



Manual de Programação em Impressoras Térmicas



ACD – Automação e Coleta de Dados
Departamento de Sistemas
Microsiga Software S/A
Av. Braz Leme, 1399
Casa Verde
São Paulo-SP

Índice

1. Considerações Iniciais	4
2. Sentido de Impressão	5
2.1. Padrão Zebra e Eltron (Programação ZPL e EPL)	5
2.2. Padrão Datamax (Programação DPL)	6
2.3. Padrão Intermec (Programação IPL)	7
3. Funções	8
3.1. MSCBPrinter	8
3.2. MSCBClosePrinter	9
3.3. MSCBBegin	10
3.4. MSCBEnd	10
3.5. MSCBSay	11
3.6. MSCBSayBar	12
3.6.1. Utilização do código 128	13
3.6.2. IDENTIFICAÇÃO DE APLICAÇÃO (Ais)	14
3.7. MSCBBox	17
3.8. MSCBLineH	18
3.9. MSCBLineV	18
3.10. MSCBLoadGraf	19
3.11. MSCBGraphic	20
3.12. MSCBChkStatus	20
3.13. MSCBInfoEti	21
3.14. MSCBWrite	22
3.15. MSCBVar	22
4. Exemplos	24
4.1. Padrão Datamax	24
4.1.1. Exemplo 01	24
4.1.2. Exemplo 02	24
4.1.3. Exemplo 03	25
4.2. Padrão Eltron	26
4.2.1. Exemplo 01	26
4.2.2. Exemplo 02	26
4.2.3. Exemplo 03	27
4.3. Padrão Intermec	28
4.3.1. Exemplo 01	28
4.3.2. Exemplo 02	28
4.3.3. Exemplo 03	29
4.4. Padrão Zebra	30
4.4.1. Exemplo 01	30
4.4.2. Exemplo 02	30
4.4.3. Exemplo 03	31
Anexo 01 (Catalogo de Fontes)	32
1. Catalogo de Fontes padrão DPL	32
2. Catalogo de Fontes padrão EPL	32
3. Catalogo de Fontes padrão IPL	32
4. Catalogo de Fontes padrão ZPL	32

1. Considerações Iniciais

Este documento não tem o objetivo de ensinar ao leitor a programação "ADVPL", nem a lógica de programação, pois neste caso presumimos que o leitor já tenha este conhecimento.

Neste documento serão apresentadas todas as funções necessárias para a geração e impressão de etiquetas com ou sem código de barras para impressoras térmicas.

Atualmente as funções para impressão térmica englobam as linguagem DPL, EPL, IPL e ZPL, sendo estas linguagens totalmente transparentes para o programador. Sendo assim o programador em questão não precisa ser um conhecedor das linguagem de programação para impressoras térmicas descritas acima, desenvolvidas por seus fabricantes.

Onde:

- **DPL** *é a linguagem de programação datamax;*
- **EPL** *é a linguagem de programação eltron;*
- **IPL** *é a linguagem de programação intermec;*
- **ZPL** *é a linguagem de programação zebra.*

2. Sentido de Impressão

O sentido de impressão não é o mesmo para todas impressoras térmicas. Alguns fabricantes de impressoras diferem o sentido de impressão um do outro, tendo suas próprias justificativas, podendo elas ser desenhpenho, qualidade ou até facilidade para o desenvolvedor.

Quando falamos em sentido de impressão, também é importante falarmos do ponto de origem (0,0) no posicionamento do desenho de imagens, linhas e textos, pois isto também é definido pelo fabricante tendo as mesmas regra justificadas por ele.

Abaixo são apresentadas os sentidos de impressão e ponto de origem para os padrões de programação e impressoras homologadas.

2.1. Padrão Zebra e Eltron (Programação ZPL e EPL)

A figura 01 ilustra o eixo cartesiano de trabalho, o ponto de origem do eixo cartesiano e principalmente o ponto de origem para início do texto ou figura, a qual fica no canto superior esquerdo (de cima para baixo) conforme figura 01.

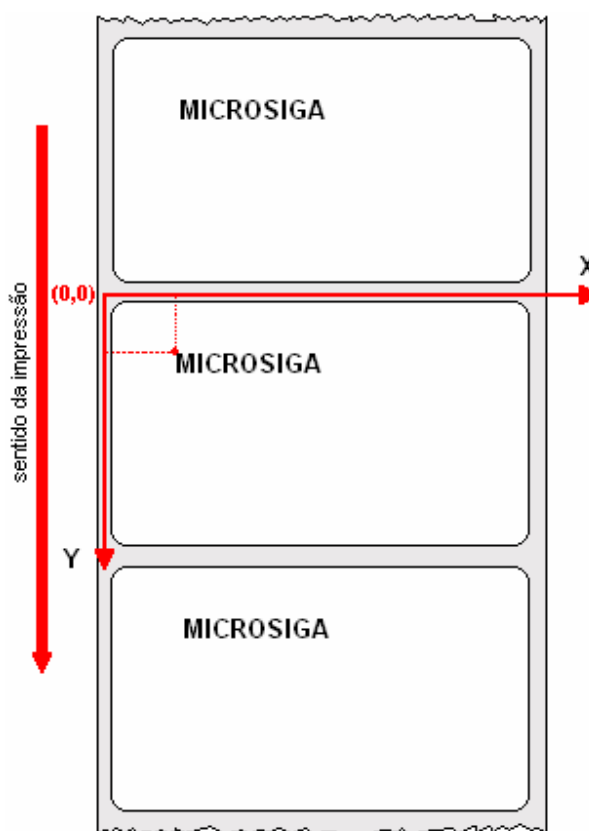


Figura 01 - Apresentação do sentido de impressão da imagem na etiqueta padrão zebra e eltron.

OBS: Alguns modelos de impressoras ELTRON, possuem o alinhamento da folha de etiqueta centralizado, por isso deve-se considerar o ponto de impressão da posição+a margem.

2.2. Padrão Datamax (Programação DPL)

Note na figura 02 que o quadrante do eixo cartesiano do padrão datamax é diferente do quadrante do eixo cartesiano da intermec e zebra. Também é importante observar que o ponto de origem para o desenho do texto ou figura inicia no canto inferior esquerdo do texto ou figura (ou seja, de baixo para cima do texto ou figura), diferentemente do padrão zebra e intermec.

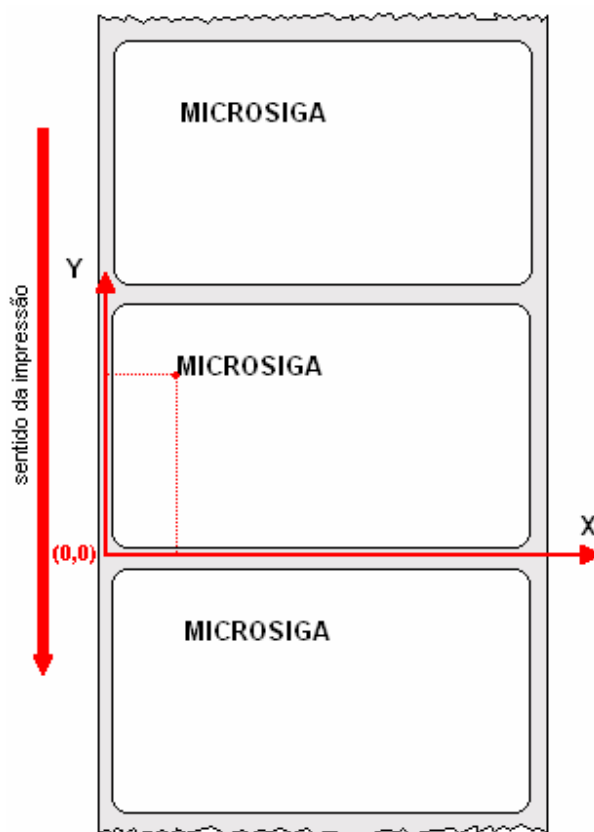


Figura 02 – Apresentação do sentido de impressão da imagem na etiqueta padrão datamax.

