



# Descrição da Solução NAVS

(Navegador Skytef)

## SKYTEF

Versão 1.8.8 – 09/01/2015

## Sumário

<b>1.0</b>	<b>Introdução</b>	<b>4</b>
<b>2.0</b>	<b>Descrição Inicial</b>	<b>5</b>
<b>3.0</b>	<b>Detalhamento dos Comandos</b>	<b>6</b>
<b>3.1</b>	<b>CONSOLE</b>	<b>7</b>
<b>3.2</b>	<b>GET</b>	<b>9</b>
3.2.1	CARTÃO MAGNÉTICO	10
3.2.2	ENTRADA NUMÉRICA	11
3.2.3	ENTRADA ALFANUMÉRICA	12
3.2.4	ENTRADA NUMÉRICA SECRETA	13
3.2.5	ENTRADA NUMÉRICA COM CASAS DECIMAIS	14
3.2.6	AGUARDA TECLA ESPECIAL	15
3.2.7	AGUARDA TECLA	16
3.2.8	RETORNA O STATUS ATUAL DA COMUNICAÇÃO NO POS	17
3.2.9	CONTROLA SESSÃO	18
3.2.10	VALOR MOEDA	19
3.2.11	NUMERO DE SÉRIE	20
3.2.12	CÓDIGO DE BARRAS	21
3.2.13	PORTA SERIAL	22
3.2.14	CONTROLES DA PORTA SERIAL	23
3.2.15	VERSÃO	24
3.2.16	VERSÃO DO SISTEMA OPERACIONAL	25
3.2.17	MODELO	26
3.2.18	IDENTIFICAÇÃO DO CHIP	27
3.2.19	MIFARE	28
3.2.20	LEITURA DE DADOS DO MIFARE	29
3.2.21	TOUCH (somente no Vx680)	30
3.2.22	DATA E HORA	32
3.2.23	CPF	33
3.2.24	HOUR	34
3.2.25	DATA	35
3.2.26	CNPJ	36
3.2.27	APDUCHIP	37
3.2.28	DESNBS	38
<b>3.3</b>	<b>DELAY</b>	<b>39</b>
<b>3.4</b>	<b>SELECT</b>	<b>40</b>
<b>3.5</b>	<b>TONE</b>	<b>42</b>
<b>3.6</b>	<b>PRINTER</b>	<b>43</b>
<b>3.7</b>	<b>CHGCONFNT</b>	<b>44</b>
<b>3.8</b>	<b>PRNFNT</b>	<b>45</b>
<b>3.9</b>	<b>WRITE_AT</b>	<b>46</b>
<b>3.10</b>	<b>DRAW (Somente para Vx680)</b>	<b>47</b>
<b>3.11</b>	<b>CHGPRNFNT</b>	<b>48</b>
<b>3.12</b>	<b>PRNLOGO</b>	<b>50</b>

## Especificação de Interface

3.13	CONLOGO	51
3.14	CODEBAR – Impressão	52
3.15	GERADOR DE QR CODES	53
3.16	Criar Código de barras 3 de 9	54
3.17	Criar Código de Barras ITF (2 de 5)	55
3.18	TELECARGA	56
3.19	SAÍDA SERIAL	57
3.20	SET TIME	58
3.21	KEYBOARD	59
3.22	SET PORTA SERIAL	60
3.23	CAPTURA ÚNICA DE ENTRADA	61
3.24	INIT	62
3.25	POST	63
3.26	FILE	64
3.27	REMOVE ARQUIVO NO POS	65
3.28	RESET	66
3.29	CONFIG_NAVS	67
3.30	IMPRESSÃO DE BOLETO	70
3.31	ENVIO DE ARQUIVO	72
	<i>Importante: Este comando substitui o POST, ou seja, ele é o último comando que deve ser inserido na lista da sessão.</i>	72
3.32	GRAVAÇÃO EM ARQUIVO DAS CAPTURAS – POSTOFF (uso para offline)	73
3.33	IMPRESSÃO OFFLINE - PRINTEROFF	74
3.34	HTMLDEBUG	75
3.35	VALIDAÇÃO OFFLINE - VALIDATEOFF	76
3	<i>OFFLINE com LUA / SQLite</i>	78
4	<i>Compilador Lua</i>	79
5	<i>MIFARE – leitura e escrita</i>	80
6	<i>Descrição da Mensagem do POS ao Servidor</i>	81
7	<i>Exemplo de uso</i>	82
8	<i>Configuração do NAVS</i>	84
9	<i>Status do Terminal</i>	90
10	<i>Histórico das Versões</i>	91

### 1.0 Introdução

NAVS é a plataforma de desenvolvimento da Skytef que proporciona às redes de POS flexibilidade nas aplicações. Possibilita rápido desenvolvimento de novas funcionalidades e serviços, atendendo aos mais diversos mercados como redes adquirentes, *private label*, bancos, financeiras, correspondente bancário, fidelidade, entre outros.

NAVS é um navegador que permite que o POS acesse o servidor, utilizando o mesmo tipo de conexão das transações. O acesso ao servidor possibilita realizar atualizações das aplicações e implementar novos serviços, o que pode ser feito em toda rede ou em um único terminal, de acordo com a necessidade do gestor.

#### Vantagens

- Reduz tempo nos processos de certificação de aplicações
- Transforma a rede de POS numa rede de serviços
- Possibilita ativação/desativação de serviços nos terminais POS remotamente
- Facilita o desenvolvimento, pois é uma plataforma amigável e intuitiva

## **2.0 Descrição Inicial**

NAVS é uma aplicação que será carregada no POS e interpretará comandos recebidos do Servidor.

Na carga inicial, deverá existir um arquivo XML contendo as informações da primeira comunicação do POS para o Servidor, como uma inicialização.

Na resposta, o Servidor deverá enviar os comandos XML para terminar a inicialização e iniciar a primeira transação.

O POS capturará os dados desta transação e enviará ao Servidor que deverá responder com os comandos para terminar a transação e iniciar a próxima.

### **IMPORTANTE:**

O NAVS utiliza o protocolo HTTP ou HTTPS.

O campo mandatório que deve ser enviado pelo Host no protocolo HTTP é o “Content\_Length”.

### 3.0 Detalhamento dos Comandos

Através de 'scripts', o servidor designará os comandos a serem executados pelo POS, como display, impressora, cartão, captura de dados, etc.

Obs.: Quando tiver variáveis nos comandos para receber algum retorno, não coloque variáveis no seu código com o mesmo nome.

Exemplo: se o comando for `<GET NAME=CAMPO1 TYPE=VALUE SIZE=X COL=Y LIN=Z>`, não coloque uma variável NAME para NAME (`<GET NAME=NAME TYPE=VALUE SIZE=X COL=Y LIN=Z>`).

Obs2.: Para os comandos que utilizam os parâmetros LINHA e COLUNA na tela do POS, é importante ressaltar que linha e coluna iniciais possuem inicialmente os respectivos valores : 1 e 0, portanto se quiser posicionar algo na primeira linha e na primeira coluna no console devemos atribuir 1 para linha e 0 para coluna.

A seguir o detalhamento de alguns 'scripts':

## Especificação de Interface

### 3.1 CONSOLE

<CONSOLE NOCLS=x> inicia texto para o display,  
</CONSOLE> finaliza texto para o display.

A mudança de linha é realizada pelo comando <BR>.

O parâmetro NOCLS determina se uma imagem que estiver no display será apagada ou não com a atualização do texto na tela.

X representa os valores que podem ser assumidos 0=não 1=sim.

No Vx520, caso a mensagem ultrapassar o número de linhas da Console, use as teclas de Scroll (definidas no menu de configuração), para navegar pelas linhas.

Obs.: Qualquer outro comando deverá ser utilizado fora de “<CONSOLE NOCLS=x>yyy</CONSOLE>”, entre eles haverá SOMENTE texto para o display.

Obs.2: Caracteres especiais devem ser passados com o respectivo número hexadecimal para que possam ser interpretados corretamente e ter a exibição desejada na tela. Para exibir na tela o caractere de seta para cima utilize “\x14\x15”, para exibir seta para baixo utilize “\x16\x17”, para exibir seta para direita utilize “\x18”, e exibir seta para esquerda utilize “\x19”.

Abaixo um exemplo de como inserir seta para cima no console :

```
<CONSOLE NOCLS=0>\x14\x15</CONSOLE>
```

A seguir tabelas com alguns valores correspondentes em hexadecimal de caracteres especiais:

Caractere	Valor Hexadecimal correspondente
!	21
"	22
#	23
\$	24
%	25
&	26
'	27
(	28
)	29
*	2A
+	2B
,	2C
-	2D
÷	F7
.	2E
:	3A
;	3B
=	3D
>	3E

Caractere	Valor Hexadecimal correspondente
?	3F
@	40
[	5B
\	5C
]	5D
^	5E
`	60
{	7B
	7C
}	7D
~	7E
ñ	F1
✓	8B
ı	A1
¢	A2
£	A3
¥	A5
§	A7
©	A9

## Especificação de Interface

Caractere	Valor Hexadecimal correspondente
ı	AC
ë	EB
®	AE
°	B0
à	E0
á	E1
â	E2
ã	E3
ä	E4
å	E5
μ	B5
¶	B6
¿	BF
À	C0
Á	C1
Â	C2
Ã	C3

Caractere	Valor Hexadecimal correspondente
Æ	C6
Ç	C7
È	C8
É	C9
Ê	CA
Ï	CC
Í	CD
İ	CE
Ĭ	CF
Ñ	D1
æ	E6
ç	E7
è	E8
é	E9
ê	EA
À	C4
Á	C5

\*É importante salientar que esses valores são válidos apenas para a fonte "DEFAULT" dos POS Verifone (Vx520 e Vx680). Para outras fontes Verifone e aparelhos Ingenico a correspondência relacionada nas tabelas acima não é garantida.

### 3.2 *GET*

<GET...> indica a captura de informações no POS, a seguir os parâmetros mandatórios deste comando:

TYPE discrimina o tipo da informação a ser capturada.

NAME discrimina o identificador do campo que contém a informação capturada.

## Especificaão de Interface

---

### 3.2.1 CARTÃO MAGNÉTICO

Para capturar os dados do cartão, através da leitura da tarja magnética, devemos passar para o parâmetro TYPE o argumento CARD.

Neste caso, haverá outro parâmetro mandatório: TRACKS, que especifica quais trilhas do cartão deverão ser coletadas.

O comando deve ser passado conforme o esquema abaixo, notando que a ordem dos parâmetros não altera o resultado.

```
<GET TYPE=CARD NAME=CARTAO1 TRACKS=X>
```

'X' pode ser qualquer valor de 0 a 7, da seguinte forma:

0-não envia nenhuma trilha.

1-envia trilha 1

2-envia trilha 2

4-envia trilha 3

3-envia trilhas 1 e 2

5-envia trilhas 1 e 3

7-envia trilhas 1, 2 e 3

**IMPORTANTE:** As informações do cartão são enviadas ao Servidor usando "URLENCODE":

Retorna uma string em que todos os caracteres não-alfanuméricos com exceão de `-_.` são substituídos com um sinal de por cento (`%`) seguido por dois dígitos hexadecimais e espaços codificados como um sinal de (`+`).

## Especificação de Interface

---

### 3.2.2 ENTRADA NUMÉRICA

Para capturar uma entrada numérica, não secreta, por meio do teclado devemos passar para o parâmetro TYPE o argumento FIELD.

Neste caso haverá os parâmetros mandatórios: SIZE, COL, LIN.  
O parâmetro NOENTER é opcional.

SIZE especifica o tamanho em caracteres do campo a ser capturado.

COL especifica a coluna da esquerda para direita onde o primeiro caractere capturado será mostrado no display.

LIN especifica a linha de cima para baixo onde o primeiro caractere capturado será mostrado no display.

NOENTER=1 especifica que o comando não aguardará o <ENTER> no final da entrada.

Devemos notar que na linha onde será mostrado o campo, todo texto previamente existente será apagado.

O comando deverá ser passado conforme o esquema abaixo:

```
<GET TYPE=FIELD NAME=CAMPO1 SIZE=X COL=Y LIN=Z NOENTER=1>
```

### 3.2.3 ENTRADA ALFANUMÉRICA

Para capturar uma entrada alfanumérica, não secreta, por meio do teclado devemos passar os mesmos argumentos da Entrada Numérica acima, incluindo o parâmetro ALPHA=1, como abaixo:

```
<GET TYPE=FIELD ALPHA=1 NAME=CAMPO1 SIZE=X COL=Y LIN=Z NOENTER=1>
```

## Especificação de Interface

---

### 3.2.4 ENTRADA NUMÉRICA SECRETA

Para capturar uma entrada numérica, secreta, por meio do teclado devemos passar para o parâmetro TYPE o argumento PASS.

