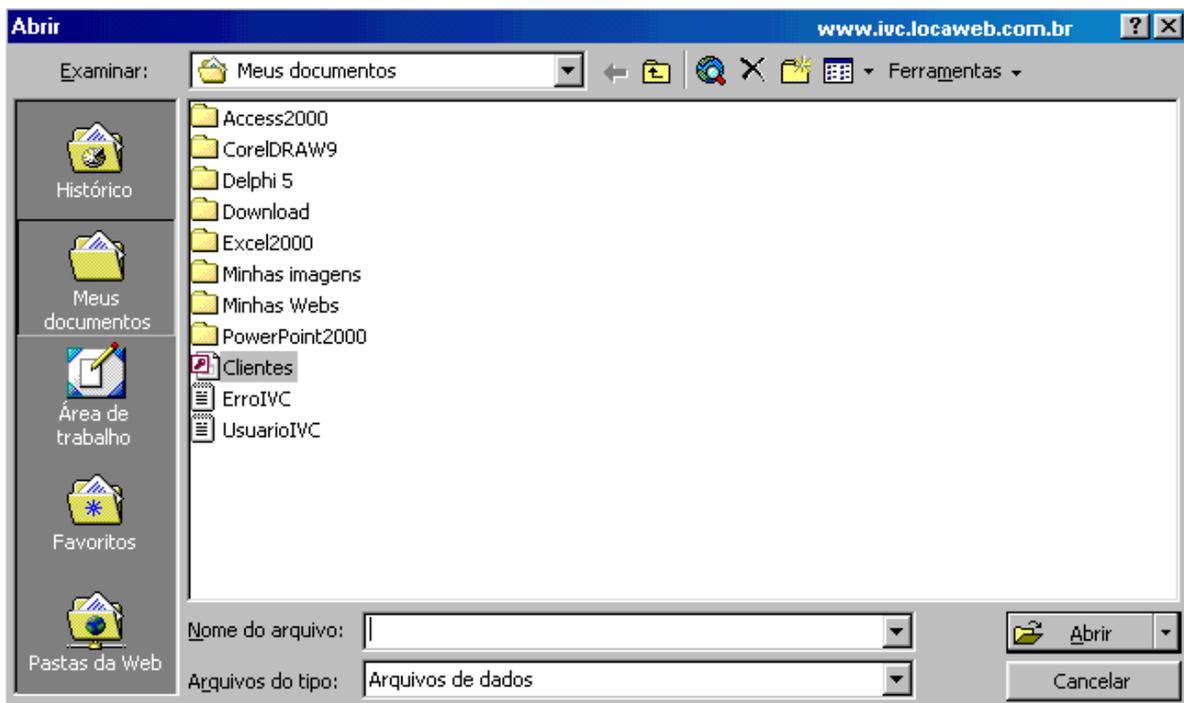
Se você não quiser salvar o registro atual, clique no botão  Desfazer. Um clique neste botão desfaz a última ação, se clicar mais de uma vez, outras ações serão desfeitas.

Para sair do banco de dados, clique no menu Arquivo/Fechar.

Para finalizar o trabalho no Access clique no menu Arquivo/Sair.

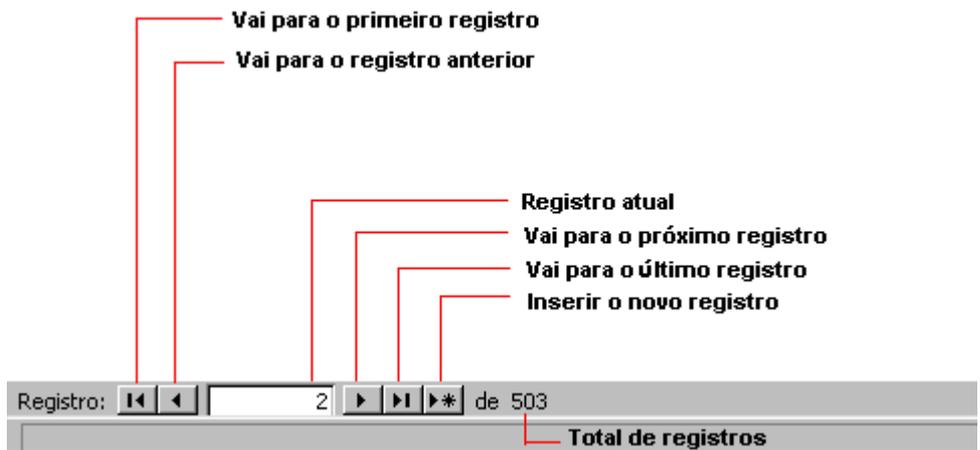
## 13. Editando a Tabela de Dados

Para abrir o banco de dados, clique em Arquivo/Abrir.



Abrindo uma tabela de dados

Para abrir uma tabela de dados, selecione a tabela desejada, clique no botão  Abrir. Para movimentar-se entre os dados da tabela, utilize os botões de rolagem.



## 14. Classificação Crescente Decrescente

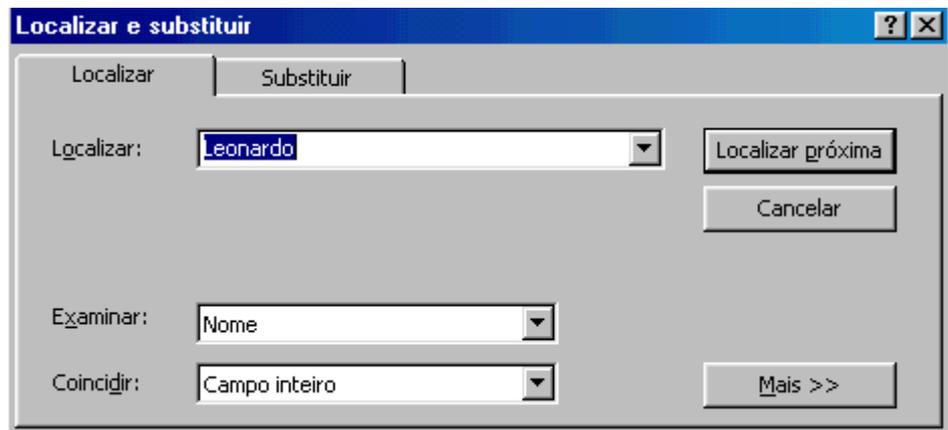
Para classificar a sua tabela, clique no campo que deverá comandar a ordem de classificação. Depois, clique em um dos botões de classificação.

Você poderá escolher entre Crescente (A, B, C, D,...) ou Decrescente (... , D, C, B, A).



## 15. Localizar Registros

Para localizar um determinado registro em sua tabela clique no botão (🔍) Localizar. Um caixa de diálogo aparecerá.



Em **Localizar**, digite a palavra que deseja localizar na tabela.

Em **Examinar**, selecione o campo.

Em **Coincidir**, selecione a opção **Campo inteiro**.

Após digitar a palavra que deseja localizar, clique no botão para Localizar.

Para ter mais opções de procura clique no botão Mais.

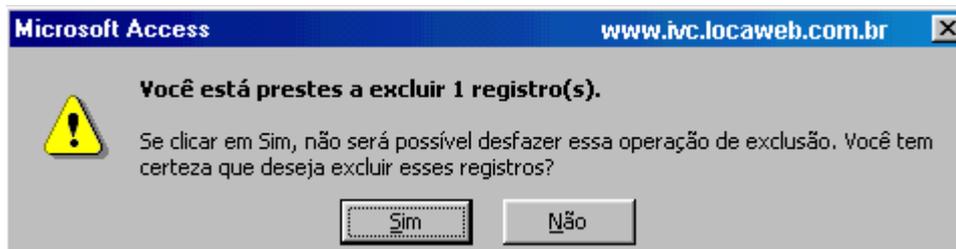
## 16. Excluir Registros

Para excluir um registro da tabela, clique no lado esquerdo da janela para poder selecioná-lo.



Em seguida clique no botão (  ) Excluir registro.

Esta ação (Excluir registro) pede a confirmação da exclusão.



Clicando no botão Sim, não será mais possível recuperar aquele registro.

Clicando no botão Não, o registro permanece na tabela.

## 17. Excluir Campos

Para excluir um campo da tabela, você deve selecionar a tabela desejada, clicar no botão (  ) Estrutura.

Ao excluir um campo de uma tabela todos os dados referentes a este campo serão perdidos.

Após selecionar a tabela desejada e entrar no modo estrutura, clique no lado esquerdo do campo que deseja excluir. Clique no botão (  ) Excluir linhas.



Uma mensagem de alerta irá aparecer.

Clicando no botão Sim, você estará dando um fim no campo selecionado.

Clicando no botão Não, o campo permanece na tabela.

## 18. Inserir Campos

Selecione a tabela desejada e clique no botão (  ) Estrutura.

Clique no lado esquerdo do nome do campo, na posição onde deseja inseri-lo.

Em seguida, clique no botão (  ) Inserir linhas.



Nome do campo	Tipo de dados
CódigoCliente	Número
Cliente	Texto
Contato	Texto
Sobrenome	Texto
Departamento	Texto
Endereço	Texto
Cidade	Texto

Após clicar no botão (inserir linhas), aparecerá uma linha em branco onde você deverá preencher com os dados sobre o novo campo.

Para salvar as alterações da tabela, clique no botão (  ) Salvar.

## 19. Consultas

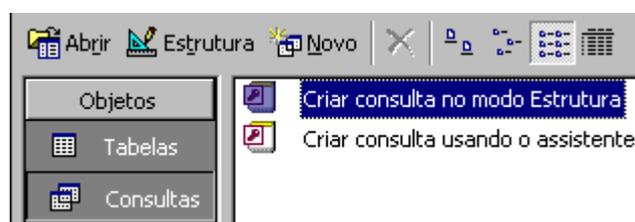
A consulta é uma maneira predefinida e personalizada de se pesquisar os dados da usa tabela. Por exemplo, em um banco de dados Clientes, você pode querer consultar todas as pessoas da cidade "São Paulo". Ou quem sabe consultar quantos clientes você conseguiu nos últimos trinta dias.

Ao invés de se estabelecer os critérios toda vez que você desejar localizar esses registros, você pode montar uma consulta, e então bastará abri-la.

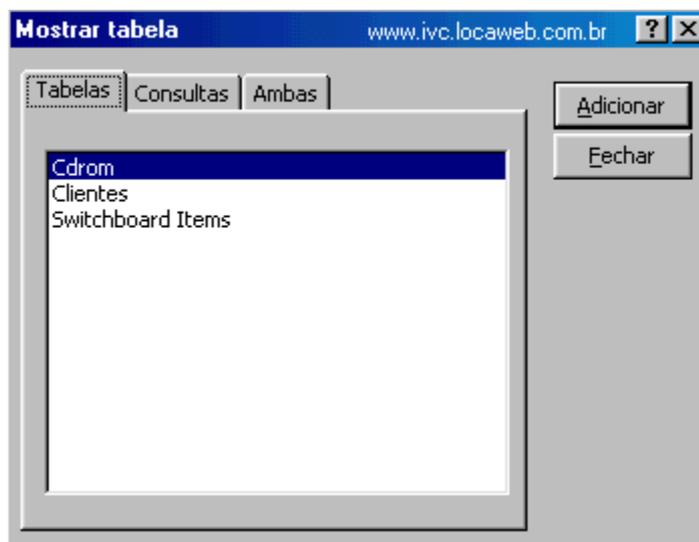
## 20. Criar Uma Consulta

Para criar uma nova consulta, dê um clique sobre o item Consultas na janela Banco de Dados.

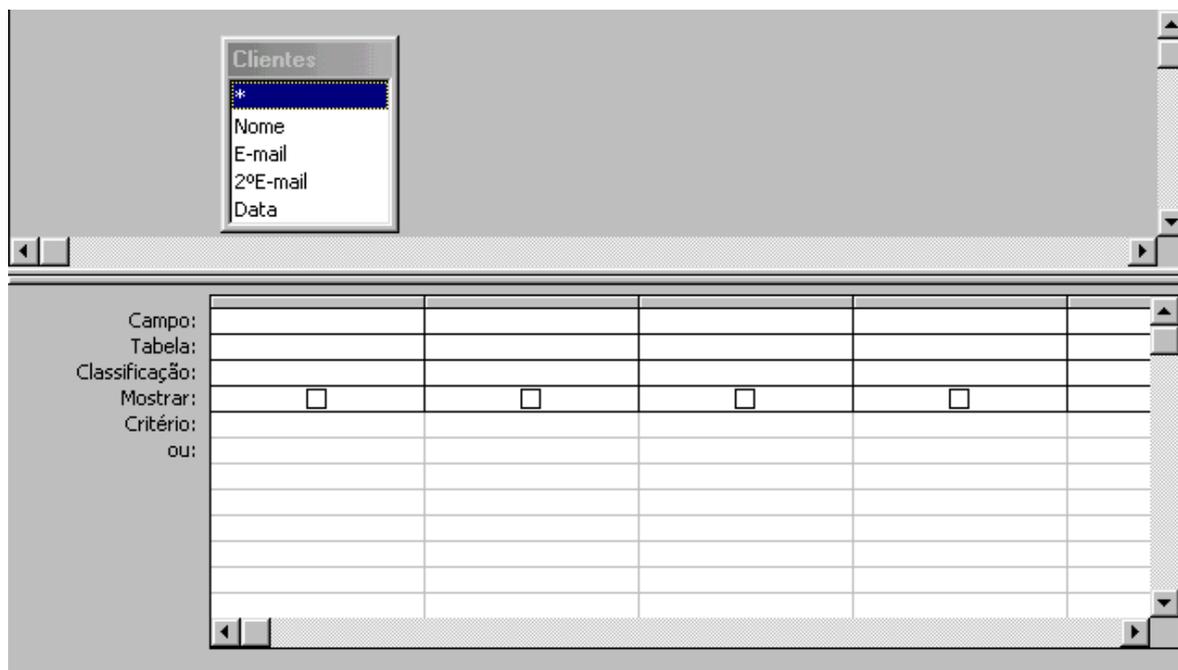
Dê um clique duplo em Criar consulta no modo estrutura.



Aparecerá a caixa de diálogo **Mostrar tabela** com todas tabelas do seu banco de dados. Selecione a tabela deseja e clique no botão Adicionar. Em seguida clique no botão Fechar.



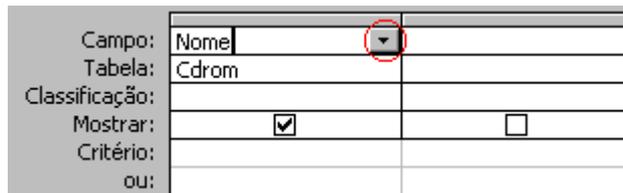
A janela de consultas contém duas partes. A parte superior contém as tabelas que você seleciona para fazerem parte da consulta, e a parte inferior define os critérios de busca das informações desejadas, bem como os campos que farão parte da consulta.



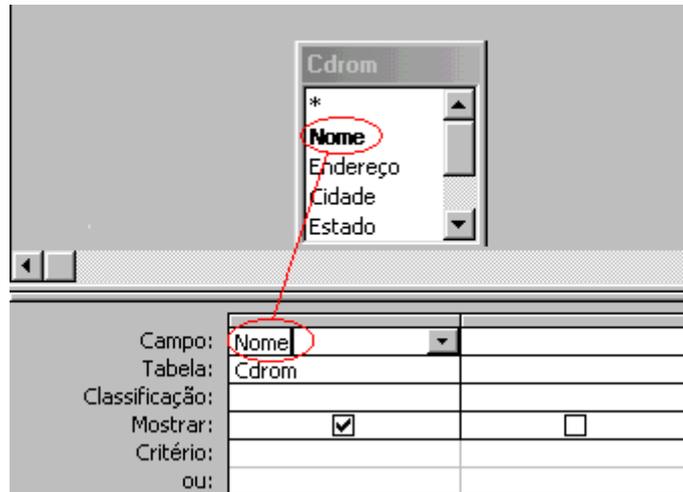
## 21. Seleção dos Campos

O primeiro passo na criação de uma consulta é a seleção dos campos que farão parte da pesquisa.

Para selecionar um campo, clique na grade Campo (parte inferior). Selecione o campo desejado clicando na seta que aparece ao lado da grade.



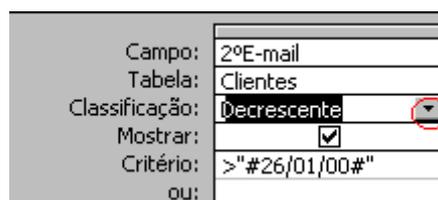
Você também pode selecionar um Campo arrastando o campo da tabela (parte superior) até a grade **Campo** (parte inferior).



Para excluir uma coluna da grade inferior, clique sobre a linha da coluna desejada e em seguida, clique em Editar/Excluir colunas.

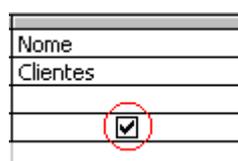
## 22. Classificando a Consulta

Você poderá configurar a consulta para exibir os dados de uma forma classificada. Clique na seta e escolha a ordem de classificação.



## 23. Exibição de Campos

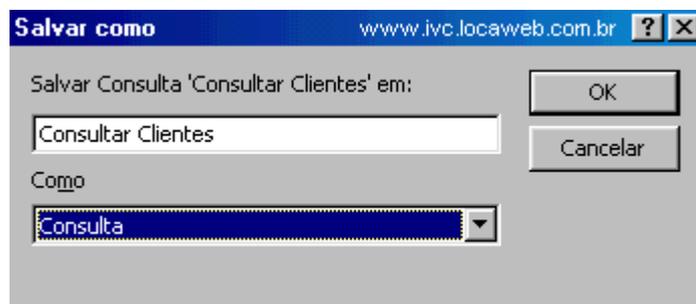
Na linha **Mostrar** da grade inferior, você nota a presença de um quadrado com uma marcação, indicando que este campo será mostrado no resultado da pesquisa. Se você retirar a marcação do quadrado, o campo não será exibido na consulta.



## 24. Salvar a Consulta

Para salvar a consulta, utilize o comando Arquivo/Salvar. Em seguida, digite o nome para a consulta e clique em OK para salvar a consulta.

Você pode salvar a pesquisa como Formulário, Consulta ou Relatório, basta clicar na seta para baixo e escolher a opção desejada.



Para fechar a consulta, clique no menu Arquivo/Fechar.

Para abrir uma pesquisa existente, selecione a consulta desejada no item Consultas, em seguida clique no botão (  Abrir ) Abrir.

Para fazer alterações em uma consulta, selecione a consulta desejada no item Consultas, em seguida clique no botão (  Estrutura ) Estrutura.

## 25. Critérios de Pesquisa

Para especificar um critério de pesquisa, clique sobre a linha Critério (Grade inferior). Digite uma expressão na coluna do campo desejado para o critério.

Utilize os seguintes operadores para construir um critério:

>	Maior
<	Menor
=	Igual
>=	Maior ou igual
<=	Menor ou igual
<>	Diferente

### Regras:

Use aspas em expressões que contenham texto.

Use o sinal “#” em expressões que contenham data.