2.3. Padrão Intermec (Programação IPL)

Note que o quadrante do eixo cartesiano do padrão intermec é diferente do quadrante do eixo cartesiano da datamax e da zebra. Porém é importante observar que o ponto de origem para o desenho do texto ou figura inicia no canto superior esquerdo do texto ou figura (ou seja, a impressão do texto ou figura se dá de cima para baixo, a partir de sua origem), igual o padrão zebra e diferente do padrão datamax.

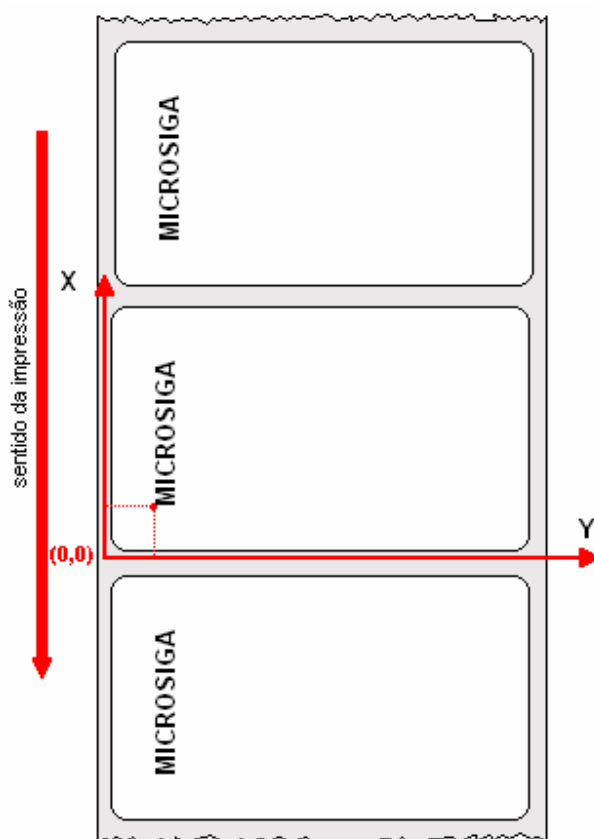


Figura 03 – Apresentação do sentido de impressão da imagem na etiqueta padrão Intermec.

3. Funções

Nesta sessão serão descritos as funções necessárias para desenvolver imagens de etiqueta para impressoras térmicas.

Os parâmetros que estiverem entre [], significa que não são OBRIGATÓRIOS, os parâmetros que estiverem com (*), significa que é uso exclusivo ZPL.

3.1. MSCBPrinter

Objetivo:

Permite realizar as configurações da impressora, as quais permitirão ao programador determinar as características de impressão, definidas pelos parâmetros abaixo.

Sintaxe:

MSCBPRINTER(cModelPrt, cPorta, nDensidade, nTamanho, ISrv, nPorta, cServer, cEnv, nMemoria, cFila, IDrvWin, cPathSpool)

Parâmetros	Descrição										
[cModelPrt]	String com o modelo de impressora: <table><tr><th>Fabricante</th><th>Impressoras</th></tr><tr><td>Datamax</td><td>ALLEGRO, ALLEGRO 2, PRODIGY, DMX, DESTINY, URANO, DATAMAX, OS 214, OS 314, PRESTIGE, ARGOX</td></tr><tr><td>Eltron</td><td>ELTRON, TLP 2722, TLP 2742, TLP 2844, TLP 3742, C4-8</td></tr><tr><td>Intermec</td><td>INTERMEC, 3400-8, 3400-16, 3600-8, 4440-16, 7421C-8</td></tr><tr><td>Zebra</td><td>S300, S400, S500-6, S500-8, Z105S-6, Z105S-8, Z160S-6, Z160S-8, Z140XI, S600, Z4M, Z90XI, Z170XI, ZEBRA</td></tr></table>	Fabricante	Impressoras	Datamax	ALLEGRO, ALLEGRO 2, PRODIGY, DMX, DESTINY, URANO, DATAMAX, OS 214, OS 314, PRESTIGE, ARGOX	Eltron	ELTRON, TLP 2722, TLP 2742, TLP 2844, TLP 3742, C4-8	Intermec	INTERMEC, 3400-8, 3400-16, 3600-8, 4440-16, 7421C-8	Zebra	S300, S400, S500-6, S500-8, Z105S-6, Z105S-8, Z160S-6, Z160S-8, Z140XI, S600, Z4M, Z90XI, Z170XI, ZEBRA
Fabricante	Impressoras										
Datamax	ALLEGRO, ALLEGRO 2, PRODIGY, DMX, DESTINY, URANO, DATAMAX, OS 214, OS 314, PRESTIGE, ARGOX										
Eltron	ELTRON, TLP 2722, TLP 2742, TLP 2844, TLP 3742, C4-8										
Intermec	INTERMEC, 3400-8, 3400-16, 3600-8, 4440-16, 7421C-8										
Zebra	S300, S400, S500-6, S500-8, Z105S-6, Z105S-8, Z160S-6, Z160S-8, Z140XI, S600, Z4M, Z90XI, Z170XI, ZEBRA										
[cPorta]	String com a porta										
[nDensidade]	Número com a densidade referente a quantidade de pixel por mm. Este parâmetro só deve ser informado quando o parâmetro cModelPrt não for informado, pois cModelPrt o atualizará automaticamente. A utilização deste parâmetro deverá ser usado quando não souber o modelo da impressora, a aplicação entenderá que se trata de uma impressora Zebra. O tamanho da etiqueta será necessário quando a mesma não for continua.										
[nTamanho]	Tamanho da etiqueta em Milímetros. Lembrando que este tamanho só deve ser passado se a etiqueta for continua.										
[ISrv]	Se .t. imprime no server,.f. no client. O seu valor padrão é .f.										
[nPorta]	Número da porta de outro server										
[cServer]	endereço IP de outro server										
[cEnv]	environment do outro server										
[nMemoria]	Número com bloco de memória da impressora térmica. Caso seja enviada muita informação para a impressora, a fim que esta venha imprimir (sobrecarregando a memória), pode ocorrer perda de dados. Por outro lado, se for informado blocos muito pequenos de memória, implicará na diminuição da performance da impressora. Sendo assim o										

	programador deverá fazer uma avaliação para ver o que melhor se adequa a sua situação.
[cFila]	Diretório onde será gravada as filas
[IDrvWin]	Indica se será utilizando os drivers do windows para impressão
[cPathSpool]	Caminho do diretório onde serão geradas as filas de impressão

Exemplo usando o padrão de programação Zebra (família ZPL)

```

Local nX
Local cPorta
/*
configuracoes mais utilizadas de porta
cPorta := "COM2:9600,n,8,1"
cPorta := "COM2:9600,n,8,2"
cPorta := "COM2:9600,n,7,1"
cPorta := "COM2:9600,n,7,2"
cPorta := "COM2:9600,e,8,1"
cPorta := "COM2:9600,e,8,2"
cPorta := "COM2:9600,e,7,1"
cPorta := "COM2:9600,e,7,2"
cPorta := "LPT1"
*/
cPorta := "COM1:9600,N,8,1"
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)
For nx:=1 to 3
  MSCBBEGIN(1,6)
  MSCBSAY(10,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09, Strzero(nX,10), "N", "0", "032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N", "0", "020,030")
  MSCBEND()
Next
MSCBCLOSEPRINTER()

```

3.2. MSCBClosePrinter

Objetivo:

Finaliza a conexão com a impressora.

Sintaxe:

MSCBClosePrinter()

Exemplo usando o padrão de programação Zebra (família ZPL):

```

Local cPorta := "COM1:9600,N,8,1"
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)
For nx:=1 to 3
  MSCBBEGIN(1,6)
  MSCBSAY(10,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09, Strzero(nX,10), "N", "0", "032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N", "0", "020,030")
  MSCBEND()
Next
MSCBCLOSEPRINTER()

```

3.3. MSCBBegin

Objetivo:

Inicializa a montagem da imagem para cada etiqueta.