Neste caso haverá os parâmetros mandatórios: SIZE, COL, LIN.

SIZE especifica o tamanho em caracteres do campo a ser capturado.

COL especifica a coluna da esquerda para direita onde o primeiro asterisco referente ao primeiro caractere capturado será mostrado no display.

LIN especifica a linha de cima para baixo onde o primeiro asterisco referente ao primeiro caractere capturado será mostrado no display.

Devemos notar que na linha onde será mostrado o campo, todo texto previamente existente será apagado.

O comando deverá ser passado conforme o esquema abaixo:

```
<GET TYPE=PASS NAME=CAMPO2 SIZE=X COL=Y LIN=Z>
```

### 3.2.5 ENTRADA NUMÉRICA COM CASAS DECIMAIS

Para capturar uma entrada numérica, não secreta, e que tenha a formatação de casas decimais, por meio do teclado devemos passar para o parâmetro TYPE o argumento NONINT.

Neste caso haverá os parâmetros mandatórios: SIZE, COL, LIN, DECIMALS.  
O parâmetro NOENTER é opcional.

SIZE especifica o tamanho em caracteres do campo a ser capturado.

COL especifica a coluna da esquerda para direita onde o primeiro caractere capturado será mostrado no display.

LIN especifica a linha de cima para baixo onde o primeiro caractere capturado será mostrado no display.

DECIMALS especifica o número de casas decimais.

NOENTER especifica que o comando não aguardará o <ENTER> no final da entrada.

Devemos notar que na linha onde será mostrado o campo, todo texto previamente existente será apagado.

O comando deverá ser passado conforme o esquema abaixo:

```
<GET TYPE=NONINT NAME=CAMPO1 SIZE=X COL=Y LIN=Z DECIMALS=W NOENTER=1>
```

## Especificação de Interface

### 3.2.6 AGUARDA TECLA ESPECIAL

Para aguardar o pressionamento de teclas especiais, devemos passar para o parâmetro TYPE o argumento SPECIALKEY, conforme demonstrado a seguir :

<GET TYPE=SPECIALKEY NAME=CAMPO1>

Os valores retornados pelas teclas são os relacionados na tabela a seguir :

TECLA	STRING RETORNADA
F1	FA
F2	FB
F3	FC
F4	FD
Roxa 1	E1
Roxa 2	E2
ALPHA	8F
Roxa 3	E3
Roxa 4	E4
*	AA
#	A3
BackSpace (Amarela)	88
Confirma (Verde)	8D

Obs.: Nos modelos VX680 da Verifone somente as teclas “BackSpace” e “Confirma” podem ser utilizadas.

## Especificação de Interface

---

### 3.2.7 AGUARDA TECLA

Para aguardar o pressionamento da tecla Verde, sem enviar qualquer informação ao Servidor, devemos passar para o parâmetro TYPE o argumento ANYKEY.

Note que neste caso o parâmetro NAME não deverá constar no comando.

<GET TYPE=ANYKEY>

### 3.2.8 RETORNA O STATUS ATUAL DA COMUNICAÃO NO POS

Para saber o status da comunicaão basta usar esse comando que ele retornar uma string. O contedo retornado  o mesma que aparece no lado direito da linha de status. A seguir a forma de utilizaão do comando :

<GET TYPE=STSCOMM NAME=xxx>, onde :

NAME dever receber o nome da varivel que receber o status atual de comunicaão do POS.

As strings retornadas so as seguintes destacadas em negrito e explicadas logo em seguida:

<b>CX..:</b>	Conectando na rede.
<b>Rdok:</b>	Rede OK, indica conexo fsica com sucesso.
<b>StNw:</b>	Inicia conexo lgica.
<b>Cxok:</b>	Conexo lgica com sucesso, conectado com a operadora.
<b>Wfok:</b>	Encontrou a rede Wi-Fi.
<b>Htok:</b>	Conectado com o Servidor.
<b>HOST:</b>	Servidor OK.
<b>Hoff:</b>	Servidor offline.
<b>RX..:</b>	Recebendo dados.
<b>Rxok:</b>	Dados do Servidor recebidos.
<b>TX..:</b>	Enviando dados ao Servidor.
<b>CXer:</b>	Erro de conexo.
<b>DNSe:</b>	Erro de DNS, tentando encontrar pgina.
<b>....:</b>	Reconectando na rede.

### 3.2.9 CONTROLA SESSÃO

Para controlar a sessão da comunicação, devemos passar para o parâmetro TYPE o argumento HIDDEN. Os parâmetros NAME e VALUE são sempre obrigatórios.

Assim, os parâmetros NAME=xxx VALUE=yyy são sempre retornados pelo POS.

<GET TYPE=HIDDEN NAME=xxx VALUE=yyy>

Obs.: o tamanho máximo para o parâmetro VALUE é 30 dígitos.

## Especificação de Interface

---

### 3.2.10 VALOR MOEDA

Para capturar um valor moeda numérica, não secreta, por meio do teclado devemos passar para o parâmetro TYPE o argumento VALUE.

Neste caso haverá os parâmetros mandatórios: SIZE, COL, LIN.

Será mostrado com R\$, 2 casas decimais e separação com pontos.

Ex.: R\$1.000.000,00.

SIZE especifica o tamanho em caracteres do campo a ser capturado.

COL especifica a coluna da esquerda para direita onde o primeiro caractere capturado será mostrado no display.

LIN especifica a linha de cima para baixo onde o primeiro caractere capturado será mostrado no display.

Devemos notar que na linha onde será mostrado o campo, todo texto previamente existente será apagado.

O comando deverá ser passado conforme o esquema abaixo:

```
<GET NAME=CAMPO1 TYPE=VALUE SIZE=X COL=Y LIN=Z>
```

## Especificação de Interface

---

### 3.2.11 NUMERO DE SÉRIE

Para capturar o número de série do terminal, devemos passar para o parâmetro TYPE o argumento SERIALNO.

O parâmetro NAME é sempre obrigatório.

Assim, o parâmetro NAME=xxx é sempre retornado pelo POS.

<GET TYPE=SERIALNO NAME=xxx>

## Especificação de Interface

---

### 3.2.12 CÓDIGO DE BARRAS

Para fazer leitura de dispositivo de código de barras através da entrada USB do terminal, devemos passar para o parâmetro TYPE o argumento CODEBAR.  
O parâmetro NAME é sempre obrigatório.

Assim, o parâmetro NAME=xxx é sempre retornado pelo POS.

<GET TYPE=CODEBAR NAME=xxx>

### 3.2.13 PORTA SERIAL

Apenas para os modelos que tenham esta entrada.

Para capturar informações da porta serial RS232, devemos passar para o parâmetro TYPE o argumento SERIAL.

O parâmetro NAME é sempre obrigatório.

SIZE especifica o tamanho dos dados que serão lidos.

Existe um timeout que pode ser alterado na configuração, que ocorrerá caso não seja lido nada ou lido um tamanho menor que o informado.

O parâmetro NAME=xxx é sempre retornado pelo POS.

<GET TYPE=SERIAL NAME=xxx SIZE=yyy>

## Especificação de Interface

---

### 3.2.14 CONTROLES DA PORTA SERIAL

Apenas para os modelos que tenham esta entrada.

Para capturar informações da porta serial RS232, devemos passar para o parâmetro TYPE o argumento SERIALINE.

O parâmetro NAME é sempre obrigatório.

CONTROL especifica a máscara dos bits que serão lidos.

O parâmetro NAME=xxx é sempre retornado pelo POS.

<GET TYPE=SERIALINE NAME=xxx CONTROL=X>

Onde X é o valor da máscara dos bits que será enviado, conforme abaixo:

8: DCD

32: CTS

Caso queira receber os pinos DCD e CTS, usar o valor 40 (8+32).

## Especificação de Interface

---

### 3.2.15 VERSÃO

Para capturar a identificação da versão do NAVS, devemos passar para o parâmetro TYPE o argumento VERSION.

O parâmetro NAME é sempre obrigatório.

Assim, o parâmetro NAME=xxx é sempre retornado pelo POS.

<GET TYPE=VERSION NAME=xxx>

### 3.2.16 VERSÃO DO SISTEMA OPERACIONAL

Para capturar a identificação da versão do sistema operacional do POS, devemos passar para o parâmetro TYPE o argumento OSVERSION.

O parâmetro NAME é sempre obrigatório.

Assim, o parâmetro NAME=xxx é sempre retornado pelo POS.

<GET TYPE=OSVERSION NAME=xxx>

## Especificação de Interface

---

### 3.2.17 MODELO

Para capturar o modelo do terminal POS, devemos passar para o parâmetro TYPE o argumento MODEL. O parâmetro NAME é sempre obrigatório.

Assim, o parâmetro NAME=xxx é sempre retornado pelo POS, que poderá ser “VX520” ou “VX680”

<GET TYPE=MODEL NAME=xxx>

### 3.2.18 IDENTIFICAÇÃO DO CHIP

Para capturar a identificação do chip GPRS, devemos passar para o parâmetro TYPE o argumento ICCID. O parâmetro NAME é sempre obrigatório.

Assim, o parâmetro NAME=xxx é sempre retornado pelo POS, com a identificação do chip.

<GET TYPE=ICCID NAME=xxx>

## Especificação de Interface

---

### 3.2.19 MIFARE

Apenas para os modelos de POS que tenham esta entrada.

Para capturar os dados de identificação de cartão Mifare (contactless), devemos passar para o parâmetro TYPE o argumento MIFARE.

O comando deve ser passado conforme abaixo, notando que a ordem dos parâmetros não altera o resultado.

```
<GET TYPE=IDMIFARE NAME=xxxxx>
```

### 3.2.20 LEITURA DE DADOS DO MIFARE

Apenas para os modelos de POS que tenham esta entrada.

Para efetuar a leitura dos dados de um determinado setor do cartão Mifare (contactless), devemos passar para o parâmetro TYPE o argumento MIFARE\_READ.

O comando deve ser passado conforme abaixo, notando que a ordem dos parâmetros não altera o resultado.

```
<GET TYPE=MIFARE_READ NAME=xxxxx KEY=yyyyyyyyyyyyy KTYPE=a BLOCK=01 SECTOR=0  
BCOUNT=2>
```

Onde:

NAME: variável de retorno do comando.

KEY: chave para leitura do setor.

KTYPE: tipo da chave, pode ser 'a' ou 'b'.

BLOCK: bloco a ser lido.

SECTOR: setor a ser lido.

BCOUNT: contador de blocos.

## Especificação de Interface

---

### 3.2.21 TOUCH (somente no Vx680)

Para capturar informações na console do modelo Vx680, através do 'touch' na tela, devemos passar para o parâmetro TYPE o argumento TOUCH.

O comando deve ser passado conforme o esquema abaixo, notando que a ordem dos parâmetros não altera o resultado.

```
<GET TYPE=TOUCH NAME=xxxxx>
```

O comando retorna na variável NAME, o nome da figura ativa que foi tocada.

## Especificação de Interface

---

Como requisito básico do comando TOUCH, figuras devem ser desenhadas na tela, conforme comandos a seguir:

**Obs.: Considerar a tela do terminal Vx680 com: 240 x 320 pixels.**

### Retângulo:

<RECTANGLE NAME=XXX X=x Y=y WIDTH=zzz HEIGHT=www VISIBLE=k>

Onde:

X e Y são as coordenadas origem do retângulo, considerado a partir do alto a esquerda.

X é a coordenada na horizontal e Y na vertical.

WIDTH é o tamanho da largura do retângulo em pixels.

HEIGHT é o tamanho da altura do retângulo em pixels.

VISIBLE indica se a figura será visível ou não. Valores: 0 (não visível) ou 1 (visível).

### Círculo:

<CIRCLE NAME=XXX X=x Y=y RADIUS=zzz VISIBLE=k>

Onde:

X e Y são as coordenadas do centro do círculo.

X é a coordenada na horizontal e Y na vertical.

RADIUS é o tamanho do raio do círculo em pixels.

VISIBLE indica se a figura será visível ou não. Valores: 0 (não visível) ou 1 (visível).

### Polígono:

<POLYGON NAME=XXX VISIBLE=k>

X,Y;

X,Y;

X,Y;

</POLYGON>

Onde:

X e Y são as coordenadas dos vértices do polígono.

X é a coordenada na horizontal e Y na vertical.

VISIBLE indica se a figura será visível ou não. Valores: 0 (não visível) ou 1 (visível).