### Exemplos:

Critério	Resultado
<>0	Diferente de zero
Em("MG,TO")	Estado deve ser MG ou TO
>=Data	Nenhuma data antes de hoje
Entre 2 e 5	Valores entre 2 e 5
Como "L??"	2 caracteres começando com L
<#1/1/99#	Apenas datas antes de 1999

L*	Começando com a letra "L"
>#17/04/99	Data após 17/04/99
"Avenida"	Palavras iguais à "Avenida".
30	Números iguais a "30"

## 26. Consultando Várias Tabelas

Você pode utilizar mais de uma tabela para criar uma consulta. Para adicionar uma nova tabela a sua consulta, clique em **Consulta/Mostrar tabela** ou clique no botão (  ) **Mostrar tabela**.



Dê um clique na tabela desejada para poder selecioná-la, e clique no botão **Adicionar**. Após adicionar todas as tabelas que gostaria que fizesse parte da consulta clique no botão **Fechar**.

O Access estabelece uma conexão entre ambas tabelas através de uma linha ligando os campos em comum.

Você pode adicionar campos na grade inferior (QBE) provenientes das duas tabelas.



### 27. SQL

A SQL (Linguagem de Consulta Estruturada) surgiu com a terceira geração de banco de dados, os RDBs-Relational Databases (Banco de dados relacionais).

SQL é na verdade uma linguagem para definir e manipular bancos de dados relacionais. Praticamente todos os produtos de banco de dados relacionais que estão no mercado suportam SQL, e com o Microsoft Access você poderá criar critérios com base na linguagem SQL.

### 28. Declarações SQL

A declaração SQL pode ser feita de duas formas:

1ª) No campo Critério digitando a declaração em português.

Exemplo:

Entre #01/01/00# E #01/05/00#

2ª) No Modo SQL. Quando estiver criando uma consulta no modo Estrutura, clique no menu Exibir/Modo SQL.

Exemplo:

```
SELECT Clientes.Nome, Clientes.[E-mail], Clientes.Data
```

```
FROM Clientes
```

```
WHERE (((Clientes.Data) Between #4/15/2000# And #5/1/2000#))
```

```
ORDER BY Clientes.Nome, Clientes.[E-mail] DESC;
```

#### Instrução SQL



**Figura 29**

- O objetivo do sistema relacional é oferecer um banco de dados de fácil compreensão e de definição clara, o que facilita sua manipulação.
- Seu princípio teórico é identificar precisamente *o que pode ou não ser feito*.
- O sistema relacional é previsível e disciplinado, sempre se sabe o que vai fazer.
- Os operadores (comandos para manipulação do banco e seu conteúdo) são de alto nível e de fácil aprendizado e entendimento.
- Querendo efetuar mudanças nas definições dos bancos de dados, pode-se inserir novas colunas às tabelas, sem ter que alterar qualquer coisa nos programas de aplicação.
- As operações relacionais defendem uma mudança radical. É como sair de um Fusca e ir entrar numa Mercedes.

A manipulação dessas tabelas é feita através de operadores, por meio dos quais se pode fazer qualquer tipo de acesso a uma tabela ou conjunto delas (projeção, seleção, junção etc.). Veja na Figura 31 um exemplo de junção.

```

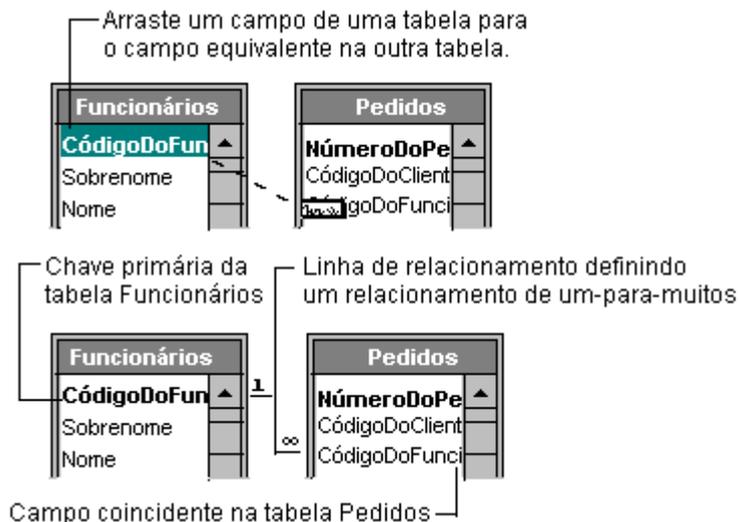
SELECT Fornecedor.[Nome de Fornecedor],
        Fornecedor.[Situação do Fornecedor],
        Pedido.[Código da Mercadoria],
        Pedido.[Quantidade Pedida]
FROM Fornecedor INNER JOIN Pedido ON
        Fornecedor.[Código do Fornecedor] = Pedido.[Código do Fornecedor];
    
```

**Figura 31**

Esta é uma sentença SQL que seleciona registros específicos das tabelas Fornecedor e Pedido, juntando todos os registros onde Código do Fornecedor na tabela Fornecedor seja igual ao Código do Fornecedor da tabela Pedido. Este é um processo denominado JOIN (junção ou união).

## 29. Relacionamentos

Você define um relacionamento adicionando as tabelas que deseja relacionar com a janela (  ) Relacionamentos e, em seguida, arrastando o campo chave de uma tabela e soltando-o no campo chave da outra tabela.



A espécie de relacionamento que o Microsoft Access cria depende de como os campos relacionados são definidos:

-  Um relacionamento um-para-muitos é criado quando somente um dos campos relacionados é uma chave primária ou tem um índice exclusivo.
-  Um relacionamento um-para-um é criado quando ambos os campos relacionados são chaves primárias ou têm índices exclusivos.
-  Na verdade, um relacionamento muitos-para-muitos são dois relacionamentos um-para-muitos com uma terceira tabela cuja chave primária consiste em dois campos — as chaves externas das outras duas tabelas.

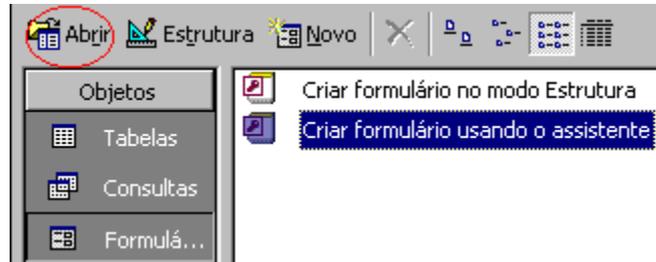
## 30. Formulários

O formulário é uma excelente interface para manipular as informações do banco de dados. O formulário contém recursos gráficos que permite filtrar e modificar os dados.

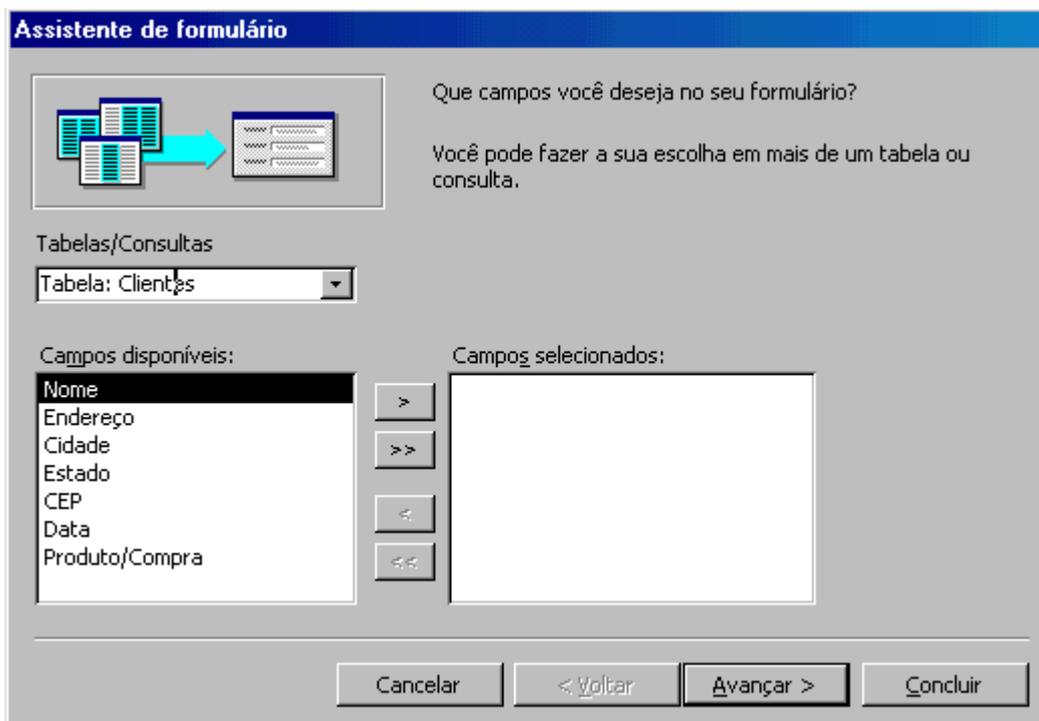
É através dos formulários que podemos incluir códigos ou chamadas que executem macros funções específicas.

### 31. Criando Um Formulário

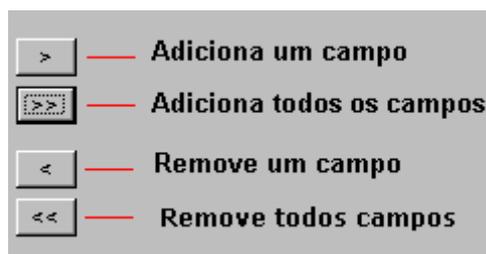
Para criar um formulário, clique na guia Formulário. Selecione “Criar formulário usando o assistente”, em seguida clique no botão (  Abrir ).



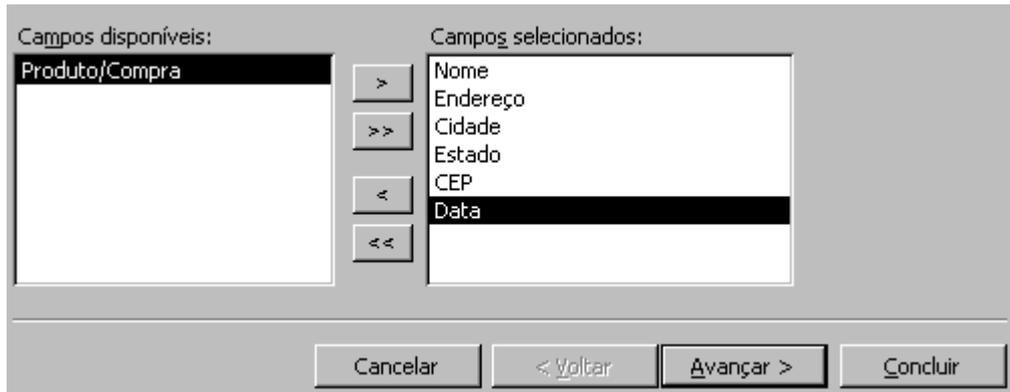
Aparecerá a tela do Assistente de formulário. Onde você deve escolher a tabela ou consulta de origem de dados e os campos que farão parte de seu formulário.



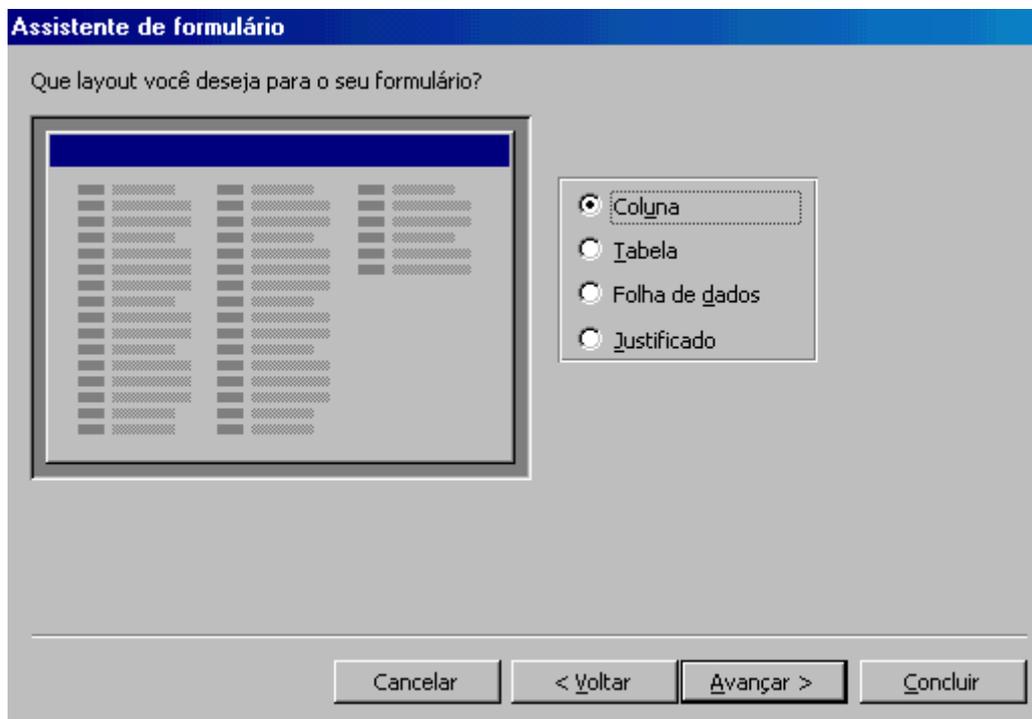
Na janela Assistente de Formulário, existem duas caixas de opção: a da esquerda exibe os campos disponíveis, e a da direita os campos escolhidos. Repare que existem botões no centro da janela Assistente de formulário que servem para movimentação dos campos.



Selecione os campos disponíveis e clicar no botão avançar.

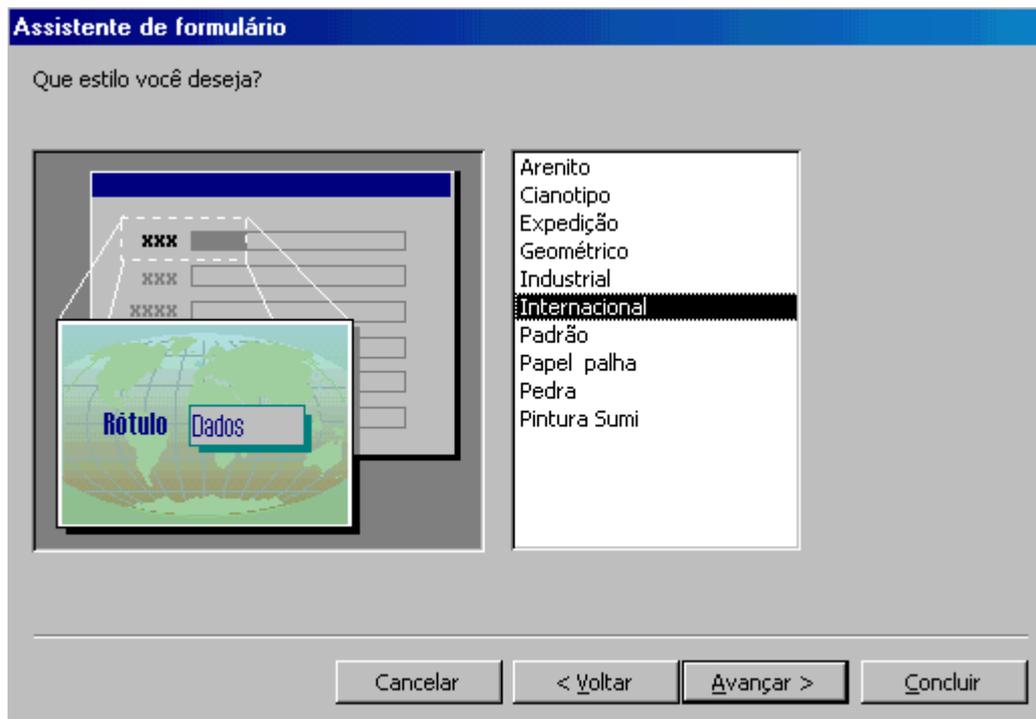


Na próxima tela você deverá definir o layout do formulário. Escolhendo a opção "Coluna" (default) o formulário apresentará um registro por vez.



Clique no botão avançar para aceitar a opção default.

No próximo passo, defina o estilo que seu formulário adotará. Após decidir a opção que irá utilizar, clique no botão avançar.



Na próxima etapa informe o título para o formulário. Clique no botão Concluir e seu formulário estará pronto.



Agora você pode visualizar os registros de uma forma gráfica, e poderá incluir novos dados em seu formulário.

Movimente-se em seu formulário utilizando as os botões para movimentar-se entre os registros.



Para incluir um novo registro na tabela de dados, clique sobre o botão (➤\*) novo registro.

## 32. Estrutura do Formulário

Você pode alterar a estrutura do formulário afim de:

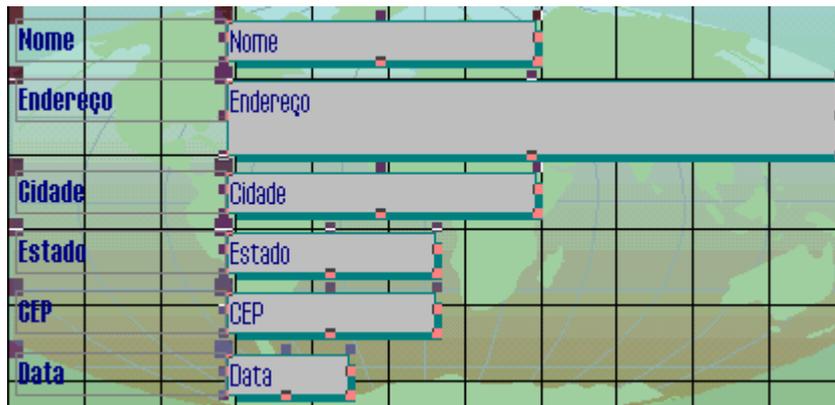
-  Alterar a largura das caixas dos campos do formulário.
-  Alterar o texto do formulário.
-  Modificar o posicionamento do texto.
-  Cor do fundo da tela.

Selecione o formulário desejado na guia Formulário. Clique no botão ( Estrutura) Estrutura.

Ao clicar em um controle, este ficará selecionado. Podendo ter seu tamanho, cor, formato modificado.



Você pode selecionar vários controles, arrastando o ponteiro do mouse ao redor deles. Após soltar o botão do mouse todos os controles estarão selecionados.



Os controles podem ser selecionados, mantendo a tecla SHIFT pressionada, enquanto clica sobre os controles.

### 33. Alterar Aparência do Texto

Para alterar a aparência de um controle, você deve primeiro selecioná-lo clicando nele uma vez.



Agora selecione uma (  ) Fonte para o texto.

O (  ) Tamanho da fonte.

A (  ) Formatação (negrito, itálico ou sublinhado).

O (  ) Alinhamento do texto (esquerdo, centro, direito).

A (  ) Cor de fundo/Preenchimento.

A (  ) Cor do primeiro plano/fonte.

A (  ) Cor da linha/borda.

A (  ) Largura da linha/borda.

O (  ) Plano.

Os resultados são aplicados imediatamente quando se tem um controle selecionado e se clica no botão correspondente ao efeito desejado.