Sintaxe:

MSCBBEGIN(nxQtde,nVeloc,nTamanho,lSalva)

Parâmetros	Descrição
[nQtde]	Quantidade de cópias
[nVeloc]	Velocidade (1,2,3,4,5,6) polegadas por segundo
[nTamanho]	Tamanho da etiqueta em Milímetros.
[lSalva]	Variável utilizada somente para definição de form's. O valor default da variável é .f.

Exemplo usando o padrão de programação Zebra (família ZPL):

```
Local nX
Local cPorta := "COM1:9600,N,8,1"
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)
For nx:=1 to 3
  MSCBBEGIN(1,6)
  MSCBSAY(10,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09, Strzero(nX,10), "N", "0", "032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N", "0", "020,030")
  MSCBEND()
Next
MSCBCLOSEPRINTER()
```

3.4. MSCBEnd

Objetivo:

Finaliza a montagem da imagem, que foi inicializada por um "MSCBBegin()".

Sintaxe:

MSCBEND()

Exemplo usando o padrão de programação Zebra (família ZPL):

```
Local nX
Local cPorta := "COM1:9600,N,8,1"
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)
For nx:=1 to 3
  MSCBBEGIN(1,6)
  MSCBSAY(10,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09, Strzero(nX,10), "N", "0", "032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N", "0", "020,030")
  MSCBEND()
Next
MSCBCLOSEPRINTER()
```

3.5. MSCBSay

Objetivo:

Imprime uma string.

Sintaxe:

MSCBSAY(nXmm, nYmm, cTexto, cRotacao, cFonte, cTam, IReverso, ISerial, cIncr, IZerosL, INoAlltrim)

Parâmetro	Descrição										
nXmm	Posição X em Milímetros										
nYmm	Posição Y em Milímetros										
cTexto	String a ser impresso ou itens especificando uma variável "@".(Ex: "@2"). Obs: quando for especificado uma variável, o seu conteúdo deverá ser apenas o caractere "@" seguido de um número, "@1" ou "@2" e assim por diante. Vide item 3.15.										
cRotação	String com o tipo de Rotação (N,R,I,B): <table border="1"> <thead> <tr> <th>Instrução</th><th>Descrição</th></tr> </thead> <tbody> <tr> <td>N</td><td>Normal</td></tr> <tr> <td>R</td><td>Cima para baixo</td></tr> <tr> <td>I</td><td>Invertido</td></tr> <tr> <td>B</td><td>Baixo para cima</td></tr> </tbody> </table>	Instrução	Descrição	N	Normal	R	Cima para baixo	I	Invertido	B	Baixo para cima
Instrução	Descrição										
N	Normal										
R	Cima para baixo										
I	Invertido										
B	Baixo para cima										
cFonte	String com os tipos de Fonte: <table border="1"> <thead> <tr> <th>Padrão</th><th>Tipos</th></tr> </thead> <tbody> <tr> <td>Zebra</td><td>(A,B,C,D,E,F,G,H,0) 0(zero)- fonte escalar</td></tr> <tr> <td>Datamax</td><td>(0,1,2,3,4,5,6,7,8,9) 9 – fonte escalar</td></tr> <tr> <td>Eltron</td><td>(0,1,2,3,4,5)</td></tr> <tr> <td>Intermec</td><td>(0,1,7,20,21,22,27)</td></tr> </tbody> </table>	Padrão	Tipos	Zebra	(A,B,C,D,E,F,G,H,0) 0(zero)- fonte escalar	Datamax	(0,1,2,3,4,5,6,7,8,9) 9 – fonte escalar	Eltron	(0,1,2,3,4,5)	Intermec	(0,1,7,20,21,22,27)
Padrão	Tipos										
Zebra	(A,B,C,D,E,F,G,H,0) 0(zero)- fonte escalar										
Datamax	(0,1,2,3,4,5,6,7,8,9) 9 – fonte escalar										
Eltron	(0,1,2,3,4,5)										
Intermec	(0,1,7,20,21,22,27)										
cTam	String com o tamanho da Fonte										
*[IReverso]	Imprime em reverso quando tiver sobre um box preto										
[ISerial]	Serializa o código										
[cIncr]	Incrementa quando for serial positivo ou negativo										
*[IZerosL]	Coloca zeros a esquerda no numero serial										
[INoAlltrim]	Permite brancos a esquerda e direita										

Exemplo usando o padrão de programação Zebra (família ZPL):

```

Local nX
Local cPorta := "COM1:9600,N,8,1"
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)
For nx:=1 to 3
  MSCBBEGIN(1,6)
  MSCBSAY(10,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09, Strzero(nX,10), "N", "0", "032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N", "0", "020,030")
  MSCBEND()
Next
MSCBCLOSEPRINTER()

```

3.6. MSCBSayBar

Objetivo:

Imprime código de barras.

Sintaxe:

MSCBSAYBAR(nXmm, nYmm, cConteudo, cRotacao, cTypePrt, nAltura, lDigVer, lLinha, lLinBaixo, cSubSetIni, nLargura, nRelacao, lCompacta, lSerial, cIncr, lZerosL)

Parâmetro	Descrição																
nXmm	Posição X em Milímetros																
nYmm	Posição Y em Milímetros																
cConteudo	String a ser impressa especificando uma variável "@" ou array somente quando o parâmetro <u>cTypePrt</u> for igual á MB07. Definição do Array: cConteudo :={"01","07893316010411"},; {"10","0000970100"+ MSCB128B() +"1"+ MSCB128C() },; {"37","0004"},; {"21","000494"} <ul style="list-style-type: none"> A posição 1 do array (ex: "01") informa o AI utilizado (que será visto no item 3.6.1. "Utilização do código de barras 128"). Já a posição 2 do array (ex: "07893316010411") é o conteúdo do AI. 																
cRotação	String com o tipo de Rotação: <table> <tr> <th>Instrução</th><th>Descrição</th></tr> <tr> <td>N</td><td>Normal</td></tr> <tr> <td>R</td><td>Cima para baixo</td></tr> <tr> <td>I</td><td>Invertido</td></tr> <tr> <td>B</td><td>Baixo para cima</td></tr> </table>	Instrução	Descrição	N	Normal	R	Cima para baixo	I	Invertido	B	Baixo para cima						
Instrução	Descrição																
N	Normal																
R	Cima para baixo																
I	Invertido																
B	Baixo para cima																
cTypePrt	String com o Modelo de Código de Barras: <table> <tr> <th>Instrução</th><th>Modelo de Código de Barras</th></tr> <tr> <td>MB01</td><td>Interleaved 2 of 5</td></tr> <tr> <td>MB02</td><td>Code 39</td></tr> <tr> <td>MB03</td><td>EAN 8</td></tr> <tr> <td>MB04</td><td>EAN 13</td></tr> <tr> <td>MB05</td><td>UPC A</td></tr> <tr> <td>MB06</td><td>UPC E</td></tr> <tr> <td>MB07</td><td>CODE 128</td></tr> </table> Obs: Caso o leitor queira utilizar o modelo do padrão de programação da impressora, o mesmo deverá consultar documentação do fabricante.	Instrução	Modelo de Código de Barras	MB01	Interleaved 2 of 5	MB02	Code 39	MB03	EAN 8	MB04	EAN 13	MB05	UPC A	MB06	UPC E	MB07	CODE 128
Instrução	Modelo de Código de Barras																
MB01	Interleaved 2 of 5																
MB02	Code 39																
MB03	EAN 8																
MB04	EAN 13																
MB05	UPC A																
MB06	UPC E																
MB07	CODE 128																
[nAltura]	Altura do código de Barras em Milímetros																
*[lDigver]	Imprime dígito de verificação																
[lLinha]	Imprime a linha de código																
*[lLinBaixo]	Imprime a linha de código acima das barras																
[cSubSetIni]	Utilizado no code128																
[nLargura]	Largura da barra mais fina em pontos default 3																
[nRelacao]	Relação entre as barras finas e grossas em pontos default 2																
[lCompacta]	Parâmetro fora de uso.																

[ISerial]	Serializa o código
[cIncr]	Incrementa quando for serial positivo ou negativo
[IZerosL]	Coloca Zeros a esquerda do conteúdo. Este parâmetro só é valido quando ISerial estiver ativado (.t.).