Exemplo para X e Y de polígono triângulo:

<POLYGON NAME=POLIG VISIBLE=1>

20,20;

20,70;

50,50;

</POLYGON>

A ordem dos vértices é importante.

**OBS.: TODAS AS FIGURAS ACIMA SAO REMOVIDAS DA MEMORIA QUANDO A TECLA 'CANCELAR' É PRESSIONADA OU QUANDO COMANDO CONSOLE É DADO.**

## Especificação de Interface

---

### 3.2.22 DATA E HORA

Para capturar a data e hora do terminal, devemos passar para o parâmetro TYPE o argumento DATETIME. O parâmetro NAME é sempre obrigatório.

Assim, o parâmetro NAME=xxx é sempre retornado pelo POS.

<GET TYPE=DATETIME NAME=xxx>

A variável NAME receberá a data e hora no formato: AAAAMMDDHHMMSS.

## Especificação de Interface

---

### 3.2.23 CPF

Para capturar um CPF, por meio do teclado devemos passar para o parâmetro TYPE o argumento CPF. Neste caso haverá os parâmetros mandatórios: COL, LIN.

COL especifica a coluna da esquerda para direita onde o primeiro caractere capturado será mostrado no display.

LIN especifica a linha de cima para baixo onde o primeiro caractere capturado será mostrado no display.

Devemos notar que na linha onde será mostrado o campo, todo texto previamente existente será apagado.

O comando deverá ser passado conforme o esquema abaixo:

```
<GET TYPE=CPF NAME=xxx COL=Y LIN=Z>
```

O comando só retorna números válidos e com campos sem formatação. Ex.: CPF capturado = 305.123.456-78, valor de CPF retornado = 30512345678.

Obs.: O usuário não precisa inserir a formatação na tela, ela aparece conforme os campos vão sendo editados.

## Especificação de Interface

---

### 3.2.24 HOUR

Para capturar um horário, por meio do teclado devemos passar para o parâmetro TYPE o argumento HOUR.

Neste caso haverá os parâmetros mandatórios: COL, LIN.

COL especifica a coluna da esquerda para direita onde o primeiro caractere capturado será mostrado no display.

LIN especifica a linha de cima para baixo onde o primeiro caractere capturado será mostrado no display.

Devemos notar que na linha onde será mostrado o campo, todo texto previamente existente será apagado.

O comando deverá ser passado conforme o esquema abaixo:

```
<GET TYPE=HOUR NAME=xxx COL=Y LIN=Z>
```

O comando só retorna horários válidos e com campos sem formatação. Ex.: HOUR capturada = 22:33:13, valor de HOUR retornado = 223313.

Obs.: O usuário não precisa inserir a formatação na tela, ela aparece conforme os campos vão sendo editados.

## Especificação de Interface

---

### 3.2.25 DATA

Para capturar uma data, por meio do teclado devemos passar para o parâmetro TYPE o argumento DATA. Neste caso haverá os parâmetros mandatórios: COL, LIN, SIZE.

SIZE especifica o tamanho em caracteres do campo a ser capturado, deve ser 6 (Ex. de saída: 181013) ou 8 (Ex. de saída : 18102013)

COL especifica a coluna da esquerda para direita onde o primeiro caractere capturado será mostrado no display.

LIN especifica a linha de cima para baixo onde o primeiro caractere capturado será mostrado no display.

Devemos notar que na linha onde será mostrado o campo, todo texto previamente existente será apagado.

O comando deverá ser passado conforme o esquema abaixo:

```
<GET TYPE=DATA NAME=xxx SIZE=8 COL=Y LIN=Z>
```

O comando só retorna datas válidas e com campos sem formatação. Ex.: DATA capturada = 18/10/2013, valor de DATA retornado = 18102013.

Obs.: O usuário não precisa inserir a formatação na tela, ela aparece conforme os campos vão sendo editados.

## Especificação de Interface

---

### 3.2.26 CNPJ

Para capturar um CNPJ, por meio do teclado devemos passar para o parâmetro TYPE o argumento CNPJ. Neste caso haverá os parâmetros mandatórios: COL, LIN.

COL especifica a coluna da esquerda para direita onde o primeiro caractere capturado será mostrado no display.

LIN especifica a linha de cima para baixo onde o primeiro caractere capturado será mostrado no display.

Devemos notar que na linha onde será mostrado o campo, todo texto previamente existente será apagado.

O comando deverá ser passado conforme o esquema abaixo:

```
<GET TYPE=CNPJ NAME=xxx COL=Y LIN=Z>
```

O comando só retorna números válidos e com campos sem formatação. Ex.: CNPJ capturado = 11.444.777/0001-61, valor de CNPJ retornado = 11444777000161.

Obs.: O usuário não precisa inserir a formatação na tela, ela aparece conforme os campos vão sendo editados.

## Especificação de Interface

---

### 3.2.27 APDUCHIP

<GET TYPE=APDUCHIP NAME=RESP APDU=xxxxxx SLOT=1 ON=N OFF=N RC\_NAME =rrr ATR\_NAME =aaa>

Onde rrr é o nome da variável que receberá o return-code  
E aaa é o nome da variável que receberá o ATR no caso de ON=Y

NAME é o campo que recebe o retorno da chamada da apdu  
SLOT =0, 1, 2 ou 3, sendo 0 o cartão externo e os demais os SAMs  
ON=Y se quer que ligue o cartão, ON=N caso contrário  
OFF=Y se quer que desligue o cartão, OFF=N caso contrário  
APDU é a apdu em hexa

O comando ainda retorna RC\_NAME que deve ser 00 se tudo for ok  
ATR\_NAME com o valor do ATR em hexa  
APDU com o valor da apdu enviada

OBS: RC\_NAME =00 não significa que a apdu funcionou, e sim que o comando funcionou. A resposta da APDU deve ser verificada  
Caso RC\_NAME não seja 00, o campo indicado por NAME tem o erro.

O limite de bytes hexa a ser enviado para APDUs é de 150 bytes o que dá uma APDU binária de 75 bytes. Caso seja necessário uma APDU maior, usar os parâmetros APDU1, APDU2, APDU3, APDU4 e APDU5 respeitando essa ordem. Os conteúdos serão concatenados ao parâmetro APDU. Lembramos que, para cada parâmetro APDUx, o limite também é de 150 bytes hexa. Só usar os parâmetros necessários para conseguir enviar a totalidade de bytes desejada.

## Especificação de Interface

---

### 3.2.28 DESNBS

<GET TYPE=DESNBS CODOP=x KEY=kkkk IN=yyyy NAME=nnnn RC\_NAME=rrr >

Criptografia no padrao DES

X = 0 cifra

X = 1 decifra

KEY = chave de criptografia com 8 bytes em hexa (16 caracteres)

IN = campo a (des)criptografar com 8 bytes em hexa (16 caracteres)

NAME = nome do campo (des)criptografado com 8 bytes em hexa (16 caracteres)

O comando ainda retorna RC\_NAME que deve ser 00 se tudo for ok

Caso RC\_NAME não seja 00, o campo indicado por NAME tem o erro.

### 3.3 *DELAY*

Usar este comando quando quiser que o POS aguarde algum tempo antes de continuar qualquer outro comando.

Ideal para mensagens para a Console temporizadas.

```
<DELAY TIME=xx>
```

Onde xx é o tempo de espera em segundos.

Exemplo para uso com mensagem para console temporizada em 3 segundos:

```
<CONSOLE> Texto para display </CONSOLE>  
<DELAY TIME=3>
```

### 3.4 **SELECT**

Para selecionar uma opção a partir de um menu de itens de caracteres, utilizar o comando abaixo:

```
<SELECT LIN=3 COL=4 SIZE=15 QTD=3 UP=E1 DOWN=E2 RIGHT=B7 LEFT=B9 NAME=SEL
TYPE_RETURN=3 INDEX=n1,n2,n3,...,nX>
item1,
item2,
item4,
etc,
fim
</SELECT>
```

Onde,

LIN = linha inicial da lista

COL=coluna inicial da lista

SIZE=Tamanho de cada item da lista (o que será mostrado em reverso)

QTD=Quantidade de itens da lista a aparecerem da tela

UP=tecla de paginação para cima

DOWN= tecla de paginação para baixo

NAME=nome da variável que retornara o valor selecionado pelo ENTER da lista

“item1,item2,item4,etc,fim”= exemplo de lista de valores entre virgulas.

RIGHT= tecla para fazer a rolagem da string na horizontal, para o lado direito, caso ela seja maior que área de apresentação.

LEFT= tecla para fazer a rolagem da string na horizontal, para o lado esquerdo, caso ela seja maior que área de apresentação.

TYPE\_RETURN é o parâmetro que define se o retorno será a string selecionada ou o número associado á ela na ordem em que as strings são listadas. Assume os valores 1 (retorna string), ou 2(retorna o número correspondente), ou 3(retorna o índice aleatório correspondente).

INDEX é o parâmetro que recebe os valores do índices aleatórios desejados. Note que n1 a nX devem ser números INTEIROS que são separados por vírgula.

Para inserir espaços usar %20:

Exemplo: “sao paulo”, tem que ficar: “sao%20paulo”.

Para inserir “>” e “<” usar %3E e %3C respectivamente:

Exemplo1: “1-numerox>500”, tem que ficar: “1-numerox%3E500”.

Exemplo2: “2-numeroy<250”, tem que ficar: “2-numeroy%3C250”.

Para inserir “,” usar %2C:

Exemplo: “0,5”, tem que ficar: “0%2C5”.

Obs.: Este comando está relacionado a “<CONSOLE>xx</CONSOLE>”, porém deverá ser acionado fora dele.

Abaixo alguns exemplos de caracteres hexadecimal das teclas para o SELECT:

<b>TECLA</b>	<b>VALOR HEXA</b>
F1	FA
F2	FB
F3	FC
F4	FD
Roxa 1	E1
Roxa 2	E2
Roxa 3	E3
Roxa 4	E4
*	AA
#	A3
0	B0
1	B1
2	B2
3	B3
5	B5
6	B6
7	B7
8	B8
9	B9

## Especificação de Interface

---

### 3.5 *TONE*

Usar este comando quando quiser fazer com que o POS emita sons.

Exemplo de uso:

<TONE TYPE=xxx>, onde “xxx” pode ser:

NORMAL: emite um beep indicando normalidade.

ERROR: emite um beep indicando anormalidade.

### 3.6 *PRINTER*

<PRINTER PRNSTS=STS ENABLE\_CANCEL=xxx> inicia texto para a impressora.

PRNSTS informa a variável que terá o retorno do status final da operação de impressão,

</PRINTER> finaliza texto para a impressora,

A mudança de linha é realizada pelo comando <BR>.

ENABLE\_CANCEL, especifica se tecla CANCEL será ativa ou não após o acionamento da impressão, até o envio do comando <POST>.

Onde xxx, recebe os seguintes valores:

0 = permite cancelamento;

1 = não cancela nada até os dados serem enviados ao servidor (comando POST).

Obs.: O texto máximo para impressora é de 3.000 caracteres.

PRNSTS, retorna na variável (Ex.: STS, o status da impressão)

Os valores de status retornados são:

0: Ok;

-4: Falha;

-5: Pouco papel;

-10: erro de RAM;

-20: Falha na impressora;

-21: Sem papel;

-23: Sequência de Escape Code não encontrada;

-24: Impressora não inicializada;

-27: Firmware corrompido.

Obs2.: Qualquer outro comando deverá ser utilizado fora de “<PRINTER PRNSTS=STS >xx</ PRINTER >”, entre eles haverá SOMENTE texto para o display.

### 3.7 CHGCONFNT

Comando que permite a mudana de fontes para o console.

<CHGCONFNT IDFONT=MODEL1 SIZE=10 INVERSE UNDERLINE>, onde :

IDFONT, indica a fonte a ser utilizada, assumindo os seguintes valores:

DEFAULT, onde o modelo padro do POS  selecionado, seu tamanho ser sempre 8.

MODEL1, corresponde ao fonte Arial, pode assumir os tamanhos 8 e 12.

MODEL2, corresponde ao fonte ASC, pode assumir o tamanho 16, suportado apenas no modelo Vx520.

MODEL3, corresponde ao fonte Courier New, pode assumir os tamanhos 12 e 16.

MODEL4, corresponde ao fonte Helvtica, pode assumir o tamanho 16, suportado apenas no modelo Vx520.

MODEL5, corresponde ao fonte K2, pode assumir o tamanho 16.

MODEL6, corresponde ao fonte Times New Roman, pode assumir os tamanhos 12 e 16.

MODEL7, corresponde ao fonte Verdana, pode assumir os tamanhos 12 e 16.

SIZE, indica o tamanho do fonte selecionado.