O resultado final pode ser visto mesmo antes de aceitar as alterações feitas na aparência.



Se você não gostar das alterações, clique no botão (↶) Desfazer.

Para aceitar as mudanças clique no botão (💾) Salvar.

### 34. Rótulo do Formulário

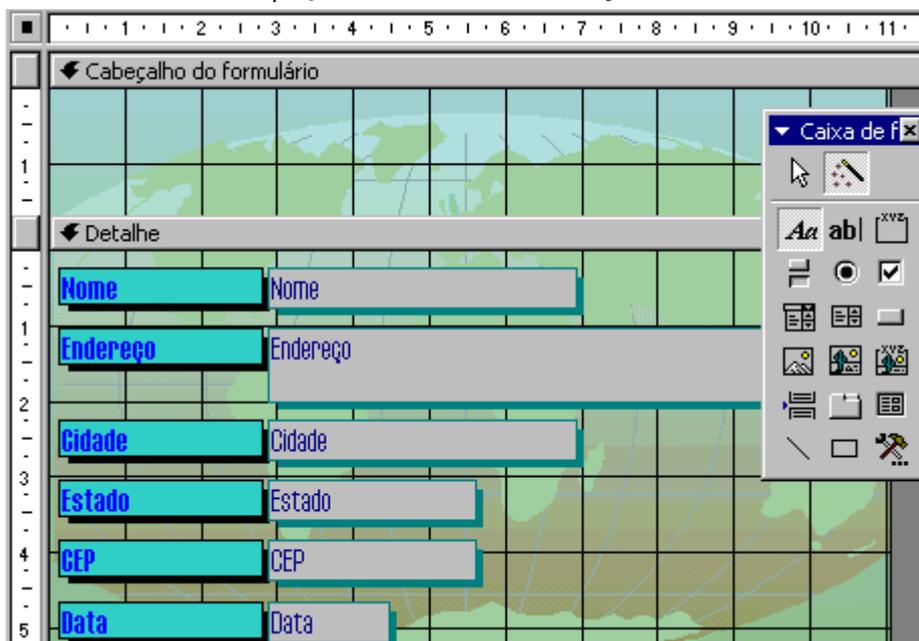
Você pode definir um rótulo para seu formulário separando o **Cabeçalho do formulário**, e o **Detalhe** dos campos e controles.



Após separar o **Cabeçalho do formulário** do **Detalhe** clique no botão (🔧) Caixa de ferramentas.

Na pequena Caixa de ferramenta selecione (Aa) Rótulo.

Clique com o mouse no espaço vazio entre o **Cabeçalho do formulário** e **Detalhe**.



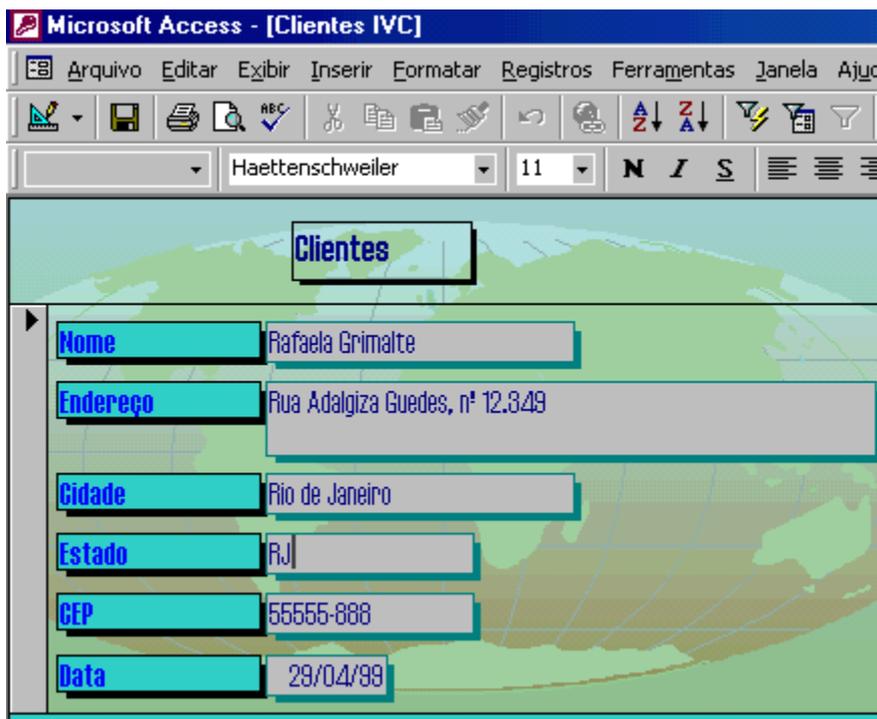
Digite o rótulo para seu formulário e pressione a tecla Enter quando terminar.

Você pode usar as mesmas formatações (fonte, cor, tamanho, etc.) que usou para o texto com o rótulo.

Para salvar as alterações em seu formulário clique em Arquivo/Salvar.

### 35. Exibir Formulário

Para exibir o resultado do formulário, clique no botão (  ) Exibir.



### 36. Relatórios

Você pode imprimir as informações de seu banco de dados utilizando tabelas ou as consultas e até mesmo os formulários. Mas a maneira mais fácil e profissional de impressão dos dados é utilizando o Relatório.

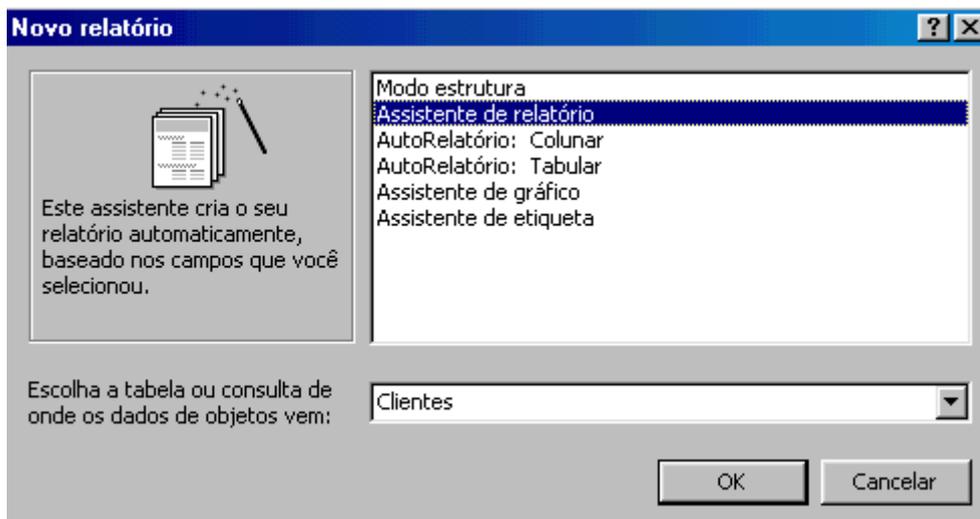
O módulo Relatório permite uma grande quantidade de controle e o utilizando assistentes para a criação de um relatório eficaz da maneira desejada.

### 37. Criando Um Relatório

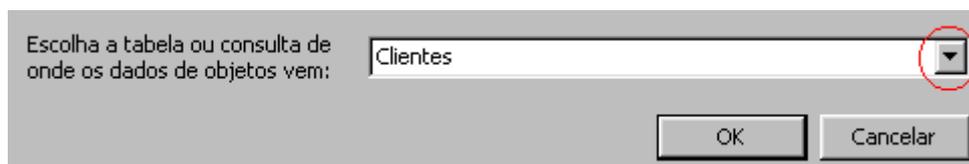
Para criar um relatório, abra o banco de dados e clique sobre a opção Relatórios.



Em seguida, clique no botão (  Novo).  
Uma janela denominada "Novo relatório" irá aparecer.

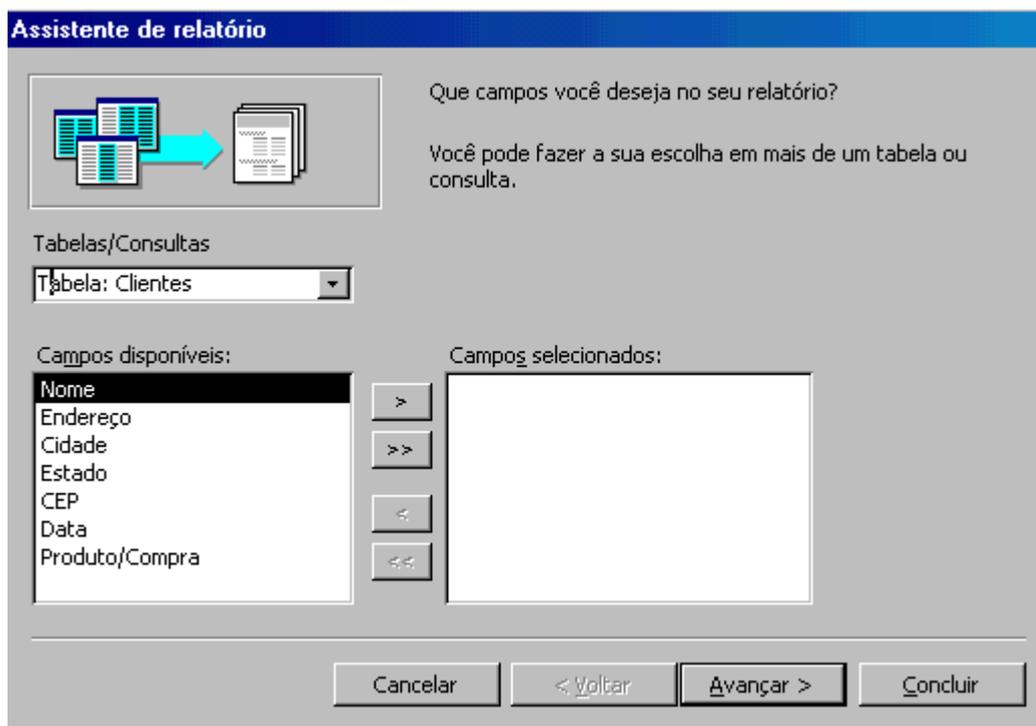


Clique na seta para baixo e escolha a Tabela ou Consulta para criar o relatório.

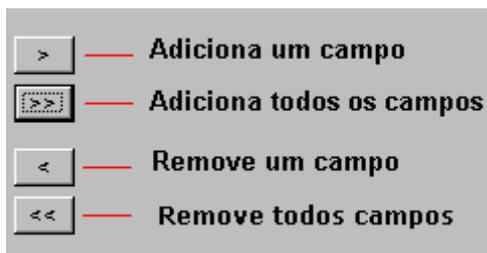


Selecione a opção Assistente de relatório e clique no botão OK.

Na próxima tela, selecione os campos que deseja colocar no relatório e movimente os para a caixa do lado direito da janela (Campos selecionados).



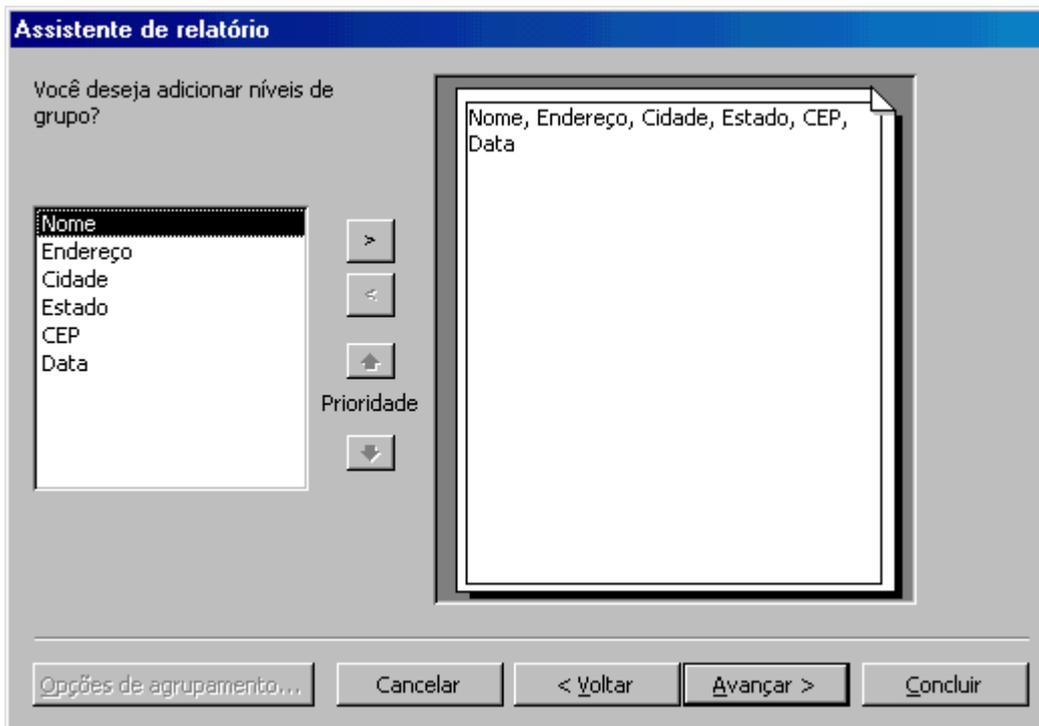
Repare que existem botões para movimentação dos campos disponíveis e selecionados.



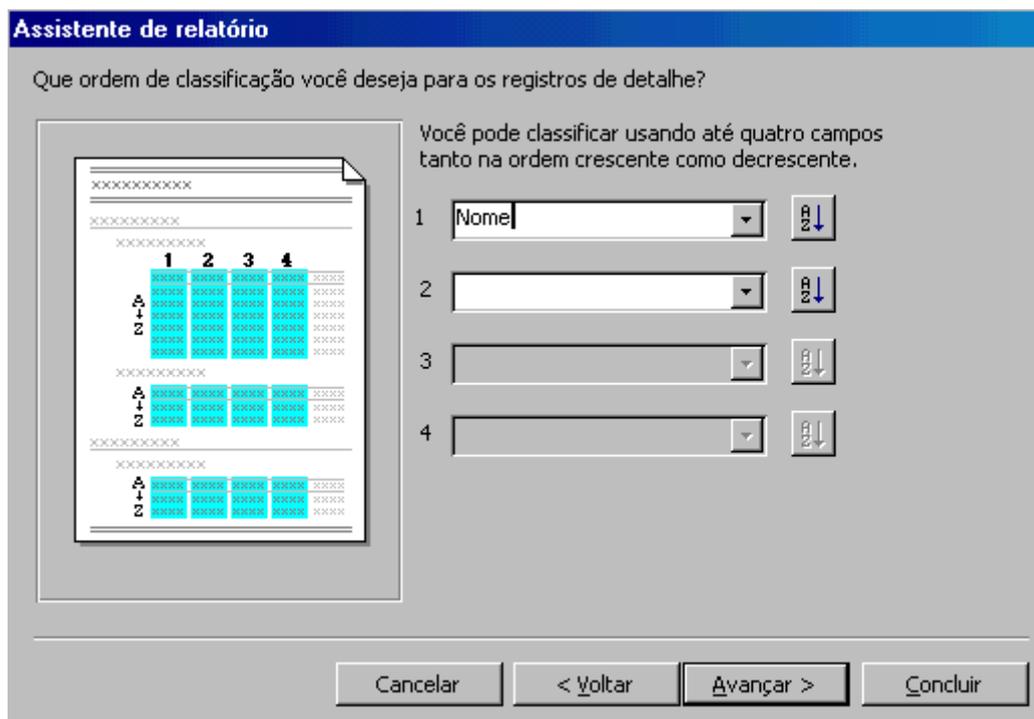
Ao terminar a seleção clique no botão Avançar.

Agora você deve definir a ordem de classificação do relatório escolhendo até quatro chaves de classificação. Clique na seta da caixa de listagem e escolha o campo.

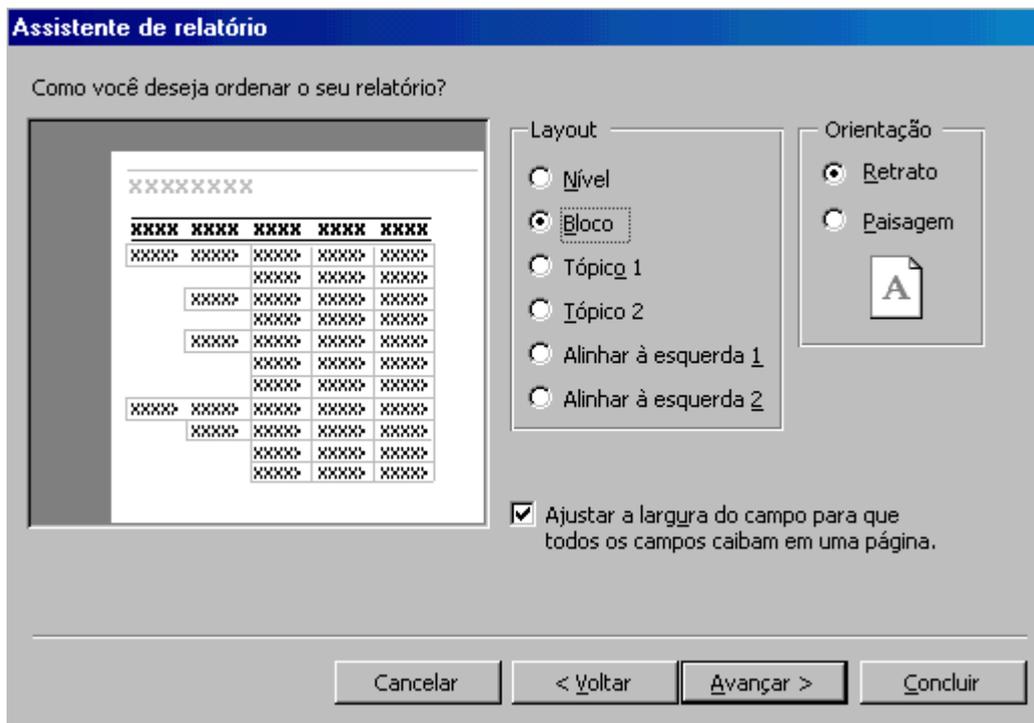
Clique no botão Avançar.