Exemplo usando o padrão de programação Zebra (família ZPL):

```

Local nX
Local cPorta := "LPT1"

MSCBPRINTER("S500-8",cPorta,      , 40 ,.f.)
MSCBLOADGRF("SIGA.GRF")
For nx:=1 to 3
  MSCBINFOETI("Exemplo 1","MODELO 1")
  MSCBBEGIN(1,6)
  MSCBBBOX(02,01,76,35)
  MSCBLineH(30,05,76,3)
  MSCBLineH(02,13,76,3,"B")
  MSCBLineH(02,20,76,3,"B")
  MSCBLineV(30,01,13)
  MSCBGRATIC(2,3,"SIGA")
  MSCBSAY(33,02,'PRODUTO',"N","0","025,035")
  MSCBSAY(33,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09,Strzero(nX,10),"N","0","032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N","0","020,030")
  MSCBSAYBAR(23,22,Strzero(nX,10),"MB07","C",8.36,.F.,.T.,.F.,,2,1)
  MSCBEND()
Next
MSCBCLOSEPRINTER()

```

3.6.1. Utilização do código 128

O código 128, permite a impressão de todos os 128 caracteres ASCII, incluindo caracteres alfanuméricos maiúsculos e minúsculos, bem como os caracteres de controle. Além disso existem alguns caracteres especiais que são de uso intrínseco ao código 128.

O código 128 está dividido em três subconjuntos: A, B e C dos quais apenas um está ativo de cada vez. Cada um dos códigos tem um sentido diferente dependendo do subconjunto ativo.

No protheus os subconjuntos A, B e C, são representados pelas funções **MSCB128A()**, **MSCB128B()** e **MSCB128C()** respectivamente.

O **subconjunto A** possibilita a impressão de todos os caracteres alfanuméricos maiúsculos, caracteres de controle e caracteres especiais.

O **subconjunto B** possibilita a impressão de todos os caracteres alfanuméricos maiúsculos, caracteres minúsculos, e caracteres especiais. Este subconjunto não possibilita a compactação, porém aceita quase todos os caracteres contidos na tabela ASCII.

O **subconjunto C** é utilizado para duplicar a densidade de códigos com conteúdo apenas **NUMÉRICO** e número **PAR** de dígitos. Este aumento de densidade é obtido pela

impressão de um conjunto de 100 pares de dígitos de 00 a 99 inclusive. Além disto permite a impressão de caracteres especiais. Ou seja o *subconjunto C* é utilizado para compactar o código de barras.

Sendo assim as funções *MSCB128A()*, *MSCB128B()* e *MSCB128C()* permitem efetuar uma mudança no conjunto dentro do símbolo (código de barras). Todos os caracteres que seguem o caractere de código (subconjunto A, B ou C) são codificados de acordo com o conjunto de caracteres correspondente, salvo quando se encontra um outro caractere do código do caractere shift, que é semelhante à tecla "shift lock" no teclado de uma máquina de escrever, porém isso é tratado internamente nas funções de geração de código de barras.

A função ***MSCB128A()*** modifica a codificação para o conjunto de caracteres A.
A função ***MSCB128B()*** modifica a codificação para o conjunto de caracteres B.
A função ***MSCB128C()*** modifica a codificação para o conjunto de caracteres C.

O caractere de mudança (shift) permite uma alteração no conjunto de caracteres para o caractere que o segue imediatamente no símbolo. Os caracteres subsequentes reverterem para o conjunto de caracteres ativo anterior ao caractere de mudança. O caractere shift opera somente entre os subconjuntos A e B. Não é possível mudar para o subconjunto C, nem mudar a partir dele. O caractere shift é semelhante à tecla "shift" no teclado de uma máquina de escrever. A função no protheus para utilização do caractere de "shift" é ***MSCB128Shift()***.

Existe também, o caractere FNC 1, usado como separador quando estão concatenados vários AIs e seus campos de dados quando os AI's são variáveis, é transmitido como caractere GS. O GS no protheus é representado pela função *MsCB128GS()*. Os AI's serão discutidos no item 3.6.2.

Lembrando ao leitor que o mesmo não deverá se preocupar com a função *MSCB128Shift()* e *MSCB128GS()*, pois as mesmas são tratadas internamente na geração do código de barras. O leitor deverá somente se preocupar em seguir alguns critérios apresentados no exemplo 02 do item 3.6.

3.6.2. IDENTIFICAÇÃO DE APLICAÇÃO (Ais)

O símbolo UCC/EAN-128 é uma simbologia extremamente flexível. Permite a representação de dados de comprimento variável e possibilita codificar várias informações em um símbolo do código de barras. Isso se denomina concatenação. Este tipo de código de barras é iniciado por um caractere "START" indicando a tabela – code "A", "B" ou "C" vistos anteriormente e que permite a identificação de 128 caracteres ASCII, seguido do caractere "FNC1", ex: "START" (Code C) "FNC1". Este caractere duplo inicial não é visível para as pessoas, mas deve estar no início das barras para que seja possível o reconhecimento e tratamento da estrutura da simbologia: os Indicadores de Aplicação, caractere separador, Dígito Verificador e Stop.

Os Ais são códigos que anunciam com exclusividade os dados que os sucedem, seu significado e seu comprimento. Os dados que aparecem após um AI podem abranger caracteres alfabéticos e/ou numéricos de qualquer comprimento, chegando em alguns

formatos até trinta caracteres. Os campos de dados têm comprimento fixo ou variável, dependendo do AI.

Os atributos de dados relacionam-se com um item: um item comercial ou de transporte, e não possuem significado caso sejam isolados. Podem ser representados no UCC/EAN-128 usando-se AIs. Existe uma faixa de AIs para medidas de itens comerciais, nos quais o peso descrito é o peso líquido, e uma faixa de AIs para medidas de unidades de transportes, denominadas unidades logísticas, nas quais o peso descrito é o peso bruto.

A tabela a seguir foi extraída da lista completa como exemplo dos AIs. Observe que um deles representa um significado – “conteúdo”, uma senha – “título de dados” para sua identificação na aplicação; e um formato” para a representação dos dados n (numéricos) ou em an (alfanuméricos), fixos ou variáveis.

A representação de AIs numa mesma linha do código de barras é possível desde que respeitadas as regras de concatenação:

- Máximo de 48 caracteres numa mesma sequência de dados (incluindo os AIs e caracter(es) separador(es) se tiver(em), mas excluindo os caracteres auxiliares e o dígito verificador);
- Máximo de 165mm do símbolo, incluindo as margens de silêncio (para uma mesma linha de código de barras);
- Uso do caractere separador “FNC1” depois de um AI de dado variável, quando concatenado com outro AI. A tabela a seguir identifica todos os Ais reconhecidos como tendo formato de dados fixos.

Os Identificadores de Aplicação que não aparecem nesta tabela deverão receber caractere separador (FNC1), quando anteceder qualquer outro AI numa concatenação. O caractere separador de dados FNC1 não aparece na representação humano-legível, mas deverá estar contido nas barras, a menos que o AI variável for o ultimo representado no código, pois neste caso o caractere “STOP” determina o final de representação.