INVERSE e UNDERLINE so parmetros opcionais, que podem ser aplicados em todos os fontes exceto o DEFAULT.

## Especificação de Interface

---

### 3.8 PRNFNT

Para imprimir texto com fontes diferentes usar o comando abaixo:

<PRNFNT SIZE=x PARAMETROS>

Onde,

X = tamanho da fonte, no range de 2 a 4 (4 fonte menor, e 2 fonte maior).

2: 5x8 24 colunas.

3: 8x14 32 colunas.

4: 8x14 42 colunas.

*OBS.: o size 4 é o padrão inicial do NAVS.*

PARAMETROS:

DBL\_HEIGHT: dobro da altura

DBL\_WIDTH: dobro da largura

INVERSE: inverte as cores

Os parâmetros DBL\_HEIGHT, DBL\_WIDTH, e INVERSE, podem ser utilizados juntos no mesmo comando, com espaço entre eles.

Obs2.: Este comando está relacionado a “<PRINTER PRNSTS=STS >xx</ PRINTER >”, porém deverá ser acionado fora dele.

### 3.9 *WRITE\_AT*

<WRITE\_AT LINE=1 COLUMN=1> inicia texto para o display,  
</WRITE\_AT> finaliza texto para o display.

Não realiza mudança de linha.

Sobrescreve qualquer coisa que estiver na tela, sem apagar o conteúdo total dela, tendo início na linha e coluna definidos respectivamente pelos parâmetros LINE e COLUMN.

Obs.: Qualquer outro comando deverá ser utilizado fora de  
“<WRITE\_AT LINE=x COLUMN=y> ... </WRITE\_AT >”, entre eles haverá SOMENTE texto para o display.

## Especificação de Interface

---

### 3.10 DRAW (Somente para Vx680)

Para capturar um desenho feito na tela do VX680 e a partir dele gerar um arquivo com extensão “.bmp” monocromático, utilize o comando abaixo:

```
<DRAW BMP=desenho.bmp TRACE=2>
```

Onde:

BMP é o parâmetro que recebe o nome do arquivo bmp a ser gerado, que deverá vir sempre no seguinte formato “nome\_desejado”+“.bmp”.

TRACE é o parâmetro que define a intensidade do traçado na tela, com valores a partir de 1.

Para finalizar a captura de traçado na tela deve-se utilizar o comando abaixo:

```
</DRAW>
```

Dentro do seguinte conjunto:

```
<DRAW BMP=desenho.bmp TRACE=2>  
</DRAW>
```

Poderá utilizar outros comandos como por exemplo :

- <GET TYPE=ANYKEY> , o que fará com que o usuário desenhe, e após concluir possa pressionar a tecla VERDE e somente depois disso gerar o arquivo bmp.
- <CAPTURE NAME=xxx>  
    <GET TYPE=FIELD ALPHA=1 SIZE=X COL=Y LIN=Z NOENTER=1>  
    <GET TYPE=SPECIALKEY>  
    </CAPTURE>

### 3.11 CHGPRNFNT

Para imprimir texto com fontes diferentes usar o comando abaixo:

```
<CHGPRNFNT SIZE=X FACE=FACE1 PARAMETRO1 PARAMETRO2>
```

Onde,

X = tamanho da fonte, no range de 2 a 4 (4 fonte menor, e 2 fonte maior).

2: 5x8 24 colunas.

3: 8x14 32 colunas.

4: 8x14 42 colunas.

OBS.: o tamanho 4 é o padrão inicial do NAVS.

FACE = identificação da fonte:

“FONTE1” é modelo default do POS.

“FONTE 2”.

“FONTE 3”.

“FONTE 4”.

“FONTE 5”.

“FONTE 6”.

Obs.: “FONTE 1” e “FONTE 2” não suportam os parâmetros “UNDERLINE” e “BOLD”.

Obs2.: “FONTE 3” e “FONTE 4” não suportam o tamanho 2, somente 3 e 4.

Obs3.: “FONTE 5” e “FONTE 6” com os parâmetros “INVERSE”, “BOLD”, “UNDERLINE” juntos não suportam o tamanho 4, somente 2 e 3.

PARAMETROS:

DBL\_HEIGHT: dobro da altura

DBL\_WIDTH: dobro da largura

BOLD: formato negrito.

INVERSE: inverte os bits da fonte.

UNDERLINE: sublinha os caracteres.

Os parâmetros: “DBL\_HEIGHT”, “DBL\_WIDTH”, “BOLD”, “INVERSE”, e “UNDERLINE”, podem ser utilizados juntos no mesmo comando, com espaço entre eles.

Obs4.: Este comando está relacionado a “<PRINTER PRNSTS=STS >xx</ PRINTER >”, porém deverá ser acionado fora dele.

Obs5.: Todos os parâmetros devem ser passados com letras maiúsculas.

A seguir , os exemplos dos modelos de todos os fontes disponíveis:



### **3.12 PRNLOGO**

Para imprimir arquivo Imagem usar o comando abaixo:

```
<PRNLOGO NAME=xxxx.bmp SPACES=Y>
```

Onde,

xxxx.bmp (name) é o nome do arquivo a ser impresso (sempre em minúsculo, e com no máximo 29 caracteres). O arquivo já deve estar carregado no POS.

Y (spaces) são os espaços à esquerda para centralização.

A imagem terá que ser no formato bmp e monocromática.

O tamanho máximo do comprimento da imagem é de 384 pixels.

### 3.13 CONLOGO

Para mostrar no display do POS um arquivo Imagem usar o comando abaixo:

```
<CONLOGO NOCLS=pppp NAME=xxxx.bmp X=x Y=y>
```

Onde,

xxxx.bmp (name) é o nome do arquivo que será colocado na tela do POS (sempre em minúsculo). O arquivo já deve estar carregado no POS.

X e Y são as coordenadas do display do POS.

“pppp” é o valor de NOCLS.

O parâmetro NOCLS determina se uma imagem que estiver no display será apagada ou não com a atualização do texto na tela.

Os valores que podem ser assumidos são 0=não 1=sim.

Para o modelo Vx520 o arquivo tem que ser do tipo bmp e monocromático. Tamanho máximo (128x64).

Já para o Vx680 tem que ser também no formato bmp, porém pode ser colorido. Se a imagem tiver exatamente 240x320 (dimensões máximas do POS), terá que ser salva com intensidade de 8 bits, se for menor que a resolução máxima (240x320) a intensidade de bits poderá ser maior.

### **3.14 CODEBAR – Impressão**

<CODEBAR VALUE=XXXXXXXXXXXX SIZE=Y HEIGHT=ZZ SPACES=W>

Onde XXXXXXXX (value), é o texto a ser convertido para o código de barras.

Y (size) é o tamanho, de 1 a 4, como recomendação usar 2.

ZZ (height) é a altura do código de barras em pixels, recomendação usar 128.

W (spaces) são os espaços à esquerda para centralização, recomendação usar 90.

O padrão utilizado é o EAN-13, onde o valor máximo para o VALUE deve ser 12. O NAVS calcula o dígito.

No EAN-13 o símbolo codifica treze números que estão divididos em quatro partes; dos treze dígitos, doze são dos dados referentes ao produto e um é o dígito verificador.

### 3.15 GERADOR DE QR CODES

Para criar um QrCode a partir de uma string usar o comando abaixo:

```
<GENERATE_QR_CODE SIZE=2 QR_ECLEVEL=3 KEEP_FILE=1 SPACES=0  
FILE_NAME=testeQRA.bmp ERR_QR=VAR_R>  
    TEXTO A SER CONVERTIDO EM QR CODE  
</GENERATE_QR_CODE>
```

Onde:

SIZE é o parâmetro que controla o tamanho do QR Code.

QR\_ECLEVEL é o parâmetro que controla a resolução do QR Code criado, o que também pode interferir no tamanho final dele. Deverá receber os seguintes valores 0 (menor resolução possível), 1, 2 ou 3 (maior resolução possível).

KEEP\_FILE é o parâmetro que indica se o arquivo QR Code criado deverá ser ou não mantido para uso posterior (Ex.: Mostrar na tela utilizando o comando CONLOGO), deverá receber os valores 0 (não manter) ou 1 (manter).

SPACES é o parâmetro que define o número de espaços á esquerda para a impressão do QR Code.

FILE\_NAME é o parâmetro que deverá conter o nome do arquivo, caso queira mantê-lo, a extensão dele deverá ser sempre bmp, pois o POS não trabalha com outros formatos. Todos os nomes deverão ter o seguinte formato : "nome.bmp".

ERR\_QR é o parâmetro que retorna se o QR Code foi o não gerado com sucesso. Os valores retornados são 1 (representa sucesso) ou 0 (representa falha). As falhas de criação estão associadas ao tamanho pequeno do QR Code final, para corrigir isso é recomendado aumentar os valores de SIZE e QR\_ECLEVEL.

</GENERATE\_QR\_CODE> indica o encerramento do texto para ser convertido em QR Code.

Exemplo de QR Code gerado:



### 3.16 Criar Código de barras 3 de 9

Para gerar e imprimir um código de barras 3 de 9, utilizar o comando abaixo:

```
<CODE_BAR_3_9 WIDTH=130 HEIGHT =6 KEEP_FILE=0 FILE_NAME=teste3x9.bmp SPACES=0  
ERROR=RETORNO3x9>
```

**TEXTO A SER CONVERTIDO EM CODE BAR 3 DE 9**

```
</CODE_BAR_3_9>
```

Onde:

WIDTH é o parâmetro que controla o tamanho da largura do código de barras.

HEIGHT é o parâmetro que controla o tamanho da altura do código de barras.

SPACES é o parâmetro que define o número de espaços á esquerda para a impressão do Código de barras.

ERROR é o parâmetro que retorna se o Código de barras foi o não gerado com sucesso. Os valores retornados são 1 (representa sucesso) ou 0 (representa falha). As falhas de criação estão associadas ao tamanho do código de barras final (grande ou pequeno demais), para corrigir isso é recomendado tomar cuidado com o valores de WIDTH e HEIGHT. Obs.: O valor de WIDTH deverá ser superior ao valor de HEIGHT.

</CODE\_BAR\_3\_9> indica o encerramento do texto para ser convertido em Código de barras.

### **3.17 Criar Código de Barras ITF (2 de 5)**

Para gerar e imprimir um código de barras ITF, utilizar o comando abaixo:

```
<CODE_BAR_2_5 VALUE=0123456789 WIDTH =2 HEIGHT=120 ERROR=MY_RET>
```

Onde:

VALUE é o parâmetro que recebe o valor a ser codificado para o código de barras.

WIDTH é o parâmetro que controla o tamanho da largura do código de barras.

HEIGHT é o parâmetro que recebe o tamanho da altura do código de barras.

ERROR é o parâmetro que retorna se o Código de barras foi o não gerado com sucesso. Os valores retornados são 1 (representa sucesso) ou 0 (representa falha). As falhas de criação estão associadas ao tamanho pequeno do Código de barras final, para corrigir isso é recomendado aumentar os valores de WIDTH e HEIGHT.

### 3.18 TELECARGA

#### 1. VIA COMANDO <FILE>

Para efetuar a Telecarga remota de uma versão mais atualizada via comando FILE, coloque a nova versão no Servidor com o nome NAVS.ZIP, e, use a sequência de comandos abaixo para iniciar a transferência do arquivo e efetivar a atualização no POS:

```
<FILE NAME=navs.zip ADDR=/navs.zip ERR=ERROARQ>  
<RESET>
```

O comando FILE irá processar a transferência do arquivo e o comando RESET irá reiniciar o POS para que a versão possa ser atualizada no POS.

#### 2. VIA SERVIDOR VERICENTRE

Para efetuar a Telecarga remota de uma versão mais atualizada usar o comando abaixo:

```
<TELECARGA ZA=NAVS ZT=555 ADDRESS=200.160.80.90 PORT=8013 TPCARGA=W>
```

Onde,

NAVS (za) é um exemplo de nome da aplicação no Servidor de Telecargas Vericentre.

555 (zt) é um exemplo da identificação do terminal no Servidor Vericentre.

200.160.80.90 (address) é o IP do Servidor Vericentre.

8013 (port) é a porta do Servidor Vericentre.

W (tpcarga) é o tipo de telecarga, que pode ser:

F – carga total com Reset do POS (apaga a carga atual)

P – carga parcial com Reset do POS.

p – carga parcial sem Reset do POS.

### **3.19 SAÍDA SERIAL**

Apenas para os modelos que tenham esta saída.

<SERIAL> inicia texto para a serial,  
</SERIAL > finaliza texto para a serial,

Obs.: O tamanho máximo para a serial é de 3.000 caracteres.