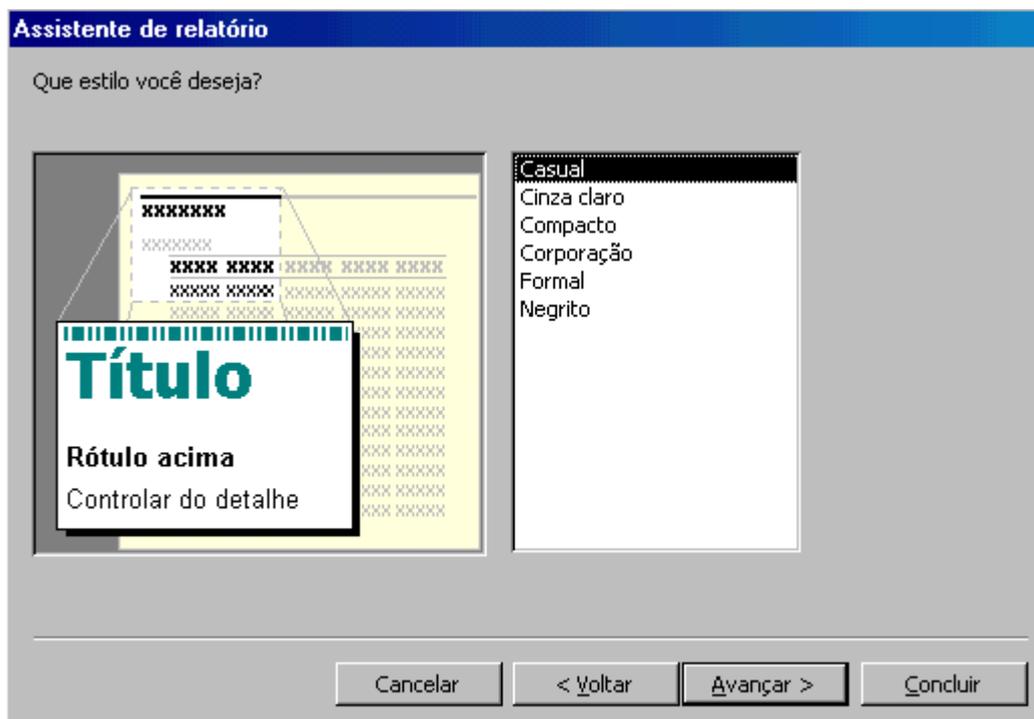
Para definir a ordem de classificação, clique no botão ao lado direito da caixa de listagem. Clique no botão Avançar.



Na tela seguinte, defina o layout para o seu relatório. Escolha o formato que mais lhe agrada ou condiz com as suas exigências. Clique no botão Avançar.



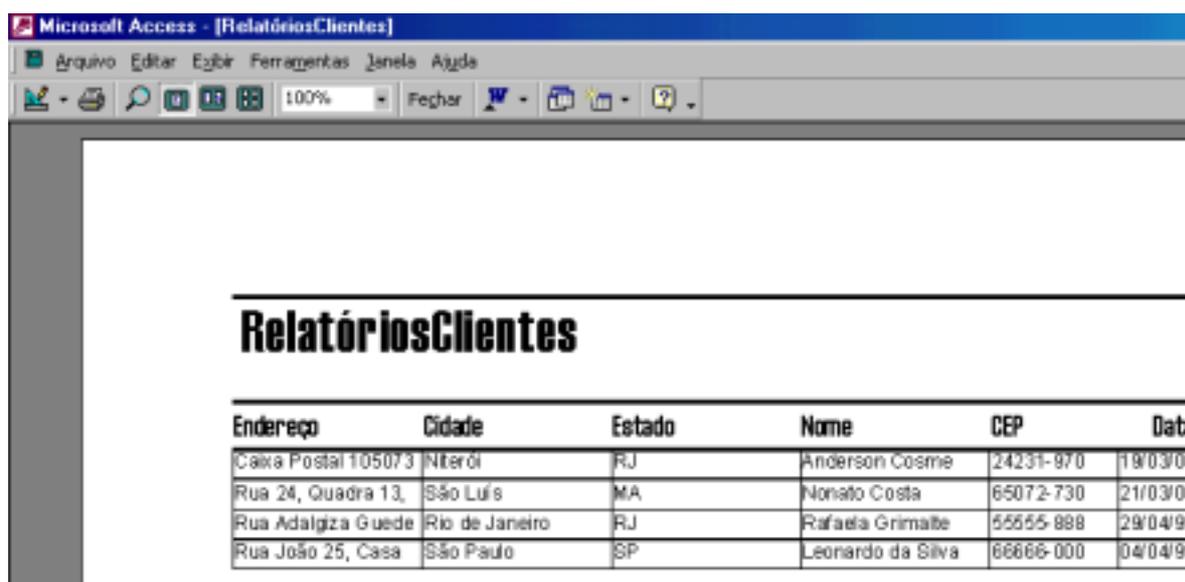
Agora você deve definir um estilo para seu relatório. Após decidir a opção que irá utilizar clique no botão Avançar.



Na próxima tela informe o título do seu relatório. Escolha a opção Visualizar o relatório, para poder visualizar sua aparência. Clique no botão Concluir.



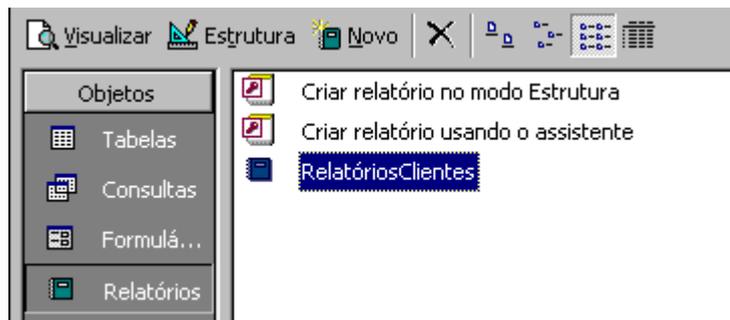
Tela do relatório já pronto.



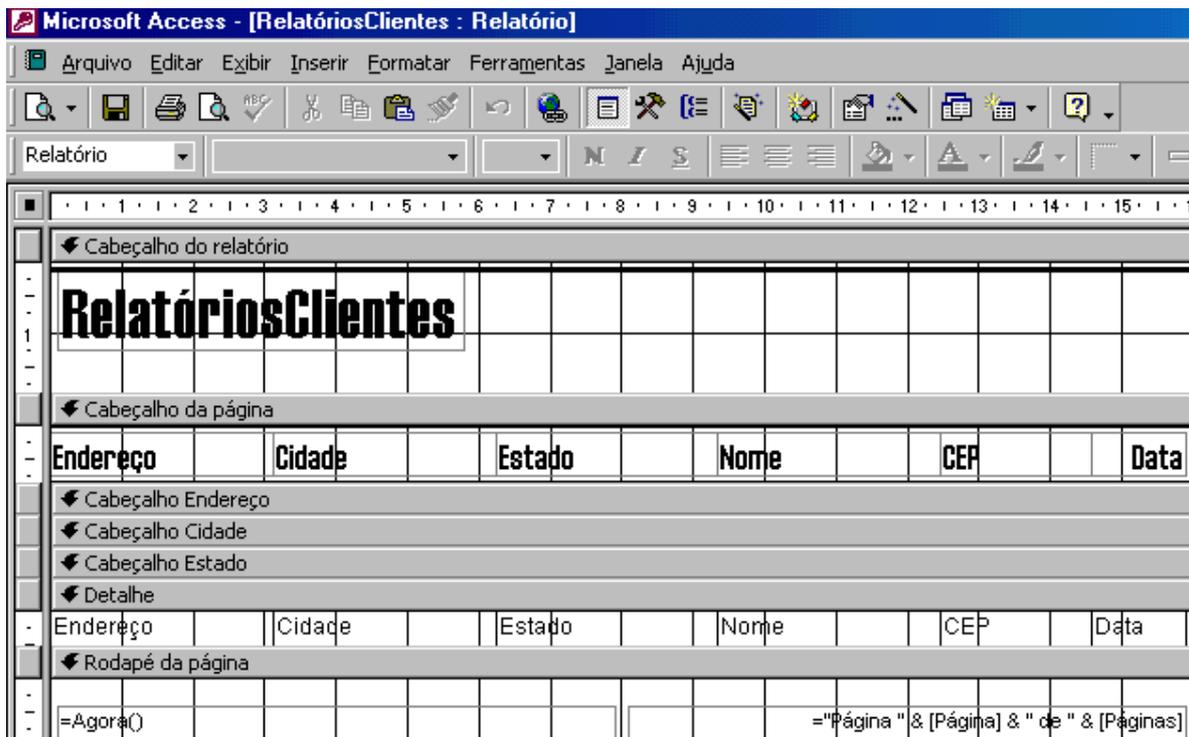
Clique no botão (🔍) Zoom para visualizar o relatório por inteiro na tela.

## 38. Estrutura do Relatório

Para entrar no modo estrutura do relatório, selecione o relatório e clique no botão (🔧) Estrutura.



No modo Estrutura, você poderá fazer várias alterações, dentre as quais:  
 Alterar a largura das caixas dos campos do relatório.  
 Alterar o texto do relatório. Modificar o posicionamento do texto.



Para alterar os objetos que compõem o relatório você precisa selecioná-los.  
 Após selecionar os objetos aplique as mudanças, que são:

Agora selecione uma (  ) Fonte para o texto.

O (  ) Tamanho da fonte.

A (  ) Formatação (negrito, itálico ou sublinhado).

O (  ) Alinhamento do texto (esquerdo, centro, direito).

A (  ) Cor de fundo/Preenchimento.

A (  ) Cor do primeiro plano/fonte.

A (  ) Cor da linha/borda.

A (  ) Largura da linha/borda.

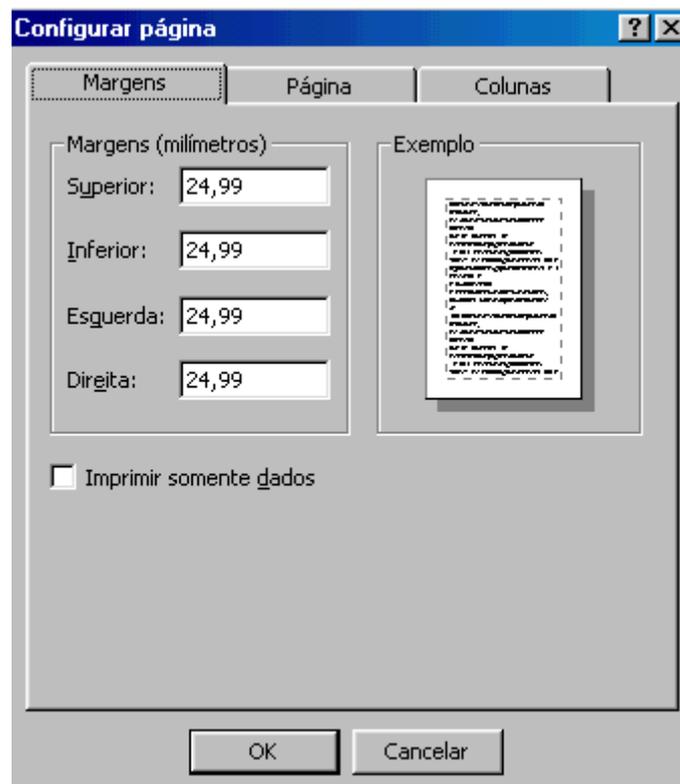
O (  ) Plano.

Após fazer as alterações clique no botão (  ) Salvar.

### 39. Imprimindo o Relatório

Para imprimir o relatório, você deve selecioná-lo e clicar no menu Arquivo/Imprimir ou clicar no botão (  ).

Antes de imprimir o relatório, você deve configurar a página a ser impressa. Selecione o relatório para impressão. Clique no menu Arquivo/Configurar página.



Você deve fazer as alterações de acordo com as dimensões da folha a ser utilizada para impressão. Exemplo: A4

Para aceitar as alterações clique no botão OK.

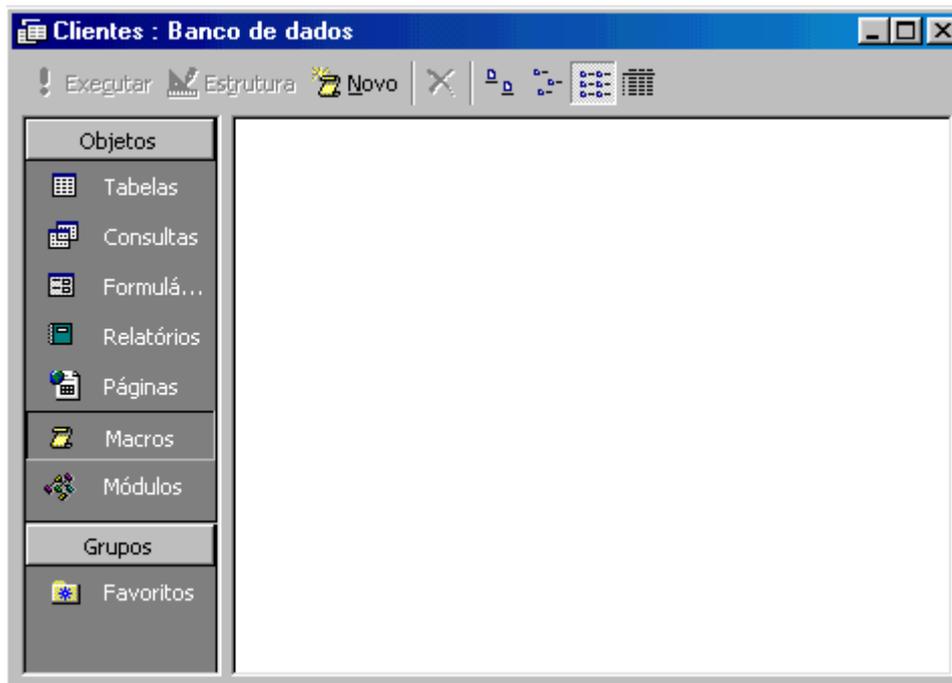
### 40. Macros

Uma macro é um conjunto de uma ou mais ações onde cada uma efetua uma determinada operação, como abrir um formulário, localizar um registro, imprimir um relatório, etc. As macros podem ajudá-lo a automatizar tarefas comuns. Por exemplo, você pode executar uma macro que imprima um relatório quando um usuário clicar em um botão de comando.

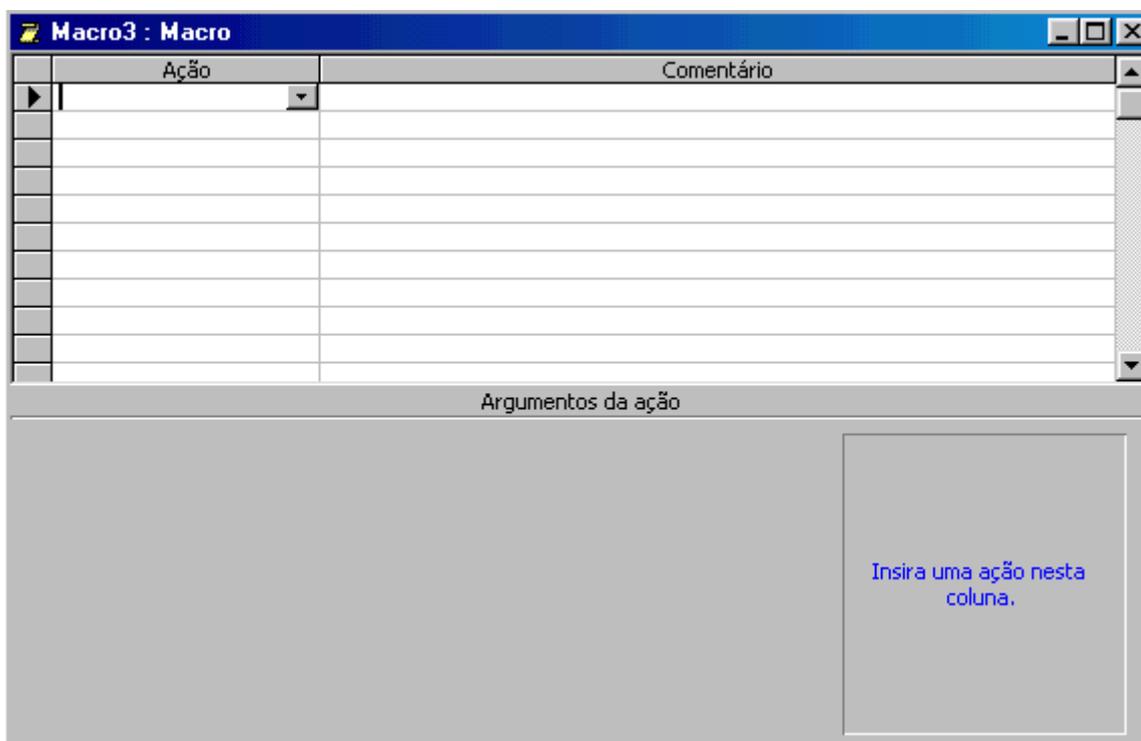
Uma macro pode ser uma macro composta de uma seqüência de ações ou pode ser um grupo de macros. Você também pode utilizar uma expressão condicional para determinar se, em alguns casos, uma ação será realizada quando uma macro for executada.

## 41. Criar Uma Macro

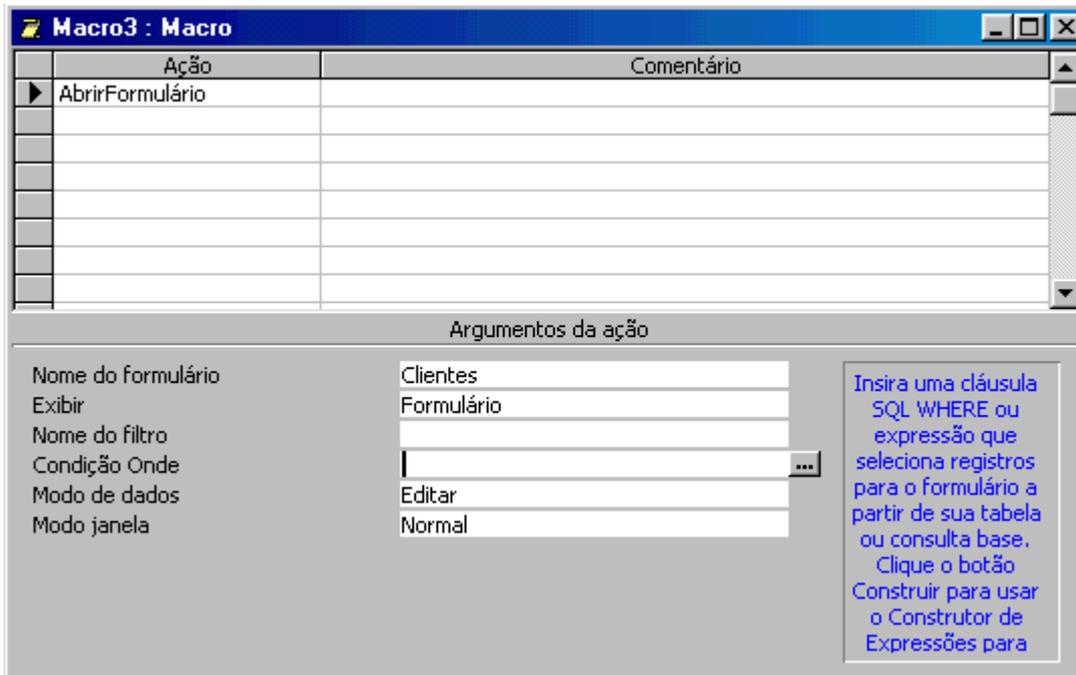
Para criar uma macro abra seu banco de dados, selecione a guia Macros, clique no botão (  Novo) Novo.



Uma nova tela irá aparecer. Na coluna Ação escolha a ação desejada para se executada pela macro. Selecione a ação clicando na seta para baixo.



Após selecionar a Ação que a macro executará, defina os argumentos da ação na parte inferior da tela.



Os argumentos da ação dependem da sua escolha na coluna Ação.