Tabela de AI´s disponibilizada pela EAN:

AI	Título Completo	Formato
00	Código de Série de Unidade Logística	n2 +n18
01	Número Global de item Comercial	n2+n14
02	GTIN de itens comerciais contidos em uma unidade logística	n2+n14
10	Número de Batch ou de lote	n2 + an..20
11 ¹	Data de Produção (AAMMDD)	n2+n6
12 ¹	Data de Vencimento/ Pagamento (AAMMDD)	n2+n6
13 ¹	Data de Embalagem (AAMMDD)	n2+n6
15 ¹	Data de Durabilidade Mínima (AAMMDD)	n2+n6
17 ¹	Data de Durabilidade Máxima (AAMMDD)	n2+n6
20	Variante do Produto	n2+n2
21	Número de Série	n2+an20
22	Dados Secundários para Produtos Específicos do Setor de Saúde	n2+an..29
23 ²	Número de lote (Uso Transitório)	n3+n..19

240	Identificação Adicional do Produto Atribuída pelo Fabricante	n3+an..30
241	Número de Cliente	n3+an..30
2504	Número de Série Secundário	n3+an..30
AI	Título Completo	Formato
2514	Referência da Entidade de Origem	n3+an..30
30	Contagem Variável	n2+n..8
31-36 ³	Medidas Comerciais e Lógicas	n4+n6
337 ³	Quilos por metro quadrado	n4+n6
37	Contagem de itens Comerciais Contidos numa Unidade Logística	n2+n..8
390 ³	Valor a Pagar – área monetária isolada	n4+n..15
391 ³	Valor a Pagar – com o código ISO de moeda	n4+n3+n..15
392n	Valor a Pagar por um item de medida variável – área monetária isolada	n4+n..15
393n	Valor a Pagar por um item de medida variável – com o código ISO de moeda	n4+n3+n..15
400	Número de Ordem de Compra do Cliente	n3+an..30
401	Número de Consignação	n3+an..30
402	Número de Identificação do Despacho	n3+n17
403	Código de Rota	n3+an..30
410	Número de Localização Global EAN/UCC “Despachar para – Entregar para”– destino imediato	n3+n13
411	Número Global de Localização EAN/UCC “Faturar para” “ Cobrar de”	n3+n13
412	Número Global de Localização EAN/UCC “ Comprado de”	n3+n13
413	Número Global de Localização EAN/UCC “Despachar para – Entregar para “– destino final	n3+n13
414	Identificação de Local Físico Número Global EAN/UCC de Localização	n3+n13
415	Número Global de Localização EAN/UCC da Parte que fatura	n3 + n13
420	Código Postal dentro de uma única Autoridade Postal “Despachar para - Entregar para”	n3+an20
421	Código Postal com Código isso de País com 3 Dígitos “ Despachar” para - Entregar para “	n3+n3+an9
422	País de Origem de um Item Comercial – Padrão ISO 3166	n3+n3
8001	Produtos em Rolo – Largura, Comprimento, Diâmetro do Centro, Direção e Junções	n4+n14
8002	Identificador Eletrônico em Série para Telefones Celulares Móveis	n4+an..20
8003	Identificador Global de Ativo Retornável	n4+n14+an..16
8004	Identificador Global de Ativo Individual	n4+an..30
8005	Preço por Unidade de Medida	n4+n6
8006	Identificação do Componente de um Item Comercial	n4+n14+n2+n2
8007	Número de Conta de Banco Internacional	n4+an..30
8018	Número de Relação de Serviço Global	n4+n..18
8020	Número de Referência de Boleto de Pagamento	n4+an..25
8100	Código Estendido de Cupom – NSC+ Código de oferta (USA/Canadá)	n4+n1+n5
8101	Código Estendido de Cupom – NSC+ Código de oferta + final do código de oferta (USA/Canadá)	n4+n1+n5+n4
8102	Código Estendido de Cupom – NSC (USA/Canadá)	n4+n1+ n1
90***	Informação Combinada Mutuamente entre Parceiros Comerciais (Incluindo FACT DIs)	n2 +na..30
91-99	Informação interna de Empresa	n2+na..30

Tabela 01 – Tabela de AI 's

Onde:

1. Quando forem necessários apenas ano e mês, DD deve ser preenchido com "00".
2. Mais um dígito para a indicação de comprimento.
3. Estes Als compreendem 4 dígitos. Os três primeiros dígitos identificam o objetivo do AI, e o quarto dígito indica a posição do ponto decimal.

***. O título do dado "vigente" deve ser especificado pelo emissor da informação.

Exemplo do código 128 usando o padrão de programação ZPL:

```
Local cPorta := "COM1:9600,N,8,1"
```

```
MSCBPRINTER("S500-8",cPorta,,35)
```

```
MSCBBEGIN(1,6)
```

```
MSCBBOX(02,01,76,34,1)
```

```
aConteudo := {"01","07893316010411"},;
```

```
    {"10","0000970100"+MSCB128B()+"1"+MSCB128C()},;
```

```
    {"37","0004"},;
```

```
    {"21","000494"}}
```

```
MSCBSAYBAR(08,10,aConteudo,"N","MB07",10,.F.,.T.,.F.,.C",2,1,.F.)
```

```
MSCBEND()
```

```
MSCBCLOSEPRINTER()
```

Note: O conteúdo é do tipo array.

3.7. MSCBBox

Objetivo:

Imprime um box.

Sintaxe:

MSCBBOX(nX1mm, nY1mm, nX2mm, nY2mm, nEspessura, cCor)

Parâmetro	Descrição
nX1mm	Posição X1 em Milímetros
nY1mm	Posição Y1 em Milímetros
nX2mm	Posição X2 em Milímetros
nY2mm	Posição Y2 em Milímetros
[nEspessura]	Numero com a espessura em pixel
*[cCor]	String com a Cor Branca ou Preta ("W" ou "B")

Exemplo usando o padrão de programação Zebra (família ZPL):

```
Local cPorta := "LPT1"
```

```
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)
```

```
MSCBBEGIN(1,6)
```

```
MSCBBOX(02,01,76,35)
```

```
MSCBEND()
```

```
MSCBCLOSEPRINTER()
```

3.8. MSCBLineH

Objetivo:

Imprime uma linha horizontal.

Sintaxe:

MSCBLineH(nX1mm, nY1mm, nX2mm, nEspessura, cCor)

Parâmetro	Descrição
nX1mm	Posição X1 em Milímetros
nY1mm	Posição Y1 em Milímetros
nX2mm	Posição X2 em Milímetros
[nEspessura]	Numero com a espessura em pixel
*[cCor]	String com a Cor Branca ou Preta ("W" ou "B")

Exemplo usando o padrão de programação Zebra (família ZPL):

```
Local cPorta := "LPT1"
```

```
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)  
MSCBBEGIN(1,6)  
MSCBLineH(30,05,76,3)  
MSCBLineH(02,13,76,3,"B")  
MSCBEND()  
MSCBCLOSEPRINTER()
```

3.9. MSCBLineV

Objetivo:

Imprime uma linha vertical.

Sintaxe:

MSCBLineV(nX1mm, nY1mm, nY2mm, nEspessura, cCor)

Parâmetro	Descrição
nX1mm	Posição X1 em Milímetros
nY1mm	Posição Y1 em Milímetros
nY2mm	Posição X2 em Milímetros
[nEspessura]	Numero com a espessura em pixel
*[cCor]	String com a Cor Branca ou Preta ("W" ou "B")

Exemplo usando o padrão de programação Zebra (família ZPL):

```
Local cPorta := "LPT1"
```

```
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)  
MSCBBEGIN(1,6)  
MSCBLineV(30,01,13)
```

MSCBEND()
MSCBCLOSEPRINTER()

3.10. MSCBLoadGraf

Objetivo:

Carrega uma imagem para memória da impressora.

Sintaxe:

MSCBLOADGRF(cImagem)

Parâmetro	Descrição
clmagem	Nome do arquivo que será carregado, inclusive o path + nome completo + extensão

Observações:

Para o padrão Zebra, o arquivo do gráfico (cImagem) tem que ser do tipo GRF, gerado através de um PCX ou TIF no software fornecido pelo fabricante da zebra.

Para o padrão Intermec, o arquivo do gráfico (cImagem) tem que ser do tipo UDC, gerado através de um BMP ou TIF(não compactado) no software fornecido pelo fabricante da intermec.

Para o padrão Datamax, o arquivo do gráfico pode ser do tipo BMP, PCX, não sendo necessário ser convertido.

Para o padrão Eltron, o arquivo do gráfico tem que ser do tipo PCX, não sendo necessário ser convertido.

Exemplo do código 128 usando o padrão de programação ZPL:

Local cPorta := "LPT1"

```
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)  
MSCBLOADGRF("SIGA.GRF")  
MSCBBEGIN(1,6)  
MSCBGRATIC(2,3,"SIGA")  
MSCBEND()  
MSCBCLOSEPRINTER()
```

3.11. MSCBGraphic

Objetivo:

Imprime gráfico que está armazenado na memória da impressora.