Expo: caso queira enviar: "TESTE", o comando será:  
<SERIAL>5445535445</SERIAL>, pois 'T' é '54', 'E' é '45' em ascii e assim por diante.

### **3.20 SET TIME**

Comando para alterar a data e/ou hora do POS:

```
<SET TYPE=TIME HOUR=223400 DATE=16102002 HDSTS=STSSTT>
```

Onde:

HOUR: É a hora que atribuiremos ao POS, se for “0” será mantida a hora atual. Deverá estar no formato hhmmss (hh = hora, mm= minuto, ss = segundos);

DATE: É a data que atribuiremos ao POS, se for “0” será mantida a data atual. Deverá estar no formato ddmmaaaa (dd = dia, mm= mês, aaaa = ano (exemplo:2013));

HDSTS: é o retorno do status da operação, que pode ser:

- OK;
- ERRO\_LEITURA, é erro de leitura de dados do POS quando um dos parâmetros (HOUR ou DATE) vai zerado;
- ERRO\_ENTRADA, é erro no formato enviado para o POS em HOUR ou DATE;
- e ERRO\_GRAVAÇÃO, quando o POS não conseguiu efetuar a gravação dos novos parâmetros, portanto a operação não foi executada;

OBS.: As datas devem ser a partir do ano 2000, caso contrário é provável que ocorra “ERRO\_GRAVAÇÃO”.

### 3.21 **KEYBOARD**

Comando para captura de strings no formato alfanumérico através de um teclado touch (letras, a-z) e teclado do POS (números), funciona somente para Vx680.

```
<KEYBOARD MODEL=A0 BEEP=ON SIZE=15 HIDE=OFF RETURN=TEC>
```

Onde :

MODEL = é o tipo de teclado a ser utilizado, no momento temos apenas o A0, mas posteriormente existirão outros modelos.

BEEP = define se o touch emitirá ou não som ao ser pressionado, assume valores ON (com beep) e OFF (sem beep).

SIZE = define o tamanho da string a ser capturada.

HIDE = define se a string será ou não mostrada na tela.

RETURN = recebe a variável para retorno.

### **3.22 SET PORTA SERIAL**

Apenas para os modelos que tenham esta saída.

Para enviar informações para a porta serial RS232, devemos passar para o parâmetro TYPE o argumento SERIALINE.

CONTROL especifica a informação que será enviada.

<SET TYPE=SERIALINE CONTROL=X>

Onde X é o valor que será enviado para a porta serial, conforme abaixo:

- 0: -BRK -RTS -DTR
- 1: -BRK -RTS +DTR
- 2: -BRK +RTS -DTR
- 3: -BRK +RTS +DTR
- 4: +BRK -RTS -DTR
- 5: +BRK -RTS +DTR
- 6: +BRK +RTS -DTR
- 7: +BRK +RTS +DTR

### **3.23 CAPTURA NICA DE ENTRADA**

Para capturar vrias entradas de leitura no mesmo comando usar o comando CAPTURE, informando quais entradas deseja capturar.

Informe o parmetro NAME, como retorno do comando:

```
<CAPTURE NAME=TESTE>  
</CAPTURE>
```

Para as entradas de captura, no use o parmetro NAME, pois ele  usado na entrada do comando CAPTURE

Exemplo de uso para capturar: entrada digitada, carto tarja, leitura de cdigo de barras via USB, e Timeout do comando:

```
<CAPTURE NAME=TESTE>  
  <GET TYPE=FIELD LIN=5 COL=1 SIZE=10>  
  <GET TYPE=CARD TRACKS=2>  
  <GET TYPE=CODEBAR>  
  <GET TYPE=TIMEOUT TIME=10 ACTION=CANCEL>  
</CAPTURE>
```

Onde TIMEOUT pode ser:

<GET TYPE=TIMEOUT TIME=10 ACTION=CANCEL>, indica que se ocorrer um Timeout de 10 segundos, haver um retorno CANCEL para o incio da transaao.

Ou

<GET TYPE=TIMEOUT TIME=10 ACTION=SEND VALUE=xxx>, indica que se ocorrer um Timeout de 10 segundos, haver um retorno da varivel TESTE=xxx.

### 3.24 *INIT*

<INIT KEEP\_COOKIES=xxx>

Define o ponto da transação que o POS retornará quando for acionada a tecla CANCELAR (X).  
Onde xxx, pode assumir os seguintes valores : 0 (**não** manter cookies), 1 (manter cookies).

### **3.25 POST**

<POST> comando para que o POS envie as informações ao Servidor.

### 3.26 FILE

Para receber um arquivo do servidor, usar o comando abaixo:

```
<FILE NAME=xxxxx ADDR=yyyyy ERR=zzzzz WRT=wwwwww>
```

Onde,

xxxxx é o nome do arquivo que será gravado na RAM do POS.

yyyyy é o nome do diretório e nome do arquivo origem no Servidor.

zzzzz é o nome da variável que receberá o resultado da operação.

wwwwww é um parâmetro enviado que decide que se já houver o arquivo sobrescreve ou não o mesmo, enviar: "SIM" se deseja sobrescrever arquivo e "NAO" se não deseja.

A variável ERR é retornada ao Servidor com o status do erro, se 0 foi sucesso, se 1 deu erro e se 2 não sobrescreveu porque já havia arquivo.

Exemplo:

```
<FILE NAME=logoprint.bmp ADDR=/logoprint.bmp ERR=ERROARQ WRT=SIM>
```

Com este comando também é possível fazer a atualização remota da versão do NAVS. Vide capítulo 3.9.1 TELECARGA VIA COMANDO FILE.

### **3.27 REMOVE ARQUIVO NO POS**

Para apagar um arquivo no POS, usar o comando abaixo:

```
<DELFILE NAME=xxxxx ERR=zzzzz>
```

Onde,

xxxxx é o nome do arquivo que será removido no POS.

zzzzz é o nome da variável que receberá o resultado da operação.

A variável ERR é retornada ao Servidor com o status da operação, se 0 foi sucesso, se 1 foi erro.

Exemplo:

```
<DELFILE NAME=logoprint.bmp ERR=ERROARQ>
```

### **3.28 *RESET***

<RESET> comando para reiniciar o POS.

### 3.29 CONFIG\_NAVS

Comando que altera as configurações NAVS remotamente.

Em caso de erro a variável que for indicada no parâmetro RETURN receberá o código de erro.

Abaixo um exemplo de uso do comando:

```
<CONFIG_NAVS RETURN=STS_ALTERA_SERVER>;  
PASSWORD_CONFIG=senha  
NEW_PASSWORD_CONFIG=novasenha  
CONNECTION_TYPE=E;  
WI_FI_SSID=skytefwifi;  
WI_FI_PASSWORD=skytef;  
LOCAL_IP=192.168.1.58;  
LOCAL_MASK=255.255.0.0;  
LOCAL_GATEWAY=192.168.0.1;  
LOCAL_DNS_1=192.168.0.103;  
LOCAL_DNS_2=192.168.0.103;  
LOCAL_PING=15;  
GPRS_CONFIG=1;  
GPRS_APN=ZAP.VIVO.COM.BR;  
GPRS_USER=VIVO;  
GPRS_PASSWORD=VIVO;  
SERVER_IP=200.160.80.90;  
SERVER_PORT=6789;  
SERVER_RESOURCE=/TESTE.PHP;  
SERVER_HOST=200.160.80.90;  
SERVER_HTTPS_ACTIVE=0;  
SERVER_HTTPS_METHOD=1;  
POSITION_STATUS_LINE=B;  
SHOW_HOUR_AT_STATUS_LINE=1;  
SCROLL_UP=62;  
SCROLL_DOWN=63;  
PRINTER_CONTRAST=1;  
KEEP_ALIVE_ATIVAR=S;  
KEEP_ALIVE_TEMPO_DE_INTERVALO=20;  
KEEP_ALIVE_IP_DESTINO=200.160.80.90;  
KEEP_ALIVE_PORT=6789;  
</CONFIG_NAVS>
```

Obs.: Não é mais obrigatória a inserção de todos os parâmetros, basta inserir o parâmetro que se deseja alterar e inserir o parâmetro, seguindo o padrão acima demonstrado.

Obs.2: Para reiniciar o POS, após indicar os novos valores é necessário utilizar o comando <RESET>.

Obs.3: Somente quando "SERVER\_IP" e "SERVER\_PORT" o POS é reiniciado sem a necessidade de uso do comando <RESET>.

## Especificação de Interface

---

Obs. 4: Para alterar os parâmetros é necessário que a senha (PASSWORD\_CONFIG) esteja correta.

### Valores que cada parâmetro pode receber:

PASSWORD\_CONFIG, senha atual do dispositivo que se deseja alterar os parâmetros, deverá ser informado sempre.

NEW\_PASSWORD\_CONFIG, nova senha, se deseja alterar a atual.

CONNECTION\_TYPE, define que tipo de conexão será utilizada pelo POS E(ETHERNET), G(GPRS), W(WI-FI). É importante lembrar que a configuração só será aplicada se o POS suportar o tipo de conexão aferido.

WI\_FI\_SSID, recebe o SSID da rede em que se deseja inserir o POS;

WI\_FI\_PASSWORD, recebe a senha da rede Wi-Fi;

LOCAL\_IP, recebe o IP local, se utilizar DHCP preencha com 0.0.0.0.

LOCAL\_MASK, recebe a mascara da rede.

LOCAL\_GATEWAY, recebe o GATEWAY da rede.

LOCAL\_DNS\_1, recebe o valor de DNS, primeira opção.

LOCAL\_DNS\_2, recebe o valor de DNS, segunda opção.

LOCAL\_PING, recebe o valor de ping.

GPRS\_CONFIG, define qual a configuração de chip (APN, USUÁRIO, SENHA), serão utilizados. Se receber 1, esses valores terão de ser enviados ao servidor, pois esse caso representa uma exceção aos padrões definidos. Se igual a 2, o padrão adotado será da operadora VIVO, se igual a 3, o padrão adotado será da operadora CLARO, se igual a 4, o padrão adotado será da operadora OI, e se igual a 5, o padrão adotado será da operadora TIM.

GPRS\_APN, recebe a APN da operadora, só deverá ser preenchido se GPRS\_CONFIG receber valor igual a 1.

GPRS\_USER, recebe o nome de usuário da operadora, só deverá ser preenchido se GPRS\_CONFIG receber valor igual a 1.

GPRS\_PASSWORD, recebe a senha de usuário da operadora, só deverá ser preenchido se GPRS\_CONFIG receber valor igual a 1.

SERVER\_IP, recebe o IP do servidor onde estará o recurso NAVS a ser acessado.

SERVER\_PORT, recebe o id da porta do servidor onde estará o recurso NAVS a ser acessado.

## Especificação de Interface

---

SERVER\_RESOURCE, recebe o nome do recurso a ser acessado, lembrando que "/" deverá estar antes do nome (Ex.: "/TESTE.PHP").

SERVER\_HOST, recebe o HOST do servidor a ser acessado.

SERVER\_HTTPS\_ACTIVE, define se a conexão será https enviando valor igual a 1, ou http enviando valor igual a 0.

SERVER\_HTTPS\_METHOD, recebe definição do tipo de conexão https. 1=SSLv3, 2=TLSv1. Poderá receber um valor nulo somente se SERVER\_HTTPS\_ACTIVE = 0.

POSITION\_STATUS\_LINE, define a posição da linha de status, e recebe os valores "T"(topo), "B"(rodapé), "N"(ausente).

SHOW\_HOUR\_AT\_STATUS\_LINE, define se a hora irá ou não aparecer na linha de status. (0=ocultar, 1=mostrar).

SCROLL\_UP, receberá os valores que são definidos para essa tecla. (Verificar referencia ao parâmetro **SCROLL** nesta mesma documentação). Não poderá ter o mesmo valor que SCROLL\_DOWN.

SCROLL\_DOWN, receberá os valores que são definidos para essa tecla. (Verificar referencia ao parâmetro **SCROLL** nesta mesma documentação). Não poderá ter o mesmo valor que SCROLL\_UP.

PRINTER\_CONTRAST, define o nível de contraste para a impressão, assume valores de 1 a 8.

KEEP\_ALIVE\_ATIVAR, define se o keep alive estará ou não ativo, assume os valores "S"(sim) e "N"(não).

KEEP\_ALIVE\_TEMPO\_DE\_INTERVALO, define a frequência de envio em segundos do número de série ao servidor e porta programados.

KEEP\_ALIVE\_IP\_DESTINO, recebe o IP de destino onde o número de série do POS será enviado.