Salve sua macro clicando no botão (  ) Salvar.

Execute a macro criada clicando no botão (  ) Executar.

## 42. Código Visual Basic No Microsoft Access

A melhor maneira de trabalhar com banco de dados é no modo Formulários. Utilizando formulários você terá uma interface gráfica que lhe permitirá visualizar dados de uma maneira mais rápido e eficiente.

Utilizando formulário você pode construir códigos Visual Basic para executarem macros ou fazer chamada de outros formulários para Editar, visualizar ou adicionar dados.

## 43. Criar Código Visual Basic

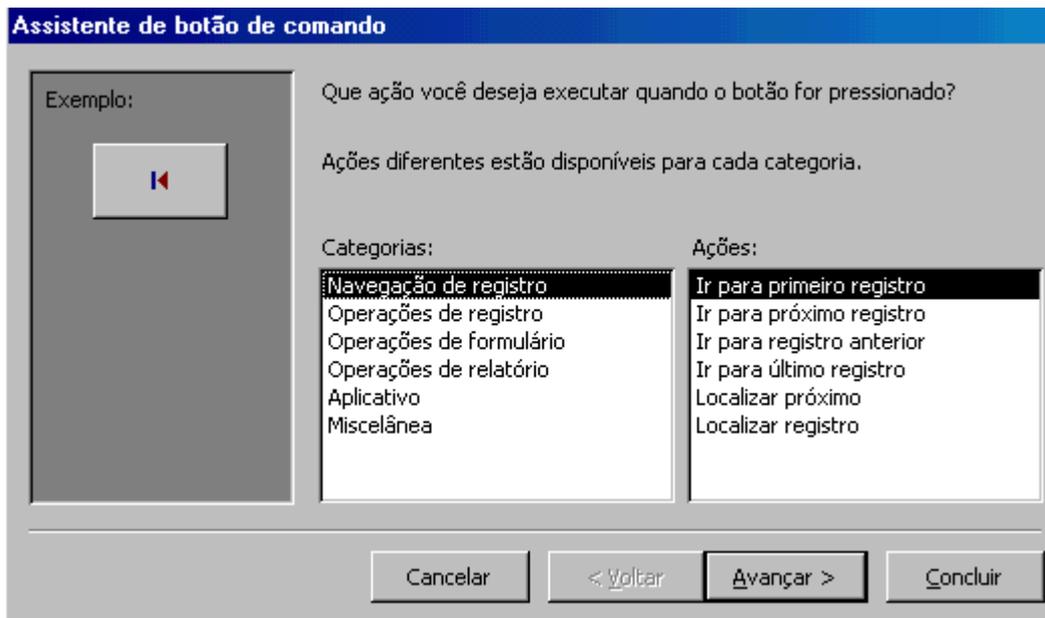
Para criar código Visual Basic selecione a guia Formulários, selecione o formulário desejado e clique no botão (  Estrutura ) Estrutura.

Suponhamos que você tenha um formulário folha de pagamento e quer obter todos os dados a respeito de um funcionário. Dados esse que estão armazenados no formulário funcionário. Portanto você deve criar uma Macro que abra o formulário funcionário, no campo desejado e na forma desejada (Editar, adicionar ou Visualizar).



O mecanismo ideal para executar uma macro é o Botão de comando, que você adiciona em seu formulário clicando no botão de (  ) Caixa de ferramentas.

Ao adicionar um botão de comando uma janela Assistente de botão de comando aparecerá. Para poder personalizar este botão e incluir código Visual Basic clique no botão Cancelar.

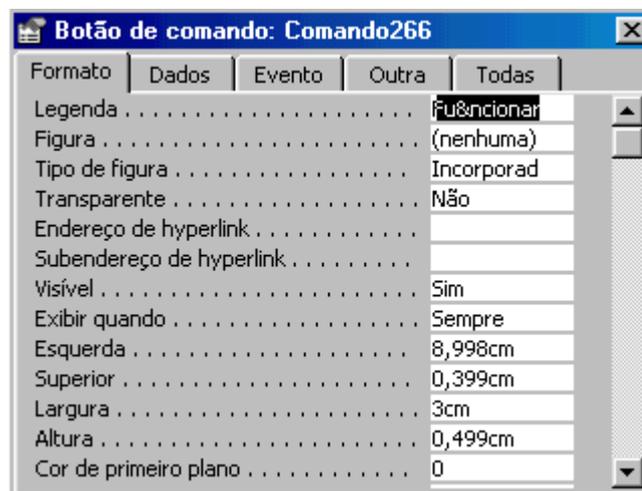


O Botão de comando será incluído no formulário bem na área onde você clicou no formulário.



Você personalizar este botão alterando seu rótulo e acrescentando a ele uma tecla especial de atalho.

Para personalizar este botão selecione-o e clique no botão (🔧) Propriedades.



Na guia Formato selecione legenda e digite:

***Fu&ncionário***

Isso faz com que se possa acessar este botão pressionado as teclas [Alt] + [n].

É também na guia Formato que alteramos a aparência do(s) objeto(s).

Você pode mudar a altura, largura, figura, visível (sim ou não), cor e tamanho da fonte etc.

Após modificar as propriedades do botão clique no botão (🔧) Código.

Uma tela do editor de código se abrirá.

Escolha o botão que criou clicando na seta para baixo.



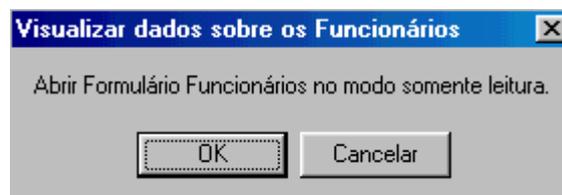
O evento que irá aparecer é o OnClick (Ato de clicar no botão).



Digite o seguinte código para abrir o formulário através de uma macro de nome "Funcionários".

```
Dim Abrir As String  
Abrir = "Abrir Formulário Funcionários no modo somente leitura"  
If MsgBox(Abrir, vbOKCancel, "Visualizar dados sobre os Funcionários") = vbOK Then  
    DoCmd.RunMacro "Funcionários"  
Else  
    Exit Sub  
End If
```

Quando este formulário for acessado e o botão que foi adicionado for clicado, um prompt aparecerá, com dois botões como opções de abertura do formulário. O botão Ok (VbOk) abre o formulário e o botão Cancelar (VbCancel) cancela a ação.



O formato de abertura do formulário foi determinado pela Macro "Funcionários", que permite que este formulário seja aberto somente no modo leitura.

Para salvar as alterações em seu formulário clique no botão (📁) Salvar.

# Normas e Padrões de Produtos Access

## 1. Normas e Padrões



### 1.1. O por quê

Existe dificuldade em se passar especificações do usuário para que o programador possa criar o código (programa).



ideal é a passagem de informações ser por este caminho direto, mas normalmente é impossível

- . O usuário não conhece e nem deve conhecer de máquinas e programação.
- . O Analista deve conhecer razoavelmente bem a programação e as áreas de negócio do usuário. Todavia não necessita ser um especialista.
- . Já o programador deve conhecer bem sua área: fazer programas.

Três pessoas falando termos, regras e processos, de 3 formas diferentes.

Daí surge a necessidade de padronização. A padronização torna possível que todos os envolvidos falem a mesma língua, ou pelo menos entendam o que os outros dizem.

Além disto, o que não está escrito muita gente confunde com “não existe padrão”, gerando muita confusão e desorganização, principalmente quando se fala em Informática, paraíso dos auto-normativos. Portanto Normas SIM!

### 1.2. Conceito

Normas ou padrões ou premissas ou metas ou regras ou semelhantes, são descrições de normas e procedimentos, premissas gerenciais, objetivos e metas a alcançar, regras de operação e todas as demais especificações referentes a padrões a serem seguidos na organização.

Constituem, em geral, de textos ou dados tabelados, formando um conjunto de determinações que servem como guia para as operações da organização, em complemento aos programas e planos.



### 1.3. Objetivos

- a) Organizar os documentos e produtos da Prefeitura.
- b) Zelar pela uniformidade dos serviços.
- c) Proporcionar um entendimento rápido e duradouro da metodologia e aspectos operacionais adotados.
- d) Proporcionar que o usuário ache facilmente arquivos gerados no disco rígido.
- e) Assegurar que todos os sistemas tenham a mesma aparência, permitindo assim, um treinamento e uma manutenção rápida e segura.
- f) Estabelecer uma harmonia total dentro da equipe de trabalho de desenvolvimento de sistemas e usuários.

### 1.4. Premissas Finais

O início de uma construção, de uma empresa, de um experimento e principalmente de um projeto é de fundamental importância. É neste momento que se formam as bases para um empreendimento sólido. E quanto maior a envergadura do empreendimento, maior será o tempo nesta base.

A base de um projeto de informatização, além da Análise e Conhecimento do Negócio e das Metodologias, é a criação das Normas e Padrões a serem adotados no decorrer deste projeto.

Portanto, para uma eficaz construção é necessário um eficaz documento de Normas e Padrões, tão abrangente quanto possível neste início de atividade.

As instruções e informações aqui descritas são resultado da busca do melhor desempenho e da diminuição dos processos de manutenção. Esta busca é resultado de experiência no ramo de consultoria e desenvolvimento de sistemas, e fazem parte do padrão de qualidade que se deseja para serviços e produtos DEMIN-PMP.

## 2. Siglas, Abreviaturas e Regras Definidas

Em trabalhos de enfoque integrado objetivando uma solução corporativa, a equipe de criação da solução deve estar tão integrada e afinada quanto se espera que o sistema esteja. Para tanto a normatização de nomes é um ponto importante para que as diversas pessoas envolvidas possam criar as definições dentro do mesmo padrão e encontrar rapidamente o que procuram.

### 2.1. Códigos dos Sistemas

Não há nomenclatura específica para sistemas.

Eis alguns sistemas da DEMIN-PMP e seus códigos respectivos:

Sistemas	Abreviatura
0. Normas <sup>1</sup>	NOR
1. CAU	CAU
2. Genérico <sup>1</sup>	GEN
3. Cadastro Único <sup>1</sup>	UNI
4. Almoxarifado	ALM
5. Ambiental	AMB
6. Atendimento	ATE
7. Contabilidade	COM
8. Compras	COM
9. Educação	EDU
10. Financeiro	FIN
11. Folha Pgto - RH	FOL
12. Frota	FRO
13. Gerência	GER
14. Intranet	INT
15. Legislação	LEG
16. Orçamento	ORC
17. Patrimônio	PAT
18. Saúde	SAL
19. Serviço Social	SOC
20. Tributação	TRI
21. Urbanismo	URB

### 2.2. Nomes de Tabelas

1. O nome da tabela deve ser escrito no **singular**, seguindo a gramática da **língua portuguesa**.
2. A primeira letra do nome de uma tabela deve ser **maiúscula** e as demais minúsculas, **salvo se o nome da entidade for composto**. Neste caso o segundo ou terceiro nome devem iniciar por maiúscula novamente.

<sup>1</sup> Não é um sistema, mas é uma área que deve ter manutenção a parte. Normas - são as Normas e Padrões para a criação e manutenção de sistemas. Genérico - são as telas país, as telas padrões e as rotinas de segurança. Cadastro Único é o cadastro de pessoas e suas tabelas únicas para todos os sistemas.

3. No caso do uso de **siglas** na composição do nome da tabela, **todas as letras devem ser maiúsculas**.
4. Ao usar abreviaturas ou siglas certifique-se de sua clareza. O que é trivial para o analista, pode não ser para o usuário. Ex.: NF, EBD, CEPs, UF são algumas abreviaturas previstas.
5. O tamanho máximo do nome deve ser **64 caracteres**.

### EXEMPLO DE NOMES DE TABELAS

Descrição	Nome da Entidade
Controle de Cheques Pré Datados	ChequePréDatado
Empresa Atual utilizando o sistema	EmpresaAtual
Pessoa	Pessoa
Forma de Pagamento	FormaPagamento
Tipos de Pessoas (Fornecedor,Usuário...)	TipoPessoa

6. Se uma tabela tem **cabeçalho e itens**, o nome da tabela item pode ter o nome item no final. Ex.: NF e NFItem.
7. Exceção para o sub-tópico acima, são as **Associativas** que devem ter como segundo nome, o dado que deverá ser fixo, e o segundo sim é que deve ser variado. Ex.: HabilidadeTecnico => Técnico é que deve ser fixo. Isto é, cadastrarei as *n* habilitações do técnico no cadastro de técnicos. CargoPessoa => Cadastro os *n* cargos para determinada pessoa.

### EXEMPLO DE NOMES DE TABELAS ASSOCIATIVAS

Descrição	Nome da Entidade
Contrato com clientes	ContratoCliente
Itens da nota fiscal	NFItem
Tipo de Serviço	TipoServiço
Nota Fiscal	<b>NF</b>
Agenda do Atendimento	AgendaAtendimento
Aluno Por Classe Por Matéria	AlunoPorMateriaPorClasse
Aptidão Por Pessoa	AptidaoPorPessoa
Pessoa Por Secretaria	PessoaPorSecretaria
Evento Por Grupo	EventoPorGrupo

## 2.3. Nomes de Colunas

1. Valem as mesmas regras para criação dos nomes de entidades (tabelas).
2. Os mnemônicos, quando necessários, devem ser escritos, sempre que possível obedecendo as abreviaturas dispostas no Anexo 1.
3. Os mnemônicos padronizados estão descritos na tabela a seguir.
4. Havendo necessidade de criar novos mnemônicos informar ao DA imediatamente.

### TABELA DE MNEMÔNICOS

Veja ANEXO 1.

Quando da criação das tabelas, acrescentar a **descrição de cada campo**, pois estas descrições serão utilizadas quando da criação das telas (formulários), devendo aparecer no Micro Help (pequeno retângulo informativo na parte inferior da tela).

Nome do Campo	Tipo de Dados	Descrição
CodPessoa	Número	Código Numérico da Pessoa
CodMembro	Texto	Código de Identificação de Membro (o mesmo do Rol de Membros)
CodIgreja	Texto	Código da Igreja a que Pertence, Visita ou Cooperar
CodTpPessoa	Número	Código do 1o. Tipo de Pessoa. Determina o real Tipo da Pessoa, o qual ele é Conhecido
CodTpPessoa1	Número	Código do 2o. Tipo de Pessoa. Determina uma Segunda Atividade da Pessoa
NomePessoa	Texto	Nome Completo da Pessoa
CodPessoaResponsável	Número	Código da Pessoa Responsável (normalmente o código do Pai ou Chefe da Família)
CodHierarquia	Número	Código do Nível Hierárquico desta pessoa com relação à pessoa responsável (1=responsável)
CodCidPessoa	Número	Código Numérico da Cidade onde Mora
CodLogrPessoa	Texto	Código do Logradouro do Endereço
EndereçoPessoa	Texto	Endereço da Pessoa
BairroPessoa	Texto	Bairro da Pessoa
CEPPessoa	Texto	CEP da Pessoa
IndTpPessoa	Texto	Indicador de Tipo de Pessoa: F - Físico, J - Jurídico
CGCCPFpessoa	Texto	CPF ou CGC da Pessoa (Depende ser Física ou Jurídica)
DtNascimentoPessoa	Data/Hora	Data de Nascimento da Pessoa
Fone1Pessoa	Texto	Primeiro Fone da Pessoa ou do Primeiro Contato
Ramal1Pessoa	Texto	Primeiro Ramal do Primeiro Fone
Fone2Pessoa	Texto	Segundo Fone da Pessoa ou do Segundo Contato

Propriedades do Campo	
Tamanho do Campo	Inteiro Longo
Formato	Número Geral
Casas Decimais	0
Máscara de Entrada	
Legenda	Código
Valor Padrão	
Regra de Validação	
Texto de Validação	
Requerido	Sim
Indexado	Sim (Duplicação Não Autorizada)

Um nome de campo pode ter até 64 caracteres incluindo espaços.  
 Pressione F1 para Ajuda sobre nomes de campo.

## 2.4. Nomes de Objetos

### 2.4.1. Nomes de Identificadores (variáveis, tabelas, colunas e outros)

- . Podem ter até 64 caracteres (utilize de preferência até 30).
- . Salvo raras exceções (abreviaturas), crie nomes no singular.
- . Podem incluir qualquer combinação de letras, números e os seguintes caracteres: -\_\$. Todavia espera-se que não se utilize, devendo os nomes apresentarem o seguinte:
  - Letras Maiúsculas no início de cada palavra ou significado
  - Não serem abreviados
  - Sem espaço ou underline ( ) separando-os
  - Nomes curtos mas que realmente identifiquem suas funções
  - Utilizar acentos sempre que possível
  - Se houver necessidade de repetições, utilizar a numeração sequencial (1, 2, 3 ...)
  - Nomes indiferentes para variáveis Globais, Locais, de String, Numéricas... Uma vez que este controle no Access ou Visual Basic é subjetivo (diferentemente de Power Builder, C, C++).

Exemplo:

```

CodPessoa
IncluiCod
SalárioAtual
Acumulador
VarAux (Variável Auxiliar - exceção de abreviatura)
TotalSalário, TotalSalário1, TotalSalário2...
EndereçoPessoa
    
```

### 2.4.2. Nomes de Recursos

Tabela

Já descrito no item 2.2. e 2.4.1.

### Consulta

Se a consulta for de uma tabela existente, deve ser de nome igual, acrescentando c\_ (devido o Access identificar como mesmo objeto os nomes de consultas e tabelas).

Exemplo: c\_Pessoa, c\_Membro, c\_Forma\_Pagamento

Se a consulta for repetida, deve-se acrescentar após o nome, um nome que identifica o porque da consulta.

Exemplo: c\_Pessoa, c\_Pessoa\_Item

Se a consulta não tiver tabela imediatamente relacionada (normalmente está relacionada a mais de uma tabela), crie nomes com letras Maiúsculas no início, sem separadores, com c\_ no início (veja nomes de identificadores).

Exemplo: c\_Valores\_De\_Gasto, c\_Etiquetas\_de\_Membros\_Inativos

### Formulário

Utiliza a mesma nomenclatura de Identificadores já descrita, somente que, quando houver nome repetido, utilizar um acréscimo que identifique melhor o formulário.

Exemplo: TelaInicial, Pessoa, Membro, Cerimônia, Prontuário, Profissão.

Além disto, quando o formulário for um sub-formulário realmente identificado como tal (não será utilizado como formulário principal), acrescente um X no início do nome. Isto fará com que ele vá para o final da listagem e rapidamente você o identificará como um sub-formulário através do nome.

Exemplo: XPessoaItem, XHoraTrabalhadaItem, XParcela, XNFItem.

### Relatório

Obedece a mesma norma de Formulário. Somente que pode ser utilizado separadores (espaços) entre nomes.

### Macro

Obedece a mesma nomenclatura de identificadores, somente que as palavras devem vir separadas com underline (\_).

### Módulo

Obedece a mesma nomenclatura de macros.

## 2.4.3. Nomes de Controles

Para que você ou outra pessoa que irá manter os programas, e para os documentadores de objetos, os nomes de controles devem obedecer uma identificação única, de tal sorte que você, ao ver o nome do controle, saiba distinguir o que ele identifica.