Sintaxe:

MSCBGRATIC(nXmm, nYmm, cArquivo, lReverso)

Parâmetro	Descrição
NXmm	Posição X em Milímetros
NYmm	Posição Y em Milímetros
cArquivo	Nome do gráfico que foi carregado na memória da impressora (não colocar a extensão do arquivo)
*[lReverso]	Imprime em reverso quando tiver sobre um box preto

Exemplo usando o padrão de programação ZPL:

```
Local cPorta := "LPT1"  
MSCBPRINTER("S500-8",cPorta, , 40 ,f.)  
MSCBLOADGRF("SIGA.GRF")  
MSCBBEGIN(1,6)  
MSCBGRATIC(2,3,"SIGA")  
MSCBEND()  
MSCBCLOSEPRINTER()
```

3.12. MSCBChkStatus

Objetivo:

Seta ou visualiza o controle de status do sistema com a impressora.

Sintaxe:

MSCBCHKStatus(lStatus)

Parâmetro	Descrição
[lStatus]	Lógica ativa/desativa o controle. Seu valor default é .t.

Observações:

Com o status ativado, sempre que a aplicação enviar qualquer informação para a impressora, será analisado o status, caso esteja com o buffer cheio, sem papel ou sem ribbon, o sistema aguardará até que os itens anteriores estejam solucionados. Caso haja

uma incompatibilidade com o sistema operacional ou uma configuração inadequada, o mesmo poderá travar por não conseguir efetuar uma comunicação correta, caso isto ocorra, desative esta opção.

Exemplo usando o padrão de programação Zebra (família ZPL):

```
Local cPorta := "LPT1"
```

```
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)
```

```
MSCBCHKSTATUS(.t.)
```

```
MSCBBEGIN(1,6)
```

```
MSCBLineV(30,01,13)
```

```
MSCBEND()
```

```
MSCBCLOSEPRINTER()
```

3.13. MSCBInfoEti

Objetivo:

Grava informações para gerenciamento do MSCBSpool.

Sintaxe:

```
MSCBInfoEti(cDescr,cFiltro)
```

Parâmetro	Descrição
cDescr	Descrição que gerará informação no MSCBSpool para identificar a etiqueta.
cFiltro	É utilizado para realizar filtros no MSCBSpool. Vide documentação MSCBSpool.

Exemplo usando o padrão de programação Zebra (família ZPL):

```
Local nX
```

```
Local cPorta := "LPT1"
```

```
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)
```

```
For nx:=1 to 3
```

```
MSCBINFOETI("Exemplo 1","MODELO 1")
```

```
MSCBBEGIN(1,6)
```

```
MSCBSAY(05,05,"IMPRESSORA ZEBRA S500-8","N", "0", "020,030")
```

```
MSCBEND()
```

```
Next
```

```
MSCBCLOSEPRINTER()
```

3.14. MSCBWrite

Objetivo:

Permite enviar para porta uma linha de programação nativa da Impressora.

Sintaxe:

MSCBWrite(cConteudo)

Parâmetro	Descrição
cConteudo	Linha de programação nativa da impressora.

Exemplo usando o padrão de programação Zebra (família ZPL):

```
Local nX
Local cPorta := "COM1:9600,N,8,1"

MSCBPRINTER("S500-8",cPorta, , 40 ,f.)
MSCBBEGIN(1,6)
MSCBWrite("^FXPARAMETROS GERAIS ^FS")
MSCBWrite("^LL120")
MSCBWrite("^LH30,30")
MSCBWrite("^PRA")
MSCBWrite("^PQ10,2,,N")
MSCBWrite("^FXCONTEUDO DA ETIQUETA^FS")
MSCBWrite("^F01,1^GB250,90,10^FS")
MSCBWrite("^F035,40^ADN,18,10^FDMicrosiga Software S/A^FS")
MSCBEND()
MSCBCLOSEPRINTER()
```

3.15. MSCBVar

Objetivo:

Cria variáveis para serem utilizadas nos form's (imagem) da etiqueta. Sendo assim podemos criar para cada um dos campos (código de barras e textos) uma variável correspondente. Normalmente este recurso é utilizado somente em layout's variáveis, ou seja, em campos que são alterados constantemente.

O uso de variáveis permite ao programador criar um layout padrão (form) o qual será enviado para a impressora térmica, e depois o mesmo somente enviará as alterações das variáveis, diminuindo a quantidade de dados que serão carregados na impressora. Isto fará com que a impressora use uma quantidade menor de memória, podendo com isto ter uma diminuição de erros provenientes do buffer de impressão e um aumento de performance (velocidade) na impressão.

Sintaxe:

MSCBVar(cVar,cDados)

Parâmetro	Descrição
cVar	Nome da variável. Lembrando que o mesmo deverá ser incremental e sempre inicializar com "@".(Exemplo: "@1")
cdados	Conteúdo da variável

Exemplo usando o padrão de programação Zebra (família ZPL):

```

Local nX
Local cPorta := "COM1:9600,N,8,1"

MSCBPRINTER("S500-8",cPorta,      ,      ,.f. )
MSCBLOADGRF("SIGA.GRF")
MSCBBEGIN(2,6,nil,.T.)
  MSCBBOX(02,01,76,35)
  MSCBLineH(30,05,76,3)
  MSCBLineH(02,13,76,3,"B")
  MSCBLineH(02,20,76,3,"B")
  MSCBLineV(30,01,13)
  MSCBGRATIC(2,3,"SIGA")
  MSCBSAY(33,02,'PRODUTO',"N","0","025,035")
  MSCBSAY(33,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09,"@1","N","0","032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N","0","020,030")
  MSCBSAYBAR(23,20,"@2" ,"N" ,"C" ,8.36 ,.F. ,.T. ,.F. ,      ,2      ,1      ,.F.,.f., "1",.T.)
MSCBEND()

For nx:=1 to 3
  MSCBINFOETI("PRODUTO","MODELO 1")
  MSCBBEGIN(10,,,F.)
  MSCBVAR("@1",StrZero(nX,10))
  MSCBVAR("@2",StrZero(nX,10))
  MSCBEND()
Next

MSCBCLOSEPRINTER()

```

Observação: Note no exemplo acima, que o primeiro bloco inicializado com MSCBBEGIN e finalizado com MSCBEND constroi o layout da etiqueta (form). Já o loop "For" imprime 3 etiquetas utilizando as variáveis que atualizam o código do produto e o código de barras, devinido com a variável "@1" e "@2" respectivamente.

4. Exemplos

4.1. Padrão Datamax

4.1.1. Exemplo 01

```
User Function ExDPL1()
Local nX
Local cPorta := "COM1:9600,n,8,2"

MSCBPRINTER("ALLEGRO",cPorta, , ,.f. )
MSCBLOADGRF("SIGA.BMP")
For nx:=1 to 3
  MSCBINFOETI("Exemplo 1","MODELO 1")
  MSCBBEGIN(1,4)
  MSCBBOX(02,01,76,34,1)
  MSCBLineH(30,30,76,1)
  MSCBLineH(02,23,76,1)
  MSCBLineH(02,15,76,1)
  MSCBLineV(30,23,34,1)
  MSCBGRAFIC(2,26,"SIGA")
  MSCBSAY(33,31,'PRODUTO',"N","2","01,01")
  MSCBSAY(33,27,"CODIGO","N","2","01,01")
  MSCBSAY(33,24,Strzero(nX,10), "N", "2", "01,01")
  MSCBSAY(05,20,"DESCRICAO","N","2","01,01")
  MSCBSAY(05,16,"IMPRESSORA ALLEGRO 2 BR","N", "2", "01,01")
  MSCBSAYBAR(22,03,Strzero(nX,10), , ,8.36,.F.,.T.,.F.,,3,2)
  MSCBEND()
Next
MSCBCLOSEPRINTER()
Return
```