KEEP\_ALIVE\_PORT, recebe a porta do IP de destino onde o número de série do POS será enviado.

RETURN deve receber a variável que terá um retorno do status da operação.

A partir da versão 1.8.8 os valores de retorno foram mudados.

A resposta será "OK" se tudo estiver correto, ou uma string com o(s) nome(s) do(s) parâmetro(s) que apresenta(m) erro.

## Especificação de Interface

---

### 3.30 IMPRESSÃO DE BOLETO

<BOLETO BOLRET=xxxxx> inicia texto para o boleto,  
</BOLETO> finaliza texto para o boleto.

Exemplo de uso com alguns campos apenas:

<BOLETO BOLRET=STSBOL> inicia texto para o boleto,  
NOME\_BANCO=Bradesco;  
BANCO\_NUM=001;  
</BOLETO > finaliza texto para o boleto.

IMPORTANTE: os campos devem sempre finalizar com ';'.

A variável BOLRET é retornada ao Servidor com o status da impressão do Boleto.  
As seguintes strings podem retornar:

OK  
CODIGO INVALIDO  
ERRO CODIGO  
SEM PAPEL  
ERRO IMPRESSAO  
ERRO DESCONHECIDO

Abaixo todos os campos do boleto, com conteúdo de exemplo:

NOME\_BANCO=Bradesco;  
BANCO\_NUM=001;  
BANCO\_DIG=9;  
NOSSO\_NUM=0000001254588;  
VALOR\_EM\_CENTAVOS=12345678;  
VALOR\_DOCUMENTO\_EM\_CENTAVOS=8999999123;  
DESC\_EM\_CENTAVOS=1;  
DED\_EM\_CENTAVOS=22;  
MULTA\_EM\_CENTAVOS=333;  
ACRESC\_EM\_CENTAVOS=4444;  
VALOR\_COBRADO\_EM\_CENTAVOS=11122233344;  
VENCIMENTO=16/07/2012;  
MOEDA=9;  
LOCAL=qualquer banco até o vencimento, após o vencimento apenas em agências XXX;  
CEDENTE=Software Express (teste);  
DATA\_DOC=14/07/2012;  
NUMERO\_DOC=678;  
ESPECIE\_DOC=DM CI;  
ACEITE=N;  
DATA\_PROC=15/07/2012;  
USO\_BANCO=0.000.000/27272;  
CARTEIRA=018;

## Especificação de Interface

---

QUANTIDADE=1;  
INSTR1=Não receber após 30 dias da data de vencimento.;  
INSTR2=Instruções linha 2;  
INSTR3=Instruções linha 3;  
INSTR4=Instruções linha 4;  
INSTR5=Instruções linha 5;  
SACADO1=Teste;  
SACADO2=Av. Paulista, 2202 Sobreloja H;  
SACADOR=abcde;  
AGENCIA\_COD\_CEDENTE=3376-4/12678-0;  
CONTRATO=00.00.00;  
CPF=20.774.232/0001-34;  
CAMPO\_LIVRE=666666677777777777775555;

Obs.: Com nova versão da LIB Boletto (atualizada dia 06/06/2014), é possível “zerar” o fator de vencimento na linha digitada se a data de vencimento fornecida for um texto qualquer. Exemplo de valores para os parâmetros:

VALOR\_EM\_CENTAVOS=0;  
VALOR\_DOCUMENTO\_EM\_CENTAVOS=0;  
VALOR\_COBRADO\_EM\_CENTAVOS=0;  
VENCIMENTO=contra apresentacao;

### 3.31 ENVIO DE ARQUIVO

<SENDFILE NAME=XXXXX > comando para enviar arquivo para o Servidor.

O parâmetro NAME informa o nome do arquivo a ser enviado.

Qualquer tipo de arquivo pode ser enviado, texto ou não texto.

Abaixo, um exemplo em PHP, de como o Servidor deve capturar o arquivo recebido:

```
<?php
var_dump ($_FILES);
copy ($_FILES["file"]["tmp_name"], "C:\\wamp\\www\\".$_FILES["file"]["name"]);

$tela = $_GET["TELA"];
switch ($tela){
case 1:
    echo "<GET TYPE=HIDDEN NAME=TELA VALUE=2>";
    echo "<CONSOLE>";
    echo "<BR><BR>ENTER PARA ENVIAR";
    echo "</CONSOLE>";
    echo "<GET TYPE=ANYKEY>";
    echo "<SENDFILE NAME=claro.bmp>";

break;
}
?>
```

**Importante:** Este comando substitui o POST, ou seja, ele é o último comando que deve ser inserido na lista da sessão.

Nenhum outro comando após ele será executado na sessão.

### **3.32 GRAVAÇÃO EM ARQUIVO DAS CAPTURAS – POSTOFF (uso para offline)**

<POSTOFF NAME=XXXXX> comando para que o POS grave no arquivo indicado no parâmetro NAME as informações coletadas.

Este comando grava um registro no arquivo informado, com as informações dos dados coletados pelos comandos anteriores ao POSTOFF.

Após a gravação do registro, o NAVS retorna para a posição do INIT.

O formato do arquivo é o mesmo do dado enviado ao Servidor pelo comando POST.

Exemplo: nome1=campo1&nome2=campo2&

Os registros são separados pelo caracter '%’.

### **3.33 IMPRESSÃO OFFLINE - PRINTEROFF**

Este comando executa a impressão dos dados como no comando PRINTER, com a possibilidade de imprimir os dados coletados offline (até o comando POSTOFF).

<PRINTEROFF> inicia texto para a impressora,  
</PRINTEROFF> finaliza texto para a impressora,

A mudança de linha é realizada pelo comando <BR>.

Para imprimir dados coletados offline, use o nome da variável entre o caracter '%’.

Exemplo: “S/N: %SERIE%”

### 3.34 *HTMLDEBUG*

Para verificar tudo o que vem do Servidor para o POS, faz a impressão do log de toda a instrução enviada.

<HTMLDEBUG ON> ativa a impressão do log.

<HTMLDEBUG OFF> desativa a impressão do log.

### **3.35 VALIDAÇÃO OFFLINE - VALIDATEOFF**

< VALIDATEOFF NAME=XXXXX> comando para validar as informações capturadas Offline.  
O parâmetro NAME informa o nome do arquivo com os dados a serem validados com as informações coletadas até o POSTOFF.

Cada dado deve ser separado pelo caracter: '%'.  
'

Exemplo do conteúdo deste arquivo:  
%528922219%VX520%5327517780006%

Todos os dados coletados, deverão estar dentro deste arquivo.  
No exemplo acima,

Caso a validação esteja correta, o NAVS prossegue para o próximo comando, que poderá ser a impressão Offline.

Caso a validação seja negada, ou seja, algum dado coletado não se encontra no arquivo informado, o NAVS apaga as informações coletadas Offline e retorna para o comando INIT.

## Especificação de Interface

---

### EXEMPLO OFFLINE:

```
<INIT>
<GET TYPE=SERIALNO NAME=SERIE>
<GET TYPE=MODEL NAME=MODPOS>

<CONSOLE><BR><BR>DIGITE O RG</CONSOLE>
<GET TYPE=FIELD NAME=NUMRG SIZE=9 COL=0 LIN=5>

<VALIDATEOFF NAME=ARQVAL>

<PRINTEROFF>
TRANSAÇÃO APROVADA OFFLINE<BR><BR>
S/N: %SERIE%<BR>
MODELO: %MODPOS%<BR>
RG: %NUMRG%<BR>
</PRINTEROFF>

<POSTOFF>
```

Na impressora sairá a informação abaixo, caso a validação dos dados esteja correta (VALIDATEOFF):

### TRANSAÇÃO APROVADA OFFLINE

```
S/N: 521000999
MODELO: VX520
RG: 12123433-2
```

A seguinte informação será gravada no arquivo:

```
SERIE=521000999&MODPOS=VX520&NUMRG=12123433-2&
```

Supondo que no arquivo ARQVAL informado no comando VALIDATEOFF, contenha os dados abaixo:  
%528123456%VX520%105554446%999555476%123456789%

O NAVS aceitará a captura, aprovando a transação e imprimindo os dados, caso o numero de série do terminal seja o 528123456, e o modelo do POS seja o VX520 e o RG coletado seja: 105554446, ou 999555476 ou 123456789.

Para remover arquivos Offline do POS, use o comando DELFILE (capítulo 3.16):

```
<DELFILE NAME=xxxxx ERR=zzzzz>
```

### 3 OFFLINE com LUA / SQLite

Desenvolvimento de transações no NAVS em modo Offline, agora de uma forma mais ampla, pode ser efetuado através da linguagem LUA com o banco de dados SQLite intergados.

Para fazer debug do LUA utilizando a porta serial do POS, utilize o seguinte comando no script lua:

`_debug("teste");` ou `_debug(StringExemplo)`, onde "StringExemplo" é uma string que recebeu algum valor em algum ponto do script LUA desenvolvido; e "teste" é uma string literal.

Tudo que estiver dentro de "`_debug("xxxx")`", ou seja, tudo o que for "xxxx" ", será enviado para a serial do POS.

Obs.: Ao criar arquivos com extensão "db" não ultrapassar 8 bytes no tamanho do nome do arquivo, sem contar com a identificação da extensão (.db).

Obs2.: A sintaxe do comando foi alterada de "debug(xxx)" para "\_debug(xxx)".

## 4 Compilador Lua

Essa ferramenta serve para não ter a necessidade de colocar todos os seus scripts .lua no POS e para mantê-los mais seguros.

Ela é um pacote zip que deverá ser extraído e que contém os seguintes arquivos:

- "ChunkSpy.lua";
- "lua.bat";
- "lua.exe"
- "lua52.dll";
- "lua52.exe";
- "luac.exe";
- "luac52.exe";
- e "wlua52.exe".

Para gerar arquivo compilado lua, é necessário alterar um único arquivo desse pacote : o "lua.bat".

Exemplo para "lua.bat":

```

1 C:
2 cd "C:\Users\Skytef\Desktop\Minhas Ferramentas\Compilador Lua"
3 luac -o temp.o -l ^
4     script1.lua ^
5     script2.lua ^
6     script3.lua
7
8
9 pause
10 lua ChunkSpy.lua temp.o --rewrite "arm standard" -o ArqFinal.o

```

Onde :

- na linha 2 deverá ter o endereço da pasta onde esses arquivos se encontram;
- "ArqFinal.o" poderá ser substituído pelo nome que quiser, porém a extensão ".o" deverá ser sempre mantida (Exemplo: "MinhaApp.o");
- Script1.lua, Script2.lua, e Script3.lua deverão ser substituídos pelos seu scripts;
- O sinal "^" no final da linha significa que existirá na próxima linha um novo script.lua a ser inserido na compilação.

Após fazer as alterações salve e execute o "lua.bat", é importante que o computador onde a compilação será gerada tenha um sistema operacional de 32 bits.

Para chamar as funções desses módulos use o comando dofile, porém substitua o módulo(scriptx.lua) pelo módulo com extensão ".o" gerado na compilação :

Exemplo:

```

<data destination=parser>
    dofile("\ArqFinal.o")
    rodaTeste()
</data>

```

## 5 MIFARE – leitura e escrita

Apenas para os modelos de POS que tenham esta entrada.

O comando para leitura de dados de cartão Mifare está especificado no capítulo: 3.2.19 (Leitura de Dados do Mifare).

Para efetuar a escrita de dados em um determinado setor do cartão Mifare (contactless), devemos usar o comando CTLWRITE, cujo conteúdo dos dados fica no corpo do comando, conforme abaixo:

```
<CTLWRITE KEY=yyyyyyyyyyyy KTYPE=a BLOCK=0 SECTOR=0 BCOUNT=3>  
dado  
</ CTLWRITE >
```

Onde:

dado: é a informação a ser gravada **(em HEXA)**.

## **6 Descrição da Mensagem do POS ao Servidor**

Todos os campos capturados serão enviados para o Servidor quando for encontrado o comando POST, da seguinte forma:

nome1=campo1&nome2=campo2&...

Onde, “nome” é o argumento coletado no parâmetro NAME do campo em questão, e “campo” é o valor capturado pelo POS.

A ordem na qual os campos e seus respectivos valores aparecem na string de mensagem é a ordem em que o comando de captura dos respectivos campos aparecem no corpo do XML.