O Access cria nomes *default* de controles, como campo1, campo58... Altere o nome para o mesmo nome do campo da tabela. Se for repetido acrescente 1, 2, 3... ao final do nome. Se não estiver ligado a campo, deve obedecer a mesma nomenclatura para identificadores descrita acima.

Exemplo:                   CodPessoa  
                                  DescrPessoa

A exceção deve ser feita para botões. Uma vez que eles são extremamente comuns, seria interessante que fossem identificados na lista de controles quando é necessário escrever códigos para eles.

Para botões acrescente Bt\_ antes do nome, que obedece a nomenclatura de Identificadores.

Exemplo:                   Bt\_Descrição  
                              Bt\_RelatórioDeMembros

E a exceção se aplica para caixas de combinação: Neste caso, acrescente \_CxC depois do nome, que também obedece a nomenclatura de Identificadores.

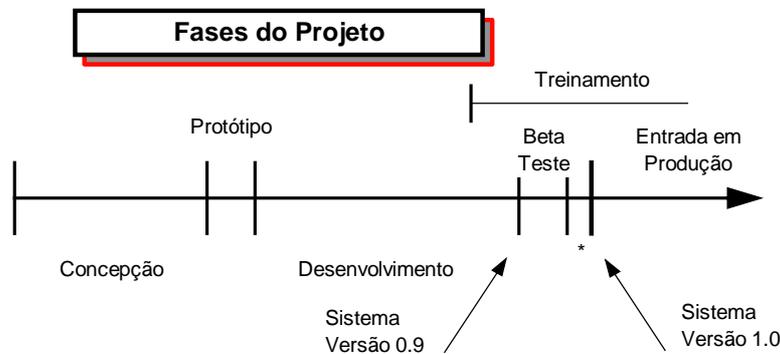
Exemplo:                   CodPessoa\_CxC  
                              Descrição\_CxC  
                              CodGrupo\_CxC



### 3. Normas para Produtos

#### 3.1. Análise

A metodologia de Análise e de sua documentação é a já em uso pela DEMIN-PMP, e, portanto, não deve sofrer modificações. O resumo da metodologia é mostrado abaixo.



Concepção	Fase em que se delinea as bases do projeto. Faz-se o levantamento das informações na Contratante e o desenho da aplicação, criando-se o Modelo de Dados.
Protótipo	Passa-se para a tela todos os dados da fase anterior, desenhando-se a navegação da aplicação. Cria-se um esqueleto do sistema, e aprova-se com a Contratante se o sistema desenhado até então, atende plenamente suas necessidades.
Desenvolvimento	É nesta fase que efetivamente cria-se o sistema, isto é, cria-se os programas para o sistema. Se houve uma fase de protótipo, esta fase tende a ser encurtada.
Beta-Teste	Após quase a totalidade do código (programa) estar criado, passa-se para a Contratante utilizar o sistema, entrando dados reais do seu dia a dia. A Contratante deverá aprovar ou não, um a um dos programas, gerando uma lista de alterações a serem feitas, se necessário. Note que esta fase significa testes em conjunto, uma vez que alguns programas serão liberados com antecedência, e testados com antecedência também.
* Alterações Finais	É feita nesta etapa, as últimas alterações para esta versão do sistema, gerando-se, ao final, a versão 1.0 do sistema, pronta para entrar em produção.
Treinamento	Treinamento do Administrador e do(s) Operador(es) do sistema assim como noções de Windows, backup, fax, internet, suporte remoto e inúmeras outras dicas.

#### 3.2. Programação

O padrão descrito abaixo deve ser adotado no desenvolvimento de aplicações. Este padrão não visa bloquear a criatividade dos técnicos no processo de desenvolvimento, mas sim assegurar que todos os sistemas tenham a mesma aparência, permitindo assim, uma manutenção rápida e segura.

É extremamente importante que as diretrizes sejam seguidas para facilitar o próprio trabalho de quem está em desenvolvimento e futuramente para quem fizer a manutenção dos sistemas.

Se ao longo do projeto for detectado que determinado padrão possa ser melhorado em função de um desenvolvimento mais preciso, pede-se que as sugestões sejam apresentadas para discussão.

### 3.2.1. Procedimentos

Um programa/objeto estará concluído quando estes requisitos estiverem cumpridos:

- estiver testado operacionalmente;
- documentado dentro dos padrões estabelecidos;
- conferido e aceito pelo analista ou usuário responsável.

### 3.2.2. Programas/Módulos

#### Modelo de construção de Módulo:

```
' *****
' Function...: Atualiza_Vlr_Total ()
' Propósito...: Atualiza no formulário de TRANSAÇÃOPRINCIPAL, o Valor Total
'               da compra do cliente
' Parâmetros...: Nenhum
' Retorno.....: Nenhum
' Observações.: Utilizado no evento AFTERUPDATE do formulário
'               XTRANSAÇÃOITEM e no evento BEFOREINSERT do formulário
'               XPARCELAS.
' Alterações..: Data      Quem      O que
'               07/08/1994  José      Baixa na agenda
'               08/08/1994  Edson     Commit das tabelas ativas
' Nr          Histórico:
' 1           A baixa do horário na agenda do técnico era efetuada somente após a
'             emissão da agenda, isto gerava duplicidades no momento de programa-
'             ções simultâneas para o mesmo técnico. Por este motivo a baixa será
'             feita imediatamente após a programação.
'
' 2           Na última alteração o Commit foi acidentalmente removido do código
'
' *****
Function Atualiza_Vlr_Total ()
On Error Goto Err_Atualiza_Vlr_Total
' NumLinhas neste script não está sendo usado
Dim NomeQuery As String, QueryMesmo As String, MeuBD As Database
Dim Acumulado As Double
' Cria Query para as n Parcelas dita em Forma de Pagamento
Set MeuBD = CurrentDB()
NomeQuery = "c_valor_Total"
QueryMesmo = "SELECT VlrUnitárioMater,QtdeMater FROM TransaçãoItem WHERE"&
"TransaçãoItem.CodTrans = " & Forms!TransaçãoPrincipal![Transação.CodTrans]&
";"
Acumulado = 0

If Cria_SnapShot(NomeQuery, QueryMesmo) >= 1 Then
Do Until rSnapShot.EOF
Acumulado = Acumulado + rSnapShot!VlrUnitárioMater * rSnapShot!QtdeMater
' Localiza próxima ocorrência.
rSnapShot.MoveNext
Loop
End If

' Calcula o Valor Total em Dinheiro
Forms!TransaçãoPrincipal![VlrTransTotal] = Acumulado
```

```
Exit_Atualiza_Vlr_Total:
    Exit Function

Err_Atualiza_Vlr_Total
    MsgBox Error$
    Resume Goto Exit_Atualiza_Vlr_Total

End Function
```

A confecção de cabeçalho para *scripts* de módulos e sua atualização criteriosa, permite uma avaliação mais precisa para que se possa determinar a necessidade de novas versões, facilitando esta criação e posteriores manutenções.

Não utilizar este cabeçalho nos *scripts* de eventos, e outros que possam haver. Utilizar somente em módulos. Entretanto nos *scripts* de eventos deverá haver comentários quanto aos nomes dos objetos envolvidos e funções chamadas, salvo se houver trivialidade exagerada (não esqueça que o que é trivial para você, pode não ser para outra pessoa).

Se os programas passados à área de desenvolvimento são excessivamente complexos, torna-se necessário, além da documentação até então conhecida, efetuar a elaboração de uma **especificação de programa** descrevendo aquele processo. A especificação de programa deverá ser a ferramenta de descrição de processos mais perto da linguagem utilizada.

Alguns detalhes da especificação de programa (que o programador julgar necessário) deverão vir no programa (*function*) como comentário de cabeçalho, e deverá ser escrito pelo programador antes de começar o desenvolvimento em si. Sua finalidade é o encontro pelo programador, no início do desenvolvimento do programa, de algo não necessariamente detalhado, ou ainda, falhas de descrição técnica, ou até não entendimento da concordância gramatical e sintaxe do "português" utilizado para descrição do processo.

Isto fará com que a perda de tempo proporcionado por erro encontrado somente quando já se escreveu boa parte do programa não seja sentida.

O conteúdo mínimo da especificação de programa, quando necessário esta especificação, será:

1. Dados que serão informados, inclusive quais bancos e como se fará acesso a estes bancos;
2. Os testes que houver em dados informados (consistência);
3. Passos de programas;
4. Consulta a tabelas (quais arquivos);
5. Testes de condições especiais;
6. Definição de estrutura de arquivo que este programa venha a gerar,
7. Fórmulas com: Nome; Representação matemática; Constantes e variáveis; Condição para uso; Exemplos com valores simples...

### 3.2.3. Observações

1. O tamanho da linha nos códigos não deve exceder as 80 colunas, salvo SQLs excessivamente longos. Mesmo assim, nos SQLs utilize a quebra em variáveis concatenadas sempre que possível.

2. A endentação de condições e outros recursos de programação deve ser feita com no mínimo 3 espaços. Todavia utilize muito o alinhamento com a segunda palavra da frase da linha de cima, salvo se for muito distante (superior a 5 espaços).

```
If Cria_SnapShot(NomeQuery, QueryMesmo) >= 1 Then
  Do Until rSnapShot.EOF
    Acumulado = Acumulado + rSnapShot!VlrUnitarioMater
    ' Localiza próxima ocorrência.
    rSnapShot.MoveNext
  Loop
End If
```

3. Utilizar inicial maiúsculas nos comandos principais, como:

```
If ... EndIf
While ... EndWhile
Entre outros.
```

4. Comentários internos ao código fonte deve respeitar a premissa de cabeçalho já citado anteriormente e ainda durante o código deve conter quando necessário para deixar sem dúvidas:

- variáveis
- chamada a arquivo de banco de dados/tabelas
- chamadas a outros objetos
- chamadas a bibliotecas
- fórmulas

5. Se houve dificuldade na confecção de determinado trecho de código, provavelmente ele irá ser de difícil interpretação para quem for fazer manutenção. Portanto este trecho deve ser muito bem documentado.
6. Estes comentários devem aparecer sempre que determinados trechos de código apresentarem complexidade, e se julgados necessários nos demais casos.

## 3.3. Telas

### 3.3.1. Mensagens

Tanto o Visual Basic como o ACCESS permite que você utilize cabeçalho nas mensagens, além da mensagem no meio da janela de mensagem. Esta última é representada pela primeira mensagem do MsgBox. E o cabeçalho da mensagem é representado pelo terceiro parâmetro. É sobre eles que padronizaremos agora:

#### “Falta de Informações”

```
MsgBox "Preencha o campo de Transação.", 48, "Falta de Informações"
```

Como por exemplo em uma consistência de Tela que o usuário não entrou com valor nenhum.

#### “Erro de Informações”

```
MsgBox "Código de Cliente = 0 ou Tipo da Transação = Compra. Só Poderá Ser Emitido Carnê de Venda!!", 48, "Erro de Informações"
```

Como por exemplo em uma consistência de Tela que o usuário entrou com valor errado.

#### “Informações Erradas”

```
MsgBox "Número de Parcelas no Sub-Formulário PARCELAS Diferente do Número Previsto pela Forma de Pagamento.", 48, "Informações Erradas"
```

Quando está ocorrendo um processamento e ocorreu erro de consistência que não deveria ocorrer.

### “Estoque Insuficiente”

```
MsgBox("Deseja fazer a transação assim mesmo?", 1, "Estoque Insuficiente")
```

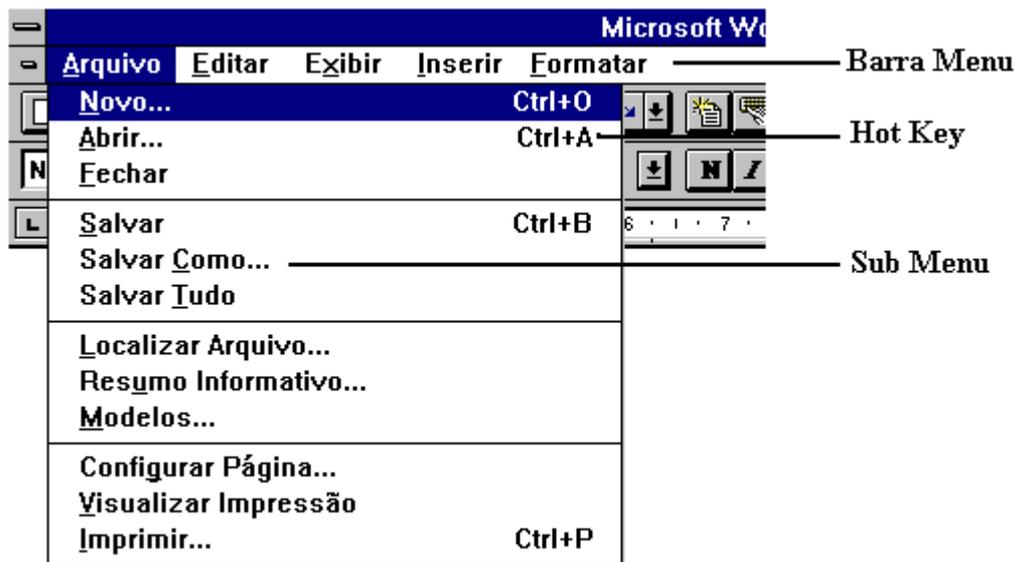
Quando houver necessidade de **Mensagens Específicas**. A criação destas mensagens deve ser evitada mas será inevitável que ocorra.

### 3.3.2. Menus

As telas no padrão GUI devem ser projetadas para serem de operação simples, fácil e intuitiva. O *mouse* pode ser exigido em determinadas funções.

A escolha da opção deve ocorrer num *click* ao ícone indicador da ação ou ALT + caractere sublinhado ou via *Hot Key*.

Os menus *Pull Down* e *Drop Down*, devem seguir o padrão do windows.

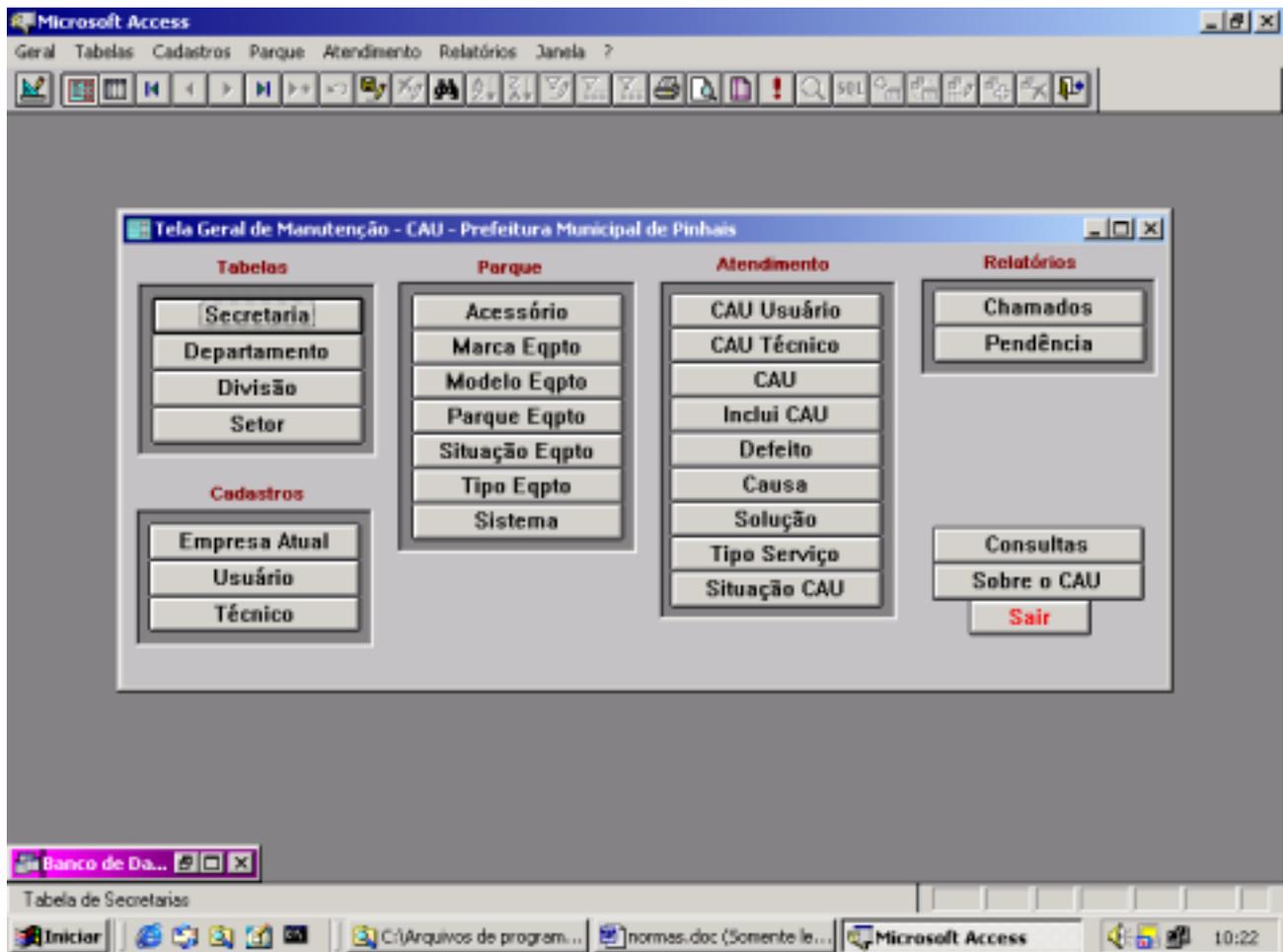


Na criação da barra de menu, a seqüência de exibição dos itens deve ser intuitiva ao usuário, evitando que ele fique procurando pelo que tem a fazer.

A criação das *hot keys* deve ser feita para as opções que tem uma utilização muito freqüente, pois de outra forma o usuário não irá utilizar e será gasto tempo desnecessário na criação da aplicação.

Quando uma opção exigir sub-menus, cuide para que estes sejam orientados por Micro Helps explicativos em linguagem simples e direta, pois o usuário quer resultados e não ser um *expert* em operação de sistemas.

**Tela Inicial do Sistema, onde são colocados os botões de opção e as opções de menu.**



### 3.3.3. Nomes de Programas

Não existe a filosofia de utilização de nomes de programas em Access. Assim como no Visual Basic, praticamente tudo é baseado em eventos e em controles.