4.1.2. Exemplo 02

```
User Function EXDPL2()
Local cPorta
cPorta := "COM1:9600,N,8,2"
MSCBPRINTER("ALLEGRO",cPorta, ,35 ,.f.)
MSCBBEGIN()
MSCBBOX(02,01,100,30)
aConteudo := {"01","07893316010411"},;
           {"10","0000970100"+MSCB128B()+"1"+MSCB128C()},;
           {"37","0004"},;
           {"21","000494"}
MSCBSAYBAR(08,10,aConteudo,"N","MB07",10,.F.,.T.,nil,'C',3,2,.f.)
MSCBEND()
MSCBCLOSEPRINTER()
Return
```


4.1.3. Exemplo 03

```
User Function ExDPL3()
Local nX
Local cPorta := "COM1:9600,n,8,2"
MSCBPRINTER("ALLEGRO",cPorta, , ,f. )
MSCBLOADGRF("SIGA.BMP")
MSCBBEGIN(1,4,,t.)
MSCBBOX(02,01,76,34,1)
MSCBLineH(30,30,76,1)
MSCBLineH(02,23,76,1)
MSCBLineH(02,15,76,1)
MSCBLineV(30,23,34,1)
MSCBGRAFIC(2,26,"SIGA")
MSCBSAY(33,31,'PRODUTO',"N","2","01,01")
MSCBSAY(33,27,"CODIGO","N","2","01,01")
MSCBSAY(33,24,"@1","N","2","01,01")
MSCBSAY(05,20,"DESCRICAO","N","2","01,01")
MSCBSAY(05,16,"IMPRESSORA ALLEGRO 2 BR","N","2","01,01")
MSCBSAYBAR(22,03,"@2","N","MB07",8.36,.F.,.T.,.F.,3,2,.F.)
MSCBEND()

For nx:=1 to 10
  MSCBINFOETI("PRODUTO","MODELO 1")
  MSCBBEGIN(10,,.F.)
  MSCBVAR("@1",StrZero(nX,10))
  MSCBVAR("@2",StrZero(nX,10))
  MSCBEND()
Next

MSCBBEGIN(1,4,NIL,NIL)
MSCBBOX(02,01,76,34,1)
MSCBLineH(30,30,76,1)
MSCBLineH(02,23,76,1)
MSCBLineH(02,15,76,1)
MSCBLineV(30,23,34,1)
MSCBGRAFIC(2,26,"SIGA")
MSCBSAY(33,31,'TESTE FINAL',"N","2","01,01")
MSCBSAY(33,27,"CODIGO","N","2","01,01")
MSCBSAY(33,24,"000001","N","2","01,01")
MSCBSAY(05,20,"DESCRICAO","N","2","01,01")
MSCBSAY(05,16,"IMPRESSORA ALLEGRO 2 BR","N","2","01,01")
MSCBSAYBAR(22,03,"000001","N","MB07",8.36,.F.,.T.,.F.,3,2,.F.)
MSCBEND()

MSCBCLOSEPRINTER()
Return
```

4.2. Padrão Eltron

4.2.1. Exemplo 01

```
User Function EXEPL1()  
Local nX  
Local cPorta:="LPT1"  
MSCBPRINTER("ELTRON",cPorta,,,F.)  
MSCBLOADGRF("SIGA.PCX")  
For nx:=1 to 3  
  MSCBINFOETI("Exemplo 1","MODELO 1")  
  MSCBBEGIN(1,6)  
  MSCBGRAFIC(04,02,"SIGA")  
  MSCBBOX(05,01,76,30,2)  
  MSCBLineH(30,06,71,2)  
  MSCBLineH(05,12,71,2)  
  MSCBLineH(05,18,71,2)  
  MSCBLineV(30,1,12,2)  
  MSCBSAY(33,02,'PRODUTO',"N","2","1,2")  
  MSCBSAY(33,07,"CODIGO", "N", "1", "1,1")  
  MSCBSAY(33,09,Strzero(nX,10), "N","1","1,2")  
  MSCBSAY(07,13,"DESCRICAO","N","1","1,1")  
  MSCBSAY(07,15,"IMPRESSORA ELTRON","N","1","1,2")  
  MSCBSAYBAR(28,19,Strzero(nX,10),'N','MB07',06,.f.,.t.,,2,2)  
  MSCBEND()  
Next  
MSCBCLOSEPRINTER()  
Return
```

4.2.2. Exemplo 02

```
User Function EXEPL2()  
Local cPorta := "LPT1"  
MSCBPRINTER("ELTRON",cPorta,,,F.)  
MSCBBEGIN(1,6)  
MSCBBOX(05,03,105,30,2)  
aConteudo := {"01","07893316010411"},;  
             {"10","0000970100"+MSCB128B()+"1"+MSCB128C()},;  
             {"37","0004"},;  
             {"21","000494"}  
MSCBSAYBAR(08,08,aConteudo,'N','MB07',08,..T.,NIL,"C",2,2,,,)   
MSCBEND()  
MSCBCLOSEPRINTER()  
Return
```

4.2.3. Exemplo 03

```
User Function EXEPL3()
Local cPorta := "LPT1"
Local nX

MSCBPRINTER("ELTRON",cPorta,,,F.)
MSCBLOADGRF("SIGA.PCX")
For nx:=1 to 1
  MSCBINFOETI("ETI TESTE","PEQ")
  MSCBBEGIN(3,6,,,t.)
  MSCBGRAFIC(04,02,"SIGA")
  MSCBBOX(05,01,76,30,2)
  MSCBLineH(30,06,71,2)
  MSCBLineH(05,12,71,2)
  MSCBLineH(05,18,71,2)
  MSCBLineV(30,1,12,2) //Monta Linha Vertical
  MSCBSAY(33,02,'PRODUTO',"N","2","1,2")
  MSCBSAY(33,07,"CODIGO", "N", "1", "1,1")
  MSCBSAY(33,09,"0000001", "N", "1", "1,2",.T. ,.t. ,'+4')
  MSCBSAY(07,13,"@1","N","1","1,1")
  MSCBSAY(07,15,"IMPRESSORA ELTRON TLP2742","N", "1", "1,2")
  MSCBSAYBAR(28,19,"@2" , 'N' , 'MB07' ,06 ,.f. ,.t. , , ,2 ,2)
  MSCBEND()
Next
For nx:=1 to 4
  MSCBINFOETI("ETI TESTE","PEQ")
  MSCBBEGIN(10,,,F.)
  MSCBVAR("@1",StrZero(nX,10))
  MSCBVAR("@2",StrZero(nX,10))
  MSCBEND()
Next

MSCBINFOETI("ETI TESTE","PEQ")
MSCBBEGIN(3,6)
MSCBGRAFIC(04,02,"SIGA")
MSCBBOX(05,01,76,30,2)
MSCBLineH(30,06,71,2)
MSCBLineH(05,12,71,2)
MSCBLineH(05,18,71,2)
MSCBLineV(30,1,12,2) //Monta Linha Vertical
MSCBSAY(33,02,'PRODUTO',"N","2","1,2")
MSCBSAY(33,07,"CODIGO", "N", "1", "1,1")
MSCBSAY(33,09,"0000001", "N", "1", "1,2",.T. ,.F. ,'+4')
MSCBSAY(07,13,"teste da folha de rosto","N","1","1,1")
MSCBSAY(07,15,"IMPRESSORA ELTRON TLP2742","N", "1", "1,2")
MSCBSAYBAR(28,19,"0000001" , 'N' , 'MB07' ,06 ,.f. ,.t. , , "C" ,2 ,2 ,.t. )
MSCBEND()

MSCBCLOSEPRINTER()
Return
```