## 7 Exemplo de uso

Como exemplo, a sequência da comunicação entre POS e Servidor poderá ser:

- 1) POS inicia comunicação com Servidor baseado no arquivo XML previamente carregado no POS.
- 2) Servidor responde com os comandos para iniciar a transação no POS:

```
<INIT>
<CONSOLE>-----VENDA-----<BR> PASSE O CARTÃO<BR><\CONSOLE>
<GET NAME=CARTAO TYPE=CARD TRACKS=3>
<CONSOLE> DIGITE O VALOR<BR><\CONSOLE>
<GET NAME=VALOR TYPE=FIELD SIZE=5 COL=8 LIN=2>
<POST>
```

- 3) POS mostrará no display:

```
"-----VENDA-----"
" PASSE O CARTAO"
```

O POS, então, aguardará a passagem do cartão pelo usuário e irá para o próximo passo da transação:

```
" DIGITE O VALOR "
```

Ao iniciar a digitação, teremos no display do POS:

```
" DIGITE O VALOR "
  23
```

Após apertar o Enter no teclado, o POS enviará ao Servidor o texto abaixo:

```
CARTAO="trilhas 1 e 2"&VALOR=23
```

- 4) Servidor poderá responder ao POS:

```
<CONSOLE>CARTAO ACEITO<BR><\CONSOLE>
<CONSOLE>QUANTIDADE: 23<BR><\CONSOLE>
<PRINTER>CARTAO: XXXXXXXXXXXX</PRINTER >
<INIT>
<CONSOLE>-----VENDA-----<BR> PASSE O CARTÃO<BR><\CONSOLE>
<GET NAME=CARTAO TYPE=CARD TRACKS=3>
<CONSOLE> DIGITE O VALOR<BR><\CONSOLE>
<GET NAME=VALOR TYPE=FIELD SIZE=5 COL=8 LIN=2>
<POST>
```

## Especificação de Interface

---

5) O POS mostrará no display:

“CARTAO ACEITO”  
“QUANTIDADE: 23”

E, imprimirá:

“CARTAO: XXXXXXXXXXXX”

6) O POS ficará pronto para a próxima transação, mostrando no display:

“-----VENDA-----”  
“ PASSE O CARTAO”

7) O POS sempre voltará para este ponto sempre que a tecla CANCELA for acionada:

“-----VENDA-----”  
“ PASSE O CARTAO”

## 8 Configuração do NAVS

Para fazer as configurações no NAVS relativas à Conexão, Rede e Servidor aperte a tecla '#' durante aproximadamente 5 segundos, e siga os passos abaixo:

Primeiramente é solicitada uma senha de acesso, controlada por uma variável do sistema operacional. A variável é \*OPP.

Caso esta variável não exista, basta acionar Enter após a solicitação da senha.

### CONFIGURACAO

**1) CONEXAO** = Escolha do tipo de comunicação, opte por: **1 GPRS, 2 ETHERNET, 3 WIFI.**

**2) REDE** = Configure de acordo com a opção escolhida em CONEXÃO:

- Se escolheu **1 GPRS**:

**APN:** Digitar a APN da operadora. Ex.: claro.com.br/ gprs.oi.com.br/ tim.br/ zap.vivo.com.br – Todas estas APN publicas.

**USUARIO:** Sempre o nome da operadora. Ex.: claro / oi / tim / vivo.

**SENHA:** Sempre o nome da operadora. Ex.: claro / oi / tim / vivo.

**PING:** Definir o tempo em minutos (padrão 15 min.) para envio do ping para manter a rede conectada. Para não enviar o ping colocar o valor 0. Obs.: são enviados 56 bytes.

- Se escolheu **2 ETHERNET**:

**IP:** Digite o IP do ponto de rede que está conectado no POS.

**MASCARA:** Digite a MASCARA do ponto de rede que está conectado no POS.

**GATEWAY:** Digite a GATEWAY do ponto de rede que está conectado no POS.

**DNS1:** Digite o DNS1 do ponto de rede que está conectado no POS.

**DNS2:** Digite o DNS2 do ponto de rede que está conectado no POS.

Caso utilize o modo DHCP, atualize os campos acima com: **0.0.0.0**

**PING:** Definir o tempo em minutos para envio do ping para manter a rede conectada. Para não enviar o ping colocar o valor 0.

## Especificação de Interface

---

- Se escolheu **3 WIFI**:

**SSID**: Digite o SSID da rede Wi-Fi que está conectado no POS.

**SENHA**: Digite a senha da rede Wi-Fi que está conectado no POS.

**IP**: Digite o IP da rede Wi-Fi que está conectado no POS.

**MASCARA**: Digite a MASCARA da rede Wi-Fi que está conectado no POS.

**GATEWAY**: Digite a GATEWAY da rede Wi-Fi que está conectado no POS.

**DNS1**: Digite o DNS1 da rede Wi-Fi que está conectado no POS.

**DNS2**: Digite o DNS2 da rede Wi-Fi que está conectado no POS.

Caso utilize o modo DHCP, atualize os campos acima com: **0.0.0.0**

**PING**: Definir o tempo em minutos para envio do ping para manter a rede conectada. **Usar o valor 1 para conexão WiFi para evitar a queda com o roteador.**

### 3) SERVIDOR

**1 IP DESTINO**: Digite o IP ou o endereço DNS do Servidor que responderá às solicitações do POS (sem o www). Exemplo: google.com.br.

**2 PORTA**: Digite a PORTA do Servidor que responderá às solicitações do POS.

**3 RECURSO**: Digite o nome da aplicação do Servidor.

**4 HOST**: Digite o nome do Host, pode ficar em branco na maioria dos casos, mas se estiver usando Ngnix é recomendável que preencha este campo.

**5 HTTPS**: Escolha **1 ATIVAR** ou **2 DESATIVAR**.

**6 METODO HTTPS**: Escolha **1 SSLv3** ou **2 TLSv1**.

### 4) DIVERSOS

**1 LINHA STATUS**

**2 SERIAL**

**3 SCROLL**

**4 CONTRASTE IMPRESS.**

## Especificação de Interface

---

### 5 ATIVAR PAGAMENTO

### 6 KEEP ALIVE

#### 4.1) LINHA STATUS

##### 1 POSIÇÃO

##### 2 HORA

##### 4.1.1) POSIÇÃO LINHA DE STATUS

**1 TOPO:** As mensagens de status do POS ficarão na primeira linha da console.

**2 RODAPÉ:** As mensagens de status do POS ficarão na última linha da console.

**3 AUSENTE:** As mensagens de status do POS não aparecerão na console.

##### 4.1.2) HORA LINHA STATUS

**1 MOSTRAR:** Mostra hora na linha de status.

**2 OCULTAR:** Não mostra hora na linha de status.

#### 4.2) SERIAL

Configure os parâmetros da comunicação Serial: Baudrate, Paridade, Data Bits, Stop Bits e Timeout (SER-TIMEOUT em milissegundos).

OBS.: Caso a opção TIMEOUT esteja vazia ou com o valor '0', não haverá o controle de Timeout.

#### 4.3) SCROLL

Configure as teclas para efetuar o scroll de linhas, teclas de subida e descida. Exclusiva para o Vx520.

**1 SCROLL UP:** Tecla para efetuar o scroll para cima.

**2 SCROLL DOW:** Tecla para efetuar o scroll para baixo.

Selecione a função Scroll (Up ou Dow) e aperte a tecla desejada, no visor aparecerá o código hexa da tecla.

## Especificação de Interface

---

Para finalizar a operação pressione CANCEL.

Não pressione ENTER, senão a tecla Enter será selecionada para o Scroll.

Abaixo alguns exemplos de caracteres hexadecimal das teclas:

TECLA	VALOR HEXA
F1	7A
F2	7B
F3	7C
F4	7D
Roxa 1	61
Roxa 2	62
Roxa 3	63
Roxa 4	64

### 4.4) CONTRASTE IMPRESS.

Configure o nível de contraste para a impressão.

**1 NÍVEL 1:** Configura para o nível 1, padrão da impressora.

**2 NÍVEL 2:** Configura para o nível 2.

**3 NÍVEL 3:** Configura para o nível 3.

**4 NÍVEL 4:** Configura para o nível 4.

**5 NÍVEL 5:** Configura para o nível 5.

**6 NÍVEL 6:** Configura para o nível 6.

**7 NÍVEL 7:** Configura para o nível 7.

**8 NÍVEL 8:** Configura para o nível 8, nível máximo.

## Especificação de Interface

---

### 4.5) KEEP ALIVE

Configure se o número de série do terminal será enviado de tempos em tempos para um endereço e porta previamente definidos.

**1 ATIVAR / DESATIVAR :** Define se keep alive estará ou não ativado.

**2 INTERVALO:** Define tempo de intervalo para envio dos pacotes com o número de série.

**3 IP:** Define o ip de destino, que pode ser diferente do IP do servidor onde está a aplicação.

**4 PORTA:** Define a porta de destino, que pode ser diferente da Porta do servidor onde está a aplicação.

Obs.: Com o Keep Alive ativo, o POS abre um socket e envia somente o número de série e depois fecha o socket. A frequência é definida de acordo com o Intervalo preenchido pelo usuário.

### 5) VERSOES

#### 1 NAVS

#### 2 SISTEMA OPERACIONAL

#### 3 EOS

#### 4 IP

#### 5 EMV

#### 6 CLISITEF

#### 5.1) NAVS

Informa a versão de NAVS em uso no terminal.

#### 5.1) SISTEMA OPERACIONAL

Informa a versão do sistema operacional em uso no terminal.

#### 5.1) EOS

Informa a versão de EOS em uso no terminal.

## **Especificação de Interface**

---

### **5.1) IP**

Informa a versão de Interface Padrão em uso no terminal.

### **5.1) EMV**

Informa a versão de EMV em uso no terminal.

### **5.1) CLISITEF**

Informa a versão de CLISITEF em uso no terminal.

## 9 Status do Terminal

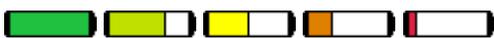
Na linha de status da console dos terminais, o NAVS apresenta o status do nível de sinal, operadora, nível da bateria, fim de papel e status da comunicação com a seguintes informações:

- Nível do SINAL: 

O nível do sinal GPRS e Wi-Fi é apresentado com as barras indicando o nível. Caso o sistema operacional não consiga informar o sinal, será apresentado a figura: 

Quando o NAVS estiver configurado com a conexão Ethernet, será apresentada a figura: 

- Operadora (GPRS): **CLARO**  **TIM** **VIVO**

- Nível da BATERIA: 

O nível da bateria será apresentado apenas para os modelos móveis (ex. Vx680), conforme o nível da bateria, através das figuras acima.

- Status de Fim de Papel: 

- Hora e Minuto do terminal: **HH:MM**

- Status da COMUNICAÇÃO: **XXXX**

O status da comunicação será apresentado em quatro caracteres com as seguintes informações:

<b>CX..:</b>	Conectando na rede.
<b>Rdok:</b>	Rede OK, indica conexão física com sucesso.
<b>StNw:</b>	Inicia conexão lógica.
<b>Cxok:</b>	Conexão lógica com sucesso, conectado com a operadora.
<b>Wfok:</b>	Encontrou a rede Wi-Fi.
<b>Htok:</b>	Conectado com o Servidor.
<b>HOST:</b>	Servidor OK.
<b>Hoff:</b>	Servidor offline.
<b>RX..:</b>	Recebendo dados.
<b>Rxok:</b>	Dados do Servidor recebidos.
<b>TX..:</b>	Enviando dados ao Servidor.
<b>CXer:</b>	Erro de conexão.
<b>DNSe:</b>	Erro de DNS, tentando encontrar página.
<b>....:</b>	Reconectando na rede.