Todavia, como será necessária a utilização de vários bancos de dados, tudo isto em rede, é praticamente certo que será necessário colocar os programas em um MDB em algum local da rede de comum acesso. Os demais MDBs ou o outro MDB, será(ão) colocado(s) em um endereço único, identificado por letras de drives de rede. **Isto deve ser definido antes pois pode mudar código.**

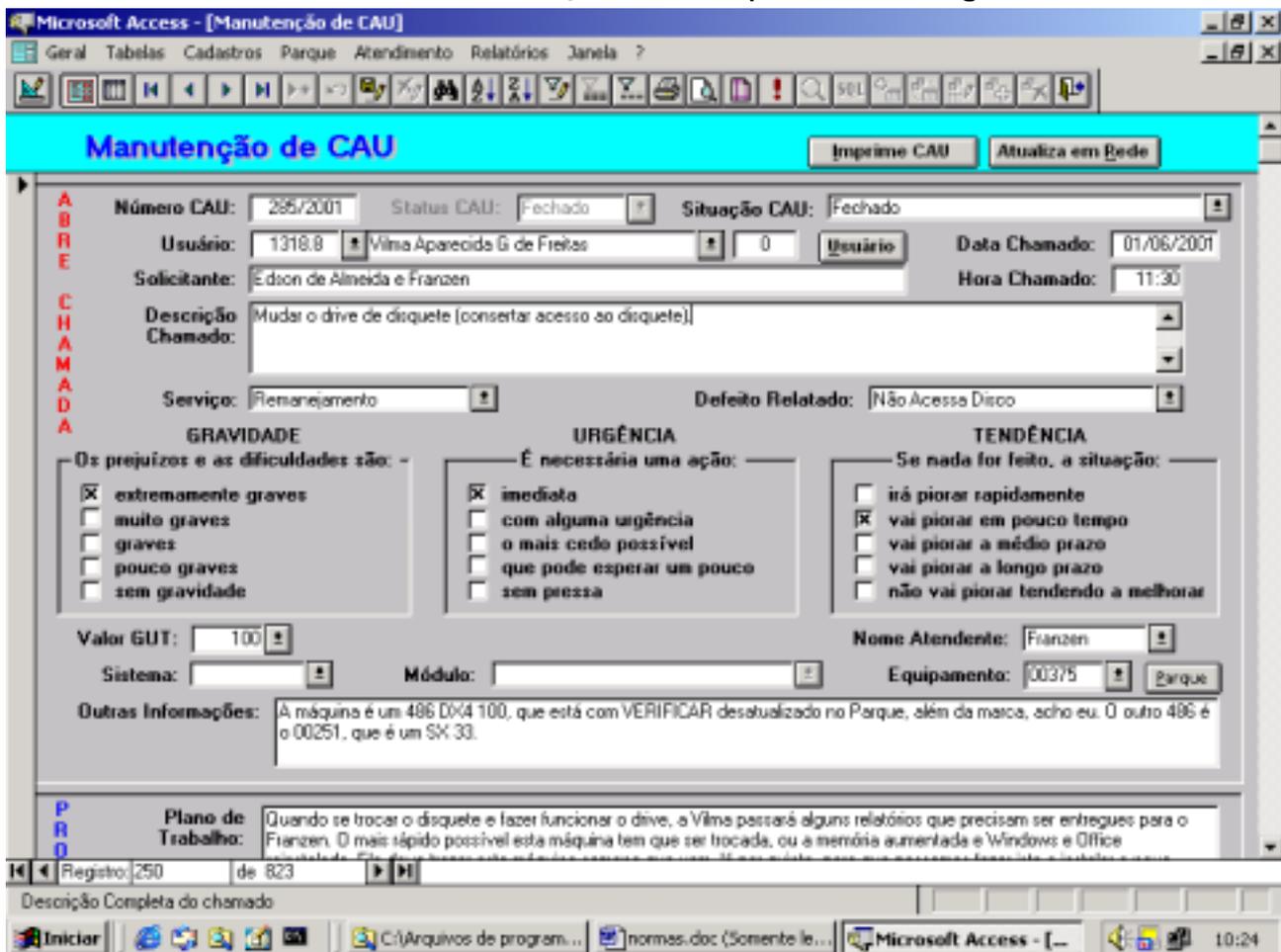
### 3.3.4. Tela de Manutenção

As telas de manutenção de tabelas devem apresentar os campos a serem imputados e/ou alterados, de forma clara, destacada e similar à seqüência do negócio do usuário.

Sempre que um campo puder ser exibido automaticamente invista no código para que o usuário ganhe confiança no sistema e invista ainda mais no uso de suas potencialidades.

O auxílio ao usuário deve ser rápido para aquelas operações que forem críticas e sujeitas a erros de digitação e/ou operação. O sistema deve, em resumo, ser previsível.

As telas de manutenção devem apresentar o seguinte:



Utilize, sempre que possível, a caixa de combinação (o controle que apresenta a seta para baixo sublinhada) para abrir pesquisa na lista de escolhas. Se você quer que o usuário visualize o código neste controle, deixe o controle mostrando o código. Todavia, na maioria das vezes mostre a descrição e não o código (Nome da Igreja em vez de Código da Igreja - por exemplo). Isto fará também com que o usuário digite letras iniciais de procura de descrição, e o controle irá mostrar o dado que o sistema encontrou.

Exemplo: Na Caixa de Combinação Logradouro, se você digitar R, o sistema irá se posicionar no primeiro R que ele encontrar. Se digitar A, ele irá se posicionar provavelmente em Av (de Avenida).

As cores não devem variar muito de uma tela para outra (de preferência nada). O fundo do cabeçalho da Tela deve ser azul claro e a letras "Pessoa" devem ser azul escuro e cinza respectivamente.

O fundo da tela de trabalho deve ser Cinza claro.

Os textos devem estar alinhados à direita, e as caixas de texto devem ficar alinhadas pela esquerda. Os valores devem ser alinhados à esquerda.

O tamanho da tela deve ser 800 por 600, padrão 256 cores.

### 3.4. Modelos de Telas e Programação

*Não importa se estamos pintando um quadro, escrevendo um poema ou esculpindo uma estátua, a simplicidade é a marca da mão do mestre.*  
Elsie de Wolfe

#### 3.4.1. A Interface Gráfica

A maioria dos programas para DOS apresentam interfaces construídas segundo a finalidade do sistema e segundo o "jeito" de cada programador. Mas sabemos que os dias da interface arbitrária estão contados. Depois que os usuários colocam as mãos no Windows ou no MAC, raramente estão dispostos a desistir deles. É como andar de Landau; você não irá se acostumar se trocar por um fusca.

As interfaces gráficas, a propósito, não foram criadas ao acaso. Envolveram uma quantidade de pesquisas de muitos anos com usuários de todo tipo e realmente funcionam. A padronização nas telas e o comportamento previsível dos programas só é facilmente possível com **linguagem orientada a objetos**, como a que o Windows foi escrita (C++ e outras).

Emular comportamentos conhecidos e familiares encontrados no mundo real é o que torna esse tipo de interface visual e **baseada em objetos** tão mais acessível aos usuários. O fato de que a interface é visual é outra razão para as pessoas gostarem dela. **Sutilezas de cor, textura visual, movimento e nuances na aparência geral dos objetos** podem comunicar uma grande quantidade de informações com muita rapidez. É isto que o Windows proporciona sem fazer segredos, e como consequência, o Access, que é baseado em Windows.

#### 3.4.2. A Interface Humana

Uma equipe de desenvolvimento de projeto é fortemente heterogênea. Ali existem analistas e programadores oriundos de diferentes empresas e ambientes. Utilizavam até aqui, as mais variadas linguagens e ferramentas, como por exemplo: Cobol, Mumps, Clipper, Zim, C, Visual Basic, Fox Pro e até geradores automáticos de sistemas.

Torna-se necessária, portanto, a criação de padrões de menu, telas, programas e documentação, sob pena de inviabilizar a integração atual dos sistemas e a manutenção futura.

Com a padronização pretende-se que todos os programas tenham aspecto semelhante, não sendo possível que haja uma uniformidade ampla, já que estaremos criando objetos para atenderem processos muitas vezes completamente distintos. Mas o que for semelhante ou idêntico, deverá ter a mesma funcionalidade (Ex.: Botão de **Atualizar** deve realmente atualizar os dados da tela, e não, por exemplo, fazer o **fechamento do mês**, "atualizando" uma base de dados.

Partir-se-á de linhas mestres e serão modificadas somente para satisfazer necessidades específicas.

Não se pretende ainda escrever um padrão final, livre de falhas, consistente e duradouro. Até porque, quando isto acontecer (se ocorrer), "teremos já morrido".

### 3.4.3. A Interface GUI

Os projetistas corporativos de interfaces gráficas com o usuário (GUI) frequentemente se deparam com uma faca de 2 gumes: uma interface com o usuário bem projetada

- aumenta a produtividade do projetista,
- reduz a necessidade de treinamento e suporte ao usuário final,
- além de reduzir a necessidade de manutenção (se houver manutenção, o seu tempo será reduzido pelo uso de uma boa interface.

Enquanto que uma interface parcamente projetada, coloca os usuários diante de uma superabundância de caixas de diálogo e controles casuais.

Devido à grande variedade de tipos de janelas e controles, o desenvolvedor de GUI bem sucedido precisa ter **uma noção exata do momento adequado para a utilização de um controle em detrimento de outro**. Da mesma forma que a fartura de fontes de letras baratas acarretou uma **tipografia carnavalesca**, o número e a qualidade dos controles definidos pelo desenvolvedor atual, freqüentemente conduzem a uma **utilização carnavalesca desses controles** (*penteadeira de madame*).

Pelo exposto anteriormente, a necessidade de linhas mestres fica evidente.

Como neste projeto deverá haver a criação de muitos controles e objetos novos completamente diferentes e além do que está definido aqui como linha mestre, descrevemos abaixo o que deve e não deve ser feito quando da criação de programas e objetos.

Note que isto ainda não libera que o desenvolvedor possa utilizar seu estilo livre de criação de GUI, uma vez que os programas devem passar pela aprovação do analista, do analista de O&M e do usuário. Mas é um norteador para você **não precisar refazer** objetos.

NÃO	SIM
1. Não crie aplicativos inconsistentes, imprevisíveis.	1. Opções de menu e botões de mesma aparência, ou até iguais, em uma e outra tela, devem se comportar de maneira semelhante e até idêntica. Botões de aparência diversa, devem se comportar de maneira diferente.
2. Não seja paternalista com o adiantado, nem "carrasco" com o principiante.	2. Crie mensagens na parte inferior da tela (micro help) que corresponda à posição do cursor (inclusive para campos).
3. Não crie um help (ajuda <i>on-line</i> ) sem significado. E nem procure dar significado escrevendo muitas linhas sobre o mesmo assunto.	3. A ajuda ao usuário deve realmente ajudar, e no menor espaço de tempo possível. Erros extremos como colocar todo o manual de usuário como help <i>on-line</i> , ou o contrário, resumindo de tal maneira os textos que se tornam incompreensíveis, não poderão ocorrer. Apêndices, Anexos, Referências Rápidas do manual do usuário, são partes características de um help.
4. Não crie muitas telas para executar uma determinada ação. Por exemplo: se o atendente ao efetuar o processo de fechar uma chamada, precisar sair da tela de fechamento de chamada para consultar cliente, item por contrato e mais alguma tabela, você projetou mal a aplicação!	4. Associe a uma tela, teclas de atalho ou teclas rápidas que acessem as opções de cadastro e consulta necessárias ao uso otimizado da tela. Ou ainda, torne sua aplicação inteligente, de tal modo que, ao colocar um novo código em um campo, e este não existe, a aplicação deverá perguntar se deseja incluir, e fará a inclusão para você. Evidente que todas as consistências deverão acontecer, como se estivesse na tela original de inclusão (ou traga a própria tela de manutenção, já que estamos em linguagem orientada a objetos).
5. Não exagere no número de informações disponíveis na tela, nem crie mais opções que o número normal de opções que o trabalho do usuário requer.	5. As opções de uma tela devem ser o reflexo do dia a dia de trabalho do usuário. Coloque na tela somente o que o usuário realmente precisa para efetivar seu trabalho da melhor maneira possível.
6. Não use jargão técnico nas mensagens. Para o usuário, mensagens como: "Erro 4143 no DMA" ou "Erro da I/O no handle de arquivo", só trazem confusão.	6. Crie objetos de mensagens padrão, personalizadas, únicas e em português.
7. Não crie opções e processos sem retorno do erro cometido.	7. Errar é humano, ainda mais na Informática. Portanto, o mais possível, deixe disponível opções que "desfaçam processos executados", que efetuem <i>rollback</i> nas linhas de código e nas ações executadas.
8. Não atender as opiniões do usuário é criar um sistema com seu ciclo de vida extremamente curto. A falta de participação do usuário faz com que ele assuma uma posterior postura de crítica e não de auxílio no processo.	8. O usuário do sistema que você está construindo é uma das pessoas que mais conhece a área de negócios abrangida pelo sistema. Por isso, é fundamental a sua opinião quanto às telas, tipo e formato de dados que serão inseridos, e sua aprovação nas mudanças, exclusão e criação de processos. Permita esta aprovação, inclusive de modo formal, com sua assinatura.
9. Nunca deixe o usuário pensar que sua máquina ou sua aplicação travou enquanto estava processando alguma função. A não ser que ela realmente travou.	9. Crie mensagens de processamento ou mude o tipo de cursor para alertar o usuário que o processamento de uma função ou um relatório está sendo executado. Mesmo que agora esta demora é de algumas frações de segundo, coloque assim mesmo, pois com o crescimento dos dados, o que agora é uma fração de segundos, no futuro poderá ser uma fração que necessita deste aviso.

### 3.4.4. Objetos de Controle

Este é um resumo dos objetos de controle utilizados pelo Access. Servirá para o auxiliar na escolha adequada do melhor controle para suas aplicações.

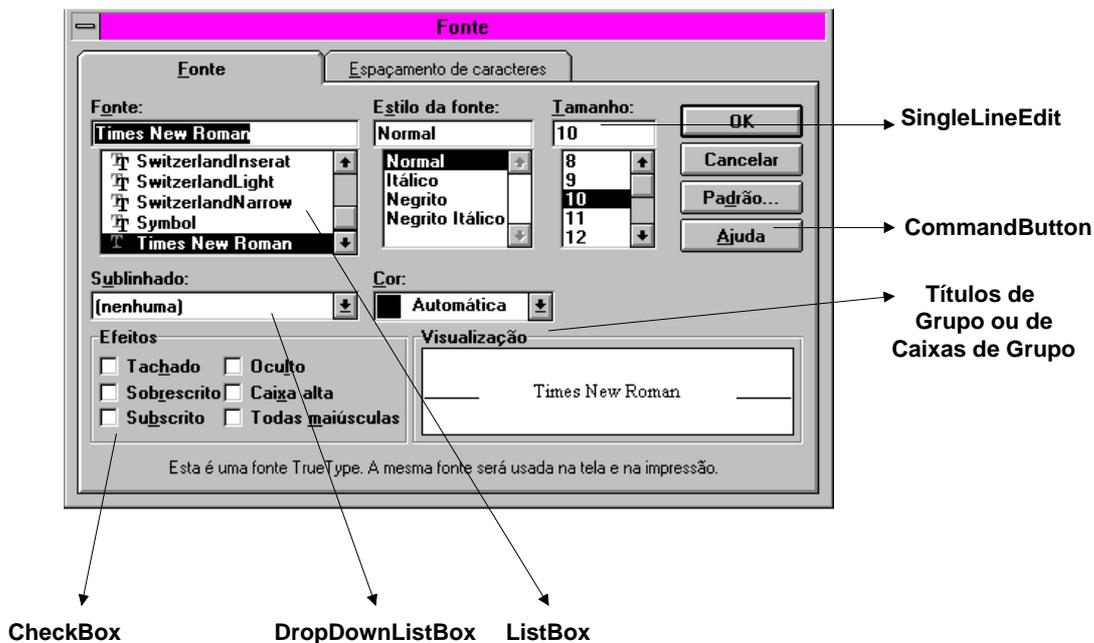
Controle	Nome	Descrição
	Botão Alternar	Dois botões mutuamente exclusiva, que permite uma escolha de opção quando clicado.
	Botão de Comando	Um botão com texto que ativa uma ação (ativa um script) quando clicado.
	Botão de Opção	Um pequeno círculo que é associado a uma opção de escolha pelo usuário.
	Caixa de Combinação	É uma caixa de texto, com uma seta que abre uma lista de opções para seleção.
	Caixa de Listagem	É uma caixa de texto, com uma seta que permite modificar o conteúdo do visor (informação que se apresenta ao usuário).
	Caixa de Texto	Um retângulo de texto que pode conter uma ou mais linhas de textos.
	Caixa de Verificação	Uma caixa quadrada associada a uma opção, que normalmente não é mutuamente exclusiva (pode ser selecionado mais de uma opção associada).
	Etiqueta	Mostra somente textos (por exemplo: cabeçalhos, avisos, mensagens...) que não podem ser alterados pelo usuário
	Gráfico	Uma caixa contendo um gráfico de representação de dados.
	Grupo de Opção	Um retângulo com um título que contém 2 ou mais controles de seleção, normalmente Botão de Opção.
	Linha	Uma linha sólida traçada.
	Moldura de Objeto	Uma caixa que abrigará um gráfico bitmap, ou qualquer outro objeto windows.
	Moldura de Objeto Vinculado	Uma caixa que abrigará um gráfico bitmap, ou qualquer outro objeto windows mas vinculado ao formulário ou relatório dito principal.
	Quebra de Página	Força que seja efetivado um salto de página na impressora.
	Retângulo	Um retângulo que pode associar um grupo de botões ou figuras de similar finalidade e escolha. Não esqueça de colocar o nome do grupo quando estiver associando vários controles como um grupo. <b>Aliás, coloque nomes em todos os controles</b> , utilizando o controle Etiqueta.
	Sub-Formulário ou Sub-Relatório	Uma caixa que permite que se insira um formulário ou relatório vinculado ao formulário ou relatório dito principal.

### 3.4.5. Não seja viciado em Controles

Os controles são elementos de interface com o usuário. Na maioria dos casos, servem para obter informações do usuário e apresentar dados sobre o status corrente do aplicativo. Por exemplo, as barras de paginação em um processador de textos não apenas permitem a movimentação dentro de um arquivo como também apresentam a posição corrente do usuário. Os botões de escolha única (Botões de Opção) apresentam a seleção corrente e permitem sua alteração.

Quase a totalidade das necessidades de controles em sistemas comerciais e científicos pode ser satisfeita pelo uso dos controles nativos do Access. Se houver ainda a requisição de outros controles, o Access trabalha com os numerosos controles do Visual Basic (.VBX). Entretanto não aconselhamos o uso de tais controles, para não cair no que chamamos de uso carnavalesco de objetos Windows.

Uso de poucos controles e cores de forma discreta é também o que apresenta o Windows 95, 98, 2000, é de onde tiramos algumas características para as linhas mestres deste trabalho.



#### Legenda

CommandButton	Botão de Comando
SingleLineEdit	Caixa de Texto
DropDownListBox	Caixa de Combinação
CheckBox	Caixa de Verificação
ListBox	Caixa de Listagem

### 3.4.6. Querida, Escondi o Controle

Imagine uma situação na qual seja necessário desabilitar controles e itens de menu não disponíveis (mantendo-os visíveis porém acinzentados), ou até mesmo ocultá-los. Em geral, manter um controle acinzentado é muito mais prudente. Os usuários **ficam normalmente confusos** quando itens de menu ou controles aparecem e desaparecem.

Os componentes de uma GUI só devem ser ocultados se o usuário não puder ter acesso a determinados controles e itens de menu.

Por exemplo, certos controles podem ser ocultados em uma janela ou caixa de diálogo baseada no nível de segurança do usuário durante a abertura de uma sessão.