4.3. Padrão Intermec

4.3.1. Exemplo 01

```
User Function ExIPL1()
Local nX
Local cPorta := "COM1:19200,E,7,1"

MSCBPRINTER("3600-8",cPorta,,40,.f.)
MSCBLOADGRF("SIGA.UDC")
For nx:=1 to 3
  MSCBINFOETI("Exemplo 1","MODELO 1")
  MSCBBEGIN(1,5,,NIL)
    MSCBBOX(01,02,34,76,1)
    MSCBLineV(30,30,76,1)
    MSCBLineV(23,02,76,1)
    MSCBLineV(15,02,76,1)
    MSCBLineH(23,30,34,1)
    MSCBGRAFIC(34,02,"SIGA",.T.)
    MSCBSAY(32,33,'PRODUTO',"R", "2", "01,01")
    MSCBSAY(29,33,"CODIGO","R", "2", "01,01")
    MSCBSAY(26,33,StrZero(nX,10), "R", "2", "01,01")
    MSCBSAY(22,05,"DESCRICAO", "R", "2", "01,01")
    MSCBSAY(19,05,"IMPRESSORA INTERMEC", "R", "2", "01,01")
    MSCBSAYBAR(12,22,StrZero(nX,10),"R", "MB07",8.36,.F.,.T.,.F.,2,2)
  MSCBEND()
Next
MSCBCLOSEPRINTER()
Return
```

4.3.2. Exemplo 02

```
User Function EXIPL2()
Local cPorta := "COM1:19200,N,8,1"
MSCBPRINTER("3600-8",cPorta,,35,.f.)
MSCBBEGIN()
MSCBBOX(02,01,30,100,3)
aConteudo := {"01","07893316010411"},;
             {"10","0000970100"+MSCB128B()+"1"+MSCB128C()},;
             {"37","0004"},;
             {"21","000494"}
MSCBSAYBAR(20,08,aConteudo,"R", "MB07",10,.F.,.T.,nil,'C',2,1,.f.)
MSCBEND()
MSCBCLOSEPRINTER()
Return
```

4.3.3. Exemplo 03

```
User Function ExIPL3()
Local nX
Local cPorta

cPorta := "COM1:19200,E,7,1"
MSCBPRINTER("I3600",cPorta,NIL ,40 ,.f. )
MSCBLOADGRF("SIGA.UDC")

MSCBBEGIN(1,5,,.T.)
  MSCBINFOETI("PRODUTO","MODELO 1")
  MSCBBOX(01,02,34,76,1)
  MSCBLineV(30,30,76,1)
  MSCBLineV(23,02,76,1)
  MSCBLineV(15,02,76,1)
  MSCBLineH(23,30,34,1)
  MSCBGRAFIC(34,02,"SIGA")
  MSCBSAY(32,33,'PRODUTO',"R","2","01,01")
  MSCBSAY(29,33,"CODIGO","R","2","01,01")
  MSCBSAY(26,33,"@1","R","2","01,01")
  MSCBSAY(22,05,"DESCRICAO","R","2","01,01")
  MSCBSAY(19,05,"PARAFUSO 1/4 POLEGADAS","R","2","01,01")
  MSCBSAYBAR(12,22,"@2","R","MB07",8.36,.F.,.T.,.F.,,2,2)
MSCBEND()

For nx:=1 to 10
  MSCBBEGIN(,,.F.)
  MSCBINFOETI("PRODUTO","MODELO 1")
  MSCBVAR("@1",StrZero(nX,10))
  MSCBVAR("@2","999"+StrZero(nX,7))
  MSCBEND()
Next

MSCBINFOETI("PRODUTO","MODELO 1")
MSCBBEGIN(2,6,,NIL)
  MSCBBOX(01,02,34,76,1)
  MSCBLineV(30,30,76,1)
  MSCBLineV(23,02,76,1)
  MSCBLineV(15,02,76,1)
  MSCBLineH(23,30,34,1)
  MSCBGRAFIC(2,26,"SIGA",.T.)
  MSCBSAY(32,33,'PRODUTO',"R","2","01,01")
  MSCBSAY(29,33,"CODIGO","R","2","01,01")
  MSCBSAY(26,33,"000010","R","2","01,01",.t.,,2)
  MSCBSAY(22,05,"DESCRICAO","R","2","01,01")
  MSCBSAY(19,05,"PARAFUSO 1/4 POLEGADAS","R","2","01,01")
  MSCBSAYBAR(12,22,"000010","R","MB07",8.36,.F.,.T.,.F.,,2,2,.F.)
MSCBEND()
MSCBCLOSEPRINTER()
Return
```

4.4. Padrão Zebra

4.4.1. Exemplo 01

```
User Function EXZPL1()
Local nX
Local cPorta
cPorta := "COM1:9600,N,8,1"
MSCBPRINTER("S500-8",cPorta, , 40 ,.f.)
MSCBLOADGRF("SIGA.GRF")
For nx:=1 to 3
  MSCBINFOETI("Exemplo 1","MODELO 1")
  MSCBBEGIN(1,6)
  MSCBBOX(02,01,76,35)
  MSCBLineH(30,05,76,3)
  MSCBLineH(02,13,76,3,"B")
  MSCBLineH(02,20,76,3,"B")
  MSCBLineV(30,01,13)
  MSCBGRATIC(2,3,"SIGA")
  MSCBSAY(33,02,'PRODUTO',"N","0","025,035")
  MSCBSAY(33,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09, Strzero(nX,10), "N", "0", "032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N", "0", "020,030")
  MSCBSAYBAR(23,22,Strzero(nX,10),"MB07","C",8.36,.F.,.T.,.F.,2,1)
  MSCBEND()
Next
MSCBCLOSEPRINTER()
Return
```

4.4.2. Exemplo 02

```
User Function EXZPL2()
Local cPorta
cPorta := "COM1:9600,N,8,1"
MSCBPRINTER("S500-8",cPorta,,35)
MSCBBEGIN(1,6)
MSCBBOX(02,01,76,34,1)
aConteudo :={"01","07893316010411"},;
           {"10","0000970100"+MSCB128B()+"1"+MSCB128C()},;
           {"37","0004"},;
           {"21","000494"}
MSCBSAYBAR(08,10,aConteudo,"N","MB07",10,.F.,.T.,.F.,"C",2,1,.F.)
MSCBEND()
MSCBCLOSEPRINTER()
Return
```

4.4.3. Exemplo 03

```
User Function EXZPL3()
Local cPorta := "COM1:9600,N,8,1"
MSCBPRINTER("S500-8",cPorta, , ,.f. )
MSCBLOADGRF("SIGA.GRF")
MSCBBEGIN(2,6,nil,.T.)
  MSCBBOX(02,01,76,35)
  MSCBLineH(30,05,76,3)
  MSCBLineH(02,13,76,3,"B")
  MSCBLineH(02,20,76,3,"B")
  MSCBLineV(30,01,13)
  MSCBGRAFIC(2,3,"SIGA")
  MSCBSAY(33,02,'PRODUTO',"N","0","025,035")
  MSCBSAY(33,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09,"@1","N","0","032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N","0","020,030")
  MSCBSAYBAR(23,20,"@2" ,"N" ,"C" ,8.36 ,.F. ,.T. ,.F. , ,2 ,1)
MSCBEND()

For nx:=1 to 3
  MSCBINFOETI("PRODUTO","MODELO 1")
  MSCBBEGIN(10,,.F.)
  MSCBVAR("@1",StrZero(nX,10))
  MSCBVAR("@2",StrZero(nX,10))
  MSCBEND()
Next

MSCBBEGIN(2,6,nil,nil)
  MSCBBOX(02,01,76,35)
  MSCBLineH(30,05,76,3)
  MSCBLineH(02,13,76,3,"B")
  MSCBLineH(02,20,76,3,"B")
  MSCBLineV(30,01,13)
  MSCBGRAFIC(2,3,"SIGA")
  MSCBSAY(33,02,'PRODUTO',"N","0","025,035")
  MSCBSAY(33,06,"CODIGO","N","A","015,008")
  MSCBSAY(33,09,"1","N","0","032,035")
  MSCBSAY(05,17,"IMPRESSORA ZEBRA S500-8","N","0","020,030")
  MSCBSAYBAR(23,20,"000000000002" ,"N" ,"C" ,8.36 ,.F. ,.T. ,.F. , ,2 ,1 )
MSCBEND()

MSCBCLOSEPRINTER()
Return
```

Anexo 01 (Catalogo de Fontes)

1. Catalogo de Fontes padrão DPL
2. Catalogo de Fontes padrão EPL
3. Catalogo de Fontes padrão IPL
4. Catalogo de Fontes padrão ZPL