## 10 Histórico das Versões

NAVS	Alteração	Autor/Revisor
1.4.8 03/06/2013	<ul style="list-style-type: none"> <li>- Touch no Vx680: comandos para desenho de figuras e captura.</li> <li>- Ajuste na leitura do cartão magnético.</li> <li>- DNS na configuração do acesso ao Servidor.</li> <li>- Ajuste no HTTPS.</li> <li>- Paginação de tela.</li> </ul>	Orlando Palermo
1.4.9 14/06/2013	<ul style="list-style-type: none"> <li>- Ajuste no Touch com CAPTURE.</li> <li>- Ajuste no HTTPS para o comando FILE.</li> <li>- Comando para captura de data e hora (GET DATETIME).</li> <li>- Ajuste no comando SERIAL com timeout configurado (SER-TIMEOUT).</li> <li>- Aumento no tamanho de entrada dos dados na Configuração.</li> <li>- Ping para manter conexão ativa no GPRS, configurável.</li> </ul>	Orlando Palermo
1.5.2 31/07/2013	<ul style="list-style-type: none"> <li>- Aumento do limite de quantidade de imagens para a tela para 50.</li> <li>- Comando SERIAL sem controle de Timeout.</li> <li>- Comando DELFILE para remover arquivos no POS.</li> <li>- Comando de impressão de Boleto.</li> <li>- Comandos para OFFLINE, capítulo 4.0.</li> </ul>	Orlando Palermo
1.5.4 30/08/2013	<ul style="list-style-type: none"> <li>- Comando de impressão de Boleto, com status de retorno.</li> <li>- Suporte a cookies na comunicação HTTP.</li> </ul>	Orlando Palermo
1.5.5 06/09/2013	<ul style="list-style-type: none"> <li>- Comando CHGPRNFNT: novo comando com diversas fontes para impressora.</li> <li>- Comando SELECT, para escolha de ítem em menus.</li> </ul>	Orlando Palermo
1.5.7 16/09/2013	<ul style="list-style-type: none"> <li>- Comando FILE com novo parâmetro WRT, para opção de sobrescrever ou não um arquivo já existente.</li> <li>- Apresentação da Hora e Minuto na linha de status</li> </ul>	Orlando Palermo
1.5.9 26/09/2013	<ul style="list-style-type: none"> <li>- Comando GET FIELD com ALPHA=1, aceitando espaços.</li> <li>- Adição de um parâmetro de configuração para o método HTTPS: "SSL_METHOD" que recebe valores 1 ou 2, onde 1(indica que o protocolo utilizado será SSLv3, mais recomendável por sua confiabilidade) e 2(indica que o protocolo utilizado será TLSv1).</li> </ul>	Thais Marques
1.6.1 01/11/2013	<ul style="list-style-type: none"> <li>- Comando para captura da versão do Sistema Operacional do POS (GET OSVERSION).</li> </ul>	Thais Marques

## Especificação de Interface

NAVS	Alteração	Autor/Revisor
	<ul style="list-style-type: none"> <li>- Comando para captura de CPF, CNPJ e Data.</li> <li>- Aumento do limite dos itens de menu do comando GET SELECT, com a criação de um novo comando: "SELECT".</li> <li>- Comando TONE: novo comando que faz o POS emitir sons;</li> <li>- Adição do parâmetro de configuração do nível de contraste na impressora no menu de configurações do NAVS.</li> </ul>	
1.6.2 12/11/2013	<ul style="list-style-type: none"> <li>- Correção no retorno do comando SELECT.</li> <li>- Correção na estrutura de navegação do NAVS, para que as telas não travem quando o botão cancela for pressionado enquanto o comando SELECT está ativo.</li> <li>- Correção do comando CAPTURE.</li> </ul>	Thais Marques
1.6.3 13/12/2013	<ul style="list-style-type: none"> <li>- Adição de status de retorno no comando &lt;PRINTER&gt;;</li> <li>- Adições no menu de configuração do NAVS:               <ul style="list-style-type: none"> <li>- Opção para ativar pagamento;</li> <li>- Opção de mostrar ou não hora na linha de status;</li> </ul> </li> <li>Adição do caractere "_" no teclado, pode ser inserido quando a tecla "0" é apertada e logo em seguida a tecla "#" é apertada 3 vezes.;</li> <li>- Adição do caractere "@" no teclado, pode ser inserido quando a tecla "0" é apertada e logo em seguida a tecla "#" é apertada 4 vezes;</li> <li>- Adição do comando SET TIME;</li> <li>- Alteração no comando SENDFILE para enviar arquivos não texto, e removida o parâmetro de retorno.</li> <li>- Ajuste nos comandos para captura de CPF, CNPJ e Data.               <ul style="list-style-type: none"> <li>- Offline com LUA / Sqlite.</li> </ul> </li> </ul>	Thais Marques
1.6.4 18/12/2013	<ul style="list-style-type: none"> <li>- Novo ajuste nos comandos para captura de CPF, CNPJ e Data.</li> <li>- Observações para utilização dos comandos SELECT, CONSOLE, PRINTER, PRNFNT e CHGPRNFNT.</li> </ul>	Thais Marques
1.6.5 26/12/2013	<ul style="list-style-type: none"> <li>- Nova correção nos comandos para captura de CPF, CNPJ e Data.</li> <li>- Correção para o uso do comando SELECT junto com o comando INIT.</li> <li>- Correção no comando SELECT para aceitar as teclas numéricas como teclas de navegação, antes essas teclas não eram capturadas.</li> </ul>	Thais Marques
1.6.6 09/01/2014	<ul style="list-style-type: none"> <li>- Ajuste no comando GET FIELD com ALPHA=1, quando usa-se o caracter de espaço.</li> <li>- Especificação atualizada com a informação do uso do parâmetro PING=1 na conexão Wi-Fi.</li> </ul>	Orlando Palermo

## Especificação de Interface

NAVS	Alteração	Autor/Revisor
1.6.7 16/01/2014	<ul style="list-style-type: none"> <li>- Adição do parâmetro HOST na configuração do servidor.</li> <li>- Adição do caractere “!” no teclado, pode ser inserido quando a tecla “0” é apertada e logo em seguida a tecla “#” é apertada 5 vezes;</li> <li>- Adição de um parâmetro de entrada NOCLS para o comando CONSOLE, para não limpar a tela.</li> </ul>	Thais Marques
1.6.9 12/02/2014	<ul style="list-style-type: none"> <li>- Correção do comportamento do &lt;CONSOLE&gt; ao tratar o parâmetro NOCLS.</li> <li>- Correções no comando &lt;SET TYPE=TIME ..&gt;, captura de dados, e travamento dos POS nas operações posteriores no POS.</li> <li>- Correções no comando &lt;SELECT&gt; (aceita “*” e “#” como parâmetros, nova correção de uso junto ao comando &lt;INIT&gt;, e atualização da documentação referente ao mesmo).</li> <li>- Correção do comando de Scroll (uso exclusivo do Vx520). <ul style="list-style-type: none"> <li>- Adição de uma mensagem de status de conexão.</li> <li>- Identificação do tipo de aparelho e disponibilização da configuração de rede conforme as possibilidades do dispositivo.</li> </ul> </li> </ul>	Thais Marques
1.7.1 10/03/2014	<ul style="list-style-type: none"> <li>- Inclusão no manual NAVS os capítulos 3.2.19 e 5.0, com comandos de leitura e escrita no Mifare. <ul style="list-style-type: none"> <li>- Criação do comando KEYBOARD.</li> </ul> </li> <li>- Suporte NAVS para aparelho Vx520 Contactless da Verifone.</li> <li>- Determinação de padrões de configuração automática da operadora, quando a conexão for GPRS.</li> <li>- Informações adicionais sobre imagens envolvendo os comandos CONLOGO, PRNLOGO.</li> </ul>	Orlando Palermo e Thais Marques
1.7.2 17/03/2014	<ul style="list-style-type: none"> <li>- Correção das informações sobre o comando PRNLOGO (especificação de resolução máxima da imagem de 128x64 para 384x240).</li> <li>- Adição de informações no comando CONLOGO.</li> <li>- Melhoria na velocidade de impressão.</li> </ul>	Thais Marques
1.7.4 30/04/2014	<ul style="list-style-type: none"> <li>- Adição do comando BINSCARD.</li> <li>- Adição do comando APDUCHIP.</li> <li>- Adição do comando DESNBS.</li> <li>- Adição do comando WRITE_AT.</li> <li>- Adição de novos fontes para o console e como consequência a criação do comando CHGCONFNT, que serve para alternar entre os fontes. <ul style="list-style-type: none"> <li>- Adição do comando CONFIG_NAVS.</li> </ul> </li> <li>- Correção no controle de funcionamento da impressora que permite que ela continue imprimindo após o comandos relacionados á impressão.</li> <li>- Adição de parâmetro no comando PRINTER (ENABLE_CANCEL).</li> </ul>	Thais Marques

## Especificação de Interface

NAVS	Alteração	Autor/Revisor
	<ul style="list-style-type: none"> <li>- Adição de parâmetro no comando INIT (KEEP_COOKIES).</li> <li>- Melhoria de performance do Lua.</li> <li>- Implementação de Log para auxiliar na programação Lua.</li> <li>- Ajuste de sincronia entre LUA e o NAVS, quando o botão CANCEL é acionado.</li> </ul>	
1.7.6 28/05/2014	<ul style="list-style-type: none"> <li>- Correção no comando SELECT.</li> <li>- Correção do comando de debug do LUA.</li> <li>- Adição do comando HTMLDEBUG.</li> <li>- Adição do comando de captura de horário com máscara (GET TYPE=HOUR).</li> <li>- Correção na recepção de parâmetros do KEEP_COOKIES (no comando INIT).</li> <li>- Inclusão dos parâmetros APDU1, APDU2, APDU3, APDU4 e APDU5 no comando APDUCHIP com intuito de ampliar o tamanho da APDU a ser executada.</li> <li>-Ajuste no comando dofile.</li> </ul>	Thais Marques
1.7.7 06/06/2014	<ul style="list-style-type: none"> <li>- Correção no comando SENDFILE para trabalhar com arquivos maiores que 32kbytes.</li> <li>- Nova alteração do comando de debug do lua (alteração de sintaxe também).</li> <li>- Atualização na parte de comunicação para que o Wi-fi também funcione com SSID oculto (somente para os modelos Verifone vx680 Wi-fi).</li> <li>- Disponibilização da ferramenta de encriptação de scripts feitos em lua.</li> <li>- Atualização da biblioteca de boleto.</li> </ul>	Thais Marques
1.7.8 24/07/2014	<ul style="list-style-type: none"> <li>- Adição do comando que cria QR Codes a partir de strings.</li> <li>- Adição do comando que gera código de barras 2 de 5 (ITF).</li> <li>- Adição do comando que gera código de barras 3 de 9.</li> <li>- Correção no comando APDUCHIP.</li> <li>- Adição do Keep Alive nas configurações NAVS.</li> <li>- Nova correção no SENDFILE para evitar que os arquivos sejam corrompidos.</li> <li>- Correção no GET TYPE=CPF e GET TYPE=CNPJ.</li> </ul>	Thais Marques
1.7.9 08/08/2014	<ul style="list-style-type: none"> <li>- Adição do comando DRAW.</li> <li>- Correção e adição de 2 novas funcionalidades ao SELECT.</li> <li>- Alterações no modo de funcionamento do comando CONFIG_NAVS.</li> <li>- Mudança de nome de alguns parâmetros de configuração do KEEP ALIVE.</li> </ul>	Thais Marques
1.8.0 12/08/2014	<ul style="list-style-type: none"> <li>- Ajustes na interface de comunicação, ficando funcional a comunicação ETH usando IP fixo, com DNS externo.</li> </ul>	Orlando Palermo

## Especificação de Interface

NAVS	Alteração	Autor/Revisor
1.8.1 15/09/2014	Correções no funcionamento entre o CAPTURE e o TIMEOUT, eliminando o erro de memória. Correções no SELECT para eliminar também um erro de memória, relacionado ao comando INIT.	Thais Marques
1.8.3 14/10/2014	Melhoria de performace. Implementação do Menu que informa as versões de software que estão no POS. Novas correções no comando SELECT.	Thais Marques
1.8.5 14/11/2014	Aumento da limitação do campo SERVER_HOST do comando Config_Navs ;	Thais Marques
1.8.6 20/11/2014	Nova Correção do SELECT relacionado ao uso conjunto com o comando INIT.	Thais Marques
1.8.7 19/12/2014	<ul style="list-style-type: none"> <li>- Ajuste na comunicação GPRS para manter por mais tempo a conexão com a operadora, mesmo que o PING estiver definido com um valor alto.</li> <li>- Ajuste na comunicação de rede wireless para funcionar em rede sem senha.</li> <li>- Adição de novos caracteres para a configuração NAVS. Os caracteres adicionados foram : '#','\$','(',')','&amp;','+','?',' ',';',':','°','ª',';','{','}','[',']','=','"', '%'</li> <li>- Ajuste na exibição do texto capturado pelo comando KEYBOARD.</li> <li>- Correção do comportamento do comando SELECT com a alteração do posicionamento da linha de status.</li> <li>- Adição do comando que retorna o status atual de comunicação (GET TYPE=STSCOMM).</li> <li>- Ajuste na especificação do comando "GET TYPE=ANYKEY".</li> <li>- Adição de conteúdo na especificação do comando Console para inserção de caracteres especiais na tela.</li> </ul>	Thais Marques
1.8.8 09/01/2015	<ul style="list-style-type: none"> <li>- Mudança na estrutura de resposta do comando Config_NAVS.</li> <li>- Melhorias para exibição na tela, reduzindo o número de vezes que a tela pisca.</li> <li>- Ajuste na gravação de parâmetros do menu de configuração NAVS.</li> </ul>	Thais Marques