Uma outra razão para não ocultar os controles desabilitados é quando a facilidade de **ajuda sensível ao contexto** está disponível. Neste caso, é possível que os usuários utilizem a tecla Tab para saltar para o controle desabilitado e pressionar F1 para obter informações e descobrir por que não podem ter acesso a esse controle.

O menu personalizado do Windows e dos produtos Office 2000, que desaparecem controles e opções de menu, e deixam disponíveis somente as opções mais utilizadas, é o que a Microsoft criou de pior nos últimos tempos. Só quem ministra aulas de informática consegue perceber o que estes tipos de menus impõem aos usuários iniciantes e aos usuários avançados. Os usuários iniciantes ficam procurando opções que não aparecem na tela. Os usuários avançados vão até as opções robótica e rapidamente, mas não as encontram. Em ambos os casos o que gera é a improdutividade. Acho que somente ganharam com os menus personalizados, alguns usuários de nível médio, que podem se acostumar mais rapidamente com estes tipos de menu.

### 3.4.7. Conclusão

Se Você deseja	Utilize	Mas no máximo
Executar uma ação	Botão de Comando	Seis por caixa de Diálogo
Escolher entre algumas opções mutuamente exclusivas	Botão de Opção	Seis por grupo
Escolher uma ou mais entre algumas opções não exclusivas	Caixa de Verificação	Dez por Grupo
Obter do usuário dados arbitrários em forma de texto	Caixa de Texto	
Escolher entre opções mutuamente exclusivas ou não exclusivas	Caixa de Listagem	50 itens, apresentando 8 a 10 de cada vez
Escolher entre opções mutuamente exclusivas utilizando um mínimo de espaço, ou inserir um novo valor	Caixa de Combinação	20 itens, apresentando um de cada vez <sup>2</sup>
Agrupar controles relacionados, especialmente botões de Opção e caixas de verificação	Grupo de Opção	
Nomear controles que não possuem seus próprios títulos (Botão de Opção, por exemplo)	Etiqueta	

A tabela acima resume os pontos que devem ser considerados ao utilizar controles em caixas de diálogo. Embora os controles do Windows possam freqüentemente acomodar um grande número de opções, não é uma boa idéia atingir este limite, sacrificando seus usuários. Observe, por exemplo, o número máximo de opções sugerido para uma lista de alternativas. Os usuários ficam frustrados após pressionar PgDn ou utilizar a barra de paginação para navegar em uma lista de mais de 50 itens.

**Todavia o Access utiliza uma pesquisa automática de valores dentro de uma Caixa de Combinação. Isto possibilita que você possa utilizar este controle em listas de mais de 50 opções que possam ser pesquisadas digitando-se as primeiras letras do nome ou primeiros números a serem pesquisados.**

A escolha adequada da janela e dos elementos de controle para um aplicativo Windows define uma base sólida para o restante do aplicativo. Se você fizer as escolhas corretas e não tentar exigir demais da tecnologia da interface, será recompensado com mais produtividade e menos problemas de suporte, como já citamos. Por outro lado, as escolhas mal feitas são responsáveis pelos **pesadelos** das interfaces gráficas.

Acorde...

<sup>2</sup> Devido à alta velocidade das máquinas e dos programas gerenciadores de banco de dados existentes, estas caixas de combinação têm sido amplamente utilizadas inclusive em listagem com centenas e milhares de nomes.

## 3.5. Relatórios

### 3.5.1. Emissão na Tela

Salvo raras exceções, todos os relatórios devem ter sua visualização também na tela.

Todavia, a criação de relatórios com visualização na tela deve ser feita com alguns cuidados especiais, principalmente no que tange à navegação e facilidade de interpretação e análise dos dados apresentados.

A limitação imposta pelo tamanho da tela deve ser contornada com alguns artifícios como por exemplo:

- parâmetros/filtro dos dados para o usuário determinar o contexto de dados desejado,
- rolagem da tela nos sentidos horizontal e vertical,
- entre outros.

### 3.5.2. Emissão em Papel

Tradicional na sua forma, as emissões em papel ganham uma nova dimensão com as impressoras de tecnologia jato de tinta colorida e laser. A qualidade visual da maioria dos relatórios deverá ser melhor pensada pelo analista, pois o usuário não aceitará tão facilmente a idéia de obter um relatório com qualidade inferior ao que ele vê na sua tela gráfica e colorida.

O Access permite que cada formulário criado para a tela possa ser impresso em papel. Todavia, existirão casos que seja necessário a criação de relatórios específicos. Nestes casos, exageros à parte, a idéia é não oferecer relatórios assim:

Empresa XYZ de Ferros Ltda.		Curitiba, 10 de março de 2002	
Fornecedores Com Entregas em Atraso		Pg. 01	
Código	Nome	Prometido Para	Dias de Atraso
20011	Fornecedor de Pregos	02/10/1994	8
89321	Fornecedor de CPU XT Turbo	05/10/1994	5
45398	Fornecedor de Monitor Verde	06/10/1994	4
Total de Fornecedores em Atraso: ..... 3			

Mas oferecer algo pelo menos assim:

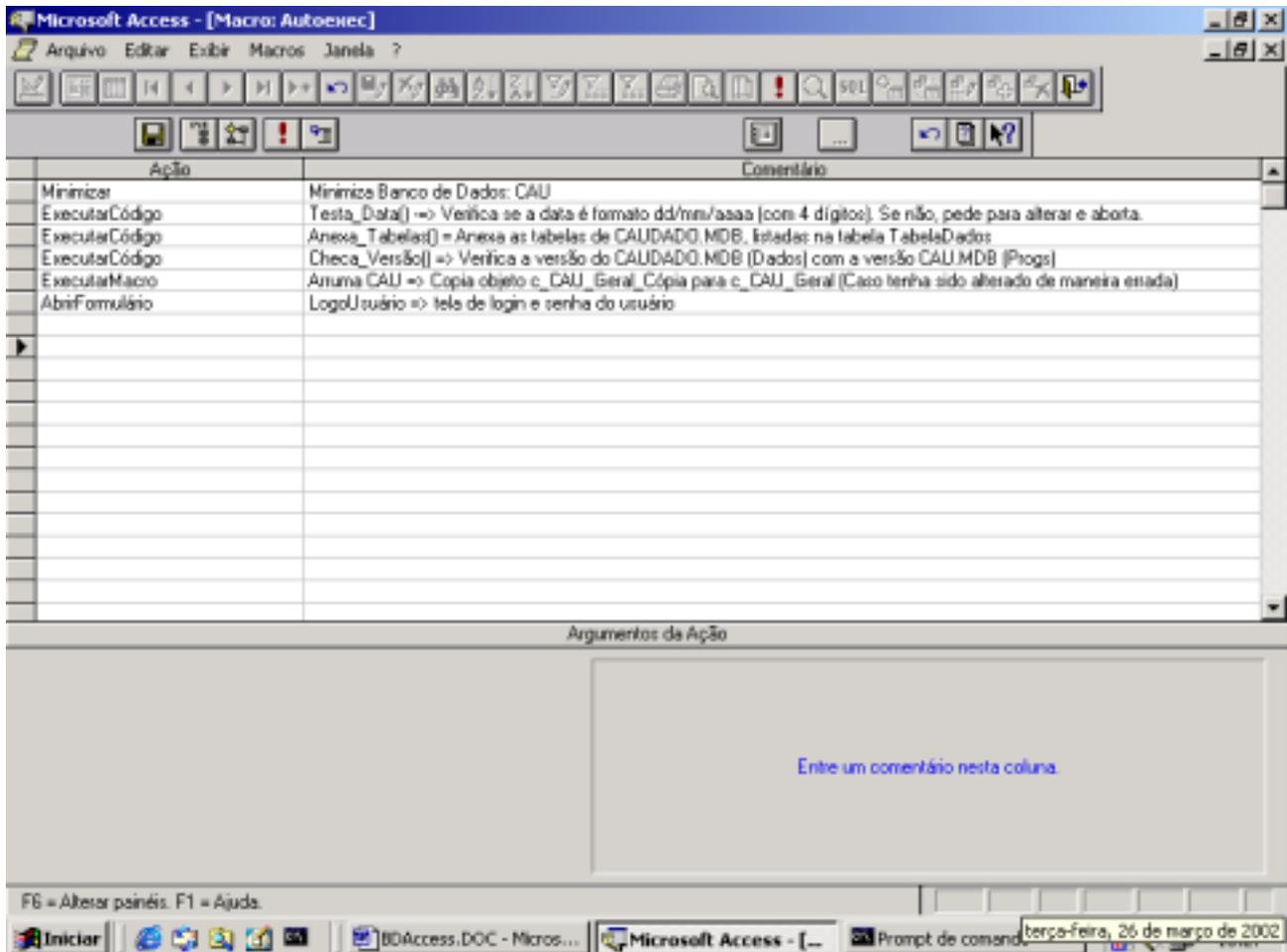
<b>EMPRESA XYZ DE FERROS Ltda.</b>		<b>Curitiba, 10 de Março de 2002</b>	
Fornecedores Com Entregas em Atraso		Pg. 01	
Código	Nome	Prometido Para	Dias de Atraso
20011	Fornecedor de Pregos	02/10/1994	8
89321	Fornecedor de CPU XT Turbo	05/10/1994	5
45398	Fornecedor de Monitor Verde	06/10/1994	4
Total de Fornecedores em Atraso: ..... 3			

## ANEXO 1 - Tabela de Mnemônicos

<i>Mnemônico</i>	<i>Descrição</i>	<i>Mnemônico</i>	<i>Descrição</i>
<b>Abrev</b>	Abreviatura	<b>Ref</b>	Referência
<b>Armaz</b>	Armazenamento	<b>Resp</b>	Responsável
<b>Auto</b>	Automático	<b>Seq</b>	Seqüência
<b>BC</b>	Base de Cálculo	<b>Sol</b>	Solicitação
<b>BD</b>	Banco de Dados	<b>Tp</b>	Tipo
<b>Bon</b>	Bonificação	<b>Transp</b>	Transporte
<b>Canc</b>	Cancelamento	<b>Vcto</b>	Vencimento
<b>CFO</b>	Código Fiscal de Operação	<b>Vlr</b>	Valor
<b>Class</b>	Classificação		
<b>Cod</b>	Código		
<b>Compl</b>	Complemento		
<b>Cond</b>	Condição		
<b>Cons</b>	Consumidor		
<b>DDD</b>	Discagem Direta à Distância		
<b>Descr</b>	Descrição		
<b>Desloc</b>	Deslocamento		
<b>Dimen</b>	Dimensiona(do)/(mento)		
<b>Doc</b>	Documento		
<b>Dpto</b>	Departamento		
<b>Dt</b>	Data (DDMMAAAA)		
<b>Dup</b>	Duplicata		
<b>DV</b>	Dígito Verificador		
<b>Ent</b>	Entrada		
<b>Fat</b>	Faturamento		
<b>Flag</b>	Flag, Status, Switch		
<b>Fone</b>	Telefone		
<b>Forn</b>	Fornecedor		
<b>Func</b>	Funcionário		
<b>Hr</b>	Hora (HHMMSS)		
<b>ICMS</b>	Imposto sobre Circ. de Mercad.		
<b>Ind</b>	Indicador		
<b>Inscr</b>	Inscrição		
<b>Inst</b>	Instala(ção)/(das)		
<b>IPI</b>	Imposto sobre Prod. Industr.		
<b>Km</b>	Kilômetro		
<b>Lab</b>	Laboratório		
<b>Mant</b>	Mantenedor		
<b>Merc</b>	Mercadoria		
<b>Mnm</b>	Mnemônico		
<b>MTBF</b>	Tempo Médio Entre Falhas		
<b>MTRR</b>	Tempo Médio de Reparos		
<b>Mvto</b>	Movimento		
<b>NAF</b>	Não Afeto		
<b>NF</b>	Nota Fiscal		
<b>Num</b>	Número		
<b>Obs</b>	Observação		
<b>Perc</b>	Percentual		
<b>Pgto</b>	Pagamento		
<b>Qtd</b>	Quantidade		
<b>Recebi</b>	Recebimento		

## Anexo 2 – Códigos Fontes

Macro Autoexec do Sistema CAU – primeira a ser executada



## Script (módulo) Anexa\_Tabelas() do Sistema CAU

```
Function Anexa_Tabelas ()
On Error GoTo Err_Parâmetro_CEP
Dim qQuery As QueryDef, rSnap As Recordset, MeuBD As Database
Dim MeuBD1 As Database, ConectarA As String
Dim MinhaÁreaDeTrabalho As Workspace
Dim MinhaDefConsulta As QueryDef
Dim MeuBancoDeDados As Database
Dim MinhaDefTabela As TableDef
Dim Critério As String, VarAux As String
Dim I As Integer, J As Integer
Dim Farei As String, Tabela As String, DescrParametro_Aux As String

' Obtém área de trabalho padrão.
Set MinhaÁreaDeTrabalho = DBEngine.Workspaces(0)

' Obtém banco de dados atual.
Set MeuBancoDeDados = MinhaÁreaDeTrabalho.Databases(0)

' Acha tabela de dados corrente
Set MeuBD = CurrentDB()

' Habilita Cursor como Ampulheta
DoCmd Hourglass True

' Desabilita mensagens de aviso do sistema
DoCmd SetWarnings False

' Deleta Consulta de Parametro
Nada = Apaga("c_ParametroAuxiliar")

' O número do Parametro de para o local dos dados
Set qQuery = MeuBD.CreateQueryDef("c_ParametroAuxiliar")
qQuery.SQL = "SELECT CodParametro, DescrParametro FROM Parametro WHERE CodParametro = 1;"
Set rSnap = MeuBD.OpenRecordset("c_ParametroAuxiliar", DB_OPEN_SNAPSHOT)

' Se não existe este Parametro
If rSnap.RecordCount = 0 Then
MsgBox "Não será possível continuar. Não existe informação em Parâmetro (CodParametro = 1) sobre a localização do Arquivo DADO.MDB. Chame a Consultoria.", 48, "Aviso de Saída do Sistema"
Close
DoCmd RunMacro "Sair"
End If

If IsNull(Trim(rSnap!DescrParametro)) Then
MsgBox "Não será possível continuar. Não existe informação em Parâmetro (DescrParametro) sobre a localização do Arquivo DADO.MDB. Chame a Consultoria.", 48, "Aviso de Saída do Sistema"
Close
DoCmd RunMacro "Sair"
End If

DescrParametro_Aux = Trim(rSnap!DescrParametro)
' Verifica se o arquivo especificado em Parametro é válido (se encontra no diretório especificado)
If Se_Existe(DescrParametro_Aux) = 0 Then
If MsgBox("O drive, diretório ou nome de arquivo não são válidos. Deseja alterar agora a letra desta informação?", 36, "Aviso para Entrada de Informações") = 6 Then
Do While 0 = 0
DescrParametro_Aux = DescrParametro_Aux + Space(100)
DescrParametro_Aux = InputBox("Entre com o drive, o Diretório e o arquivo DADO.MDB", "Entrada de Informações", DescrParametro_Aux)
If Se_Existe(Trim(DescrParametro_Aux)) = 1 Then
```

```
' Atualiza a descrição de Parâmetro com o novo dado
DoCmd RunSQL "UPDATE Parametro SET DescrParametro = '" & Trim(DescrParametro_Aux) &
"' WHERE CodParametro = 1;"

Exit Do
Else
MsgBox "O drive, diretório ou nome de arquivo não são válidos. Saindo do Sistema!",
48, "Aviso de Saída do Sistema"
Close
DoCmd RunMacro "Sair com Problemas"
End If
Loop
Else
' Desabilita Cursor como Ampulheta
DoCmd Hourglass False

' Desabilita mensagens de aviso do sistema
DoCmd SetWarnings True

MsgBox "Não será possível continuar. O programa será abortado. Entre no programa segurando
a tecla SHIFT e mude a localização do Arquivo DADO.MDB na tabela Parametro.", 48, "Aviso de Saída do
Sistema"
Close
DoCmd RunMacro "Sair com Problemas"
End If
End If

' *****
' Atualiza data e hora desta máquina com a data e hora do servidor de dados
' Se for mudado o nome do arquivo de dados (caudado.mdb=12 bytes), é necessário rever a rotina
VarAux = Left(Trim(DescrParametro_Aux), Len(Trim(DescrParametro_Aux)) - 12)
Nada = Shell(VarAux + "\SYNCDAT.EXE", 4)

' Volta para o Access
AppActivate "Microsoft Access"

' Local de onde virão as tabelas de dados
ConectarA = ";DATABASE=" + Trim(DescrParametro_Aux)

' Deleta Consulta de Parametro
Nada = Apaga("c_ParametroAuxiliar")

' Deleta Query Auxiliar de TabelaDados
Nada = Apaga("c_Tabela_Dados")

' Cria Query para Tabela Dados
Set qQuery = MeuBD.CreateQueryDef("c_Tabela_Dados")
qQuery.SQL = "SELECT NomeTabela FROM TabelaDados ORDER BY NomeTabela;"
Set rSnap = MeuBD.OpenRecordset("c_Tabela_Dados", DB_OPEN_SNAPSHOT)

If rSnap.RecordCount = 0 Then
MsgBox "Não será possível continuar. Não há Tabelas na entidade TabelaDados.", 48, "Aviso de
Saída do Sistema"
DoCmd RunMacro "Sair"
End If

' *****
' Anexa as Tabelas da entidade TabelaDados
' Faz a partir daqui, quantas vezes forem as Tabelas gravadas em TabelaDados
' *****
Do Until rSnap.EOF
Tabela = Trim(rSnap!NomeTabela)
Farei = "Anexar"
```

```
' Procura por Tabela de Dados
For I = 0 To MeuBancoDeDados.TableDefs.Count - 1
    If Tabela = MeuBancoDeDados.TableDefs(I).Name Then
        Farei = "Deletar"
        Exit For
    End If
Next I

' Deleta a tabela, pois ela existe
If Farei = "Deletar" Then
    MeuBancoDeDados.TableDefs.Delete Tabela
End If

Set MinhaDefTabela = MeuBancoDeDados.CreateTableDef(Tabela)

' Anexa tabela Parâmetro ao banco de dados
MinhaDefTabela.Connect = ConectarA
MinhaDefTabela.SourceTableName = Tabela

' Anexar a tabela
MeuBancoDeDados.TableDefs.Append MinhaDefTabela
ConectaFonte = True

' Localiza próxima ocorrência.
rSnap.MoveNext
Loop

' Deleta Query Auxiliar de TabelaDados
Nada = Apaga("c_Tabela_Dados")

Exit_Parametro_CEP:
' Desabilita Cursor como Ampulheta
DoCmd Hourglass False

' Desabilita mensagens de aviso do sistema
DoCmd SetWarnings True

Exit Function

Err_Parametro_CEP:
' Desabilita Cursor como Ampulheta
DoCmd Hourglass False

' Desabilita mensagens de aviso do sistema
DoCmd SetWarnings True

MsgBox Error$
MsgBox "Não será possível continuar a função Anexa_Tabelas. Tente executar novamente. Se
persistir o erro, chame a Consultoria.", 48, "Aviso de Saída do Sistema"
DoCmd RunMacro "Sair"

End Function
```