



v5.05 XL

©2005-2013 :: <http://sibra.caclipperwebsite.com>
©2013-2024 :: <https://www.linguagemclipper.com.br/sibra>

.....

Desenvolvido por **Anderson Cardoso Silva**, Brasil 

Parabéns por acreditar e investir em uma solução brasileira nata, você incentivou o nosso orgulho de ser brasileiro e nossa economia, você investiu em nosso país!

A partir desta versão 5.5 XL todo o mecanismo de leitura do relatório foi otimizado utilizando apenas 1 arquivo temporário STM – Sibra Text Markup. Trouxemos o efeito zebrado e facilitamos o desenho de quadros no papel além de readequá-la para utilização com a famosa Visual Lib e compatibilidade com o LetoDBf.

A função que invoca o **SIBRA** se chama **RELATORIO()**, na Primeira Parte falaremos dos parâmetros desta função; na Segunda Parte, falaremos dos comandos de personalização de relatório do SIBRA; na Terceira parte falaremos do modo *Stand Alone* do SIBRA; na Quarta parte, falaremos sobre alguns detalhes sobre o SIBRAH (versão para xHarbour).

Ajude a evoluir o SIBRA, conte aos colegas e parceiros de atividades –Incentive a idéia! A felicidade é um bem que se multiplica ao ser dividido. **Deixe seu depoimento no site.**

A utilização do SIBRA é muito fácil e prático, bastando conhecer os parâmetros da função (Parte I) para por a idéia em execução.

Para testar o SIBRA é preciso ter internet. Você terá **9 dias** para testar o SIBRA, leia este manual e tire suas dúvidas com o Anderson pelo Whatsapp/Telegram (82) 99141-9420.

INTRODUÇÃO: Abrindo o pacote.zip.....	4
1. Tenho que usar o SIBRA.LIB e SIBRA.EXE?.....	4
2. O que tem no FUNCOES.TXT ? Para que serve?.....	4
3. Como compilar o seu sistema com o SIBRA.....	4
Compilando para o Clipper.....	5
Compilando para o xHarbour.....	5
Compilando para o Harbour.....	5
Compilando com o LETODbf.....	6
4. Como funciona?.....	6
PARTE I: PARÂMETROS DA FUNÇÃO.....	6
1a. Chamada: MARCA INICIO DO RELATORIO.....	6
1o. Parâmetro: 1.....	6
2o. Parâmetro: (0-4) Saída do relatório.....	6
4º. Parâmetro: Permissão de Usuário.....	7
5º. Parâmetro: Modo de Impressão.....	7
2a. Chamada: MARCA FIM DO RELATORIO.....	10
PARTE II: CRIANDO RELATÓRIOS NO SIBRA.....	13
Strings para tamanho de caracteres.....	14
Strings para estilo de fonte.....	14
Strings para espaçamento de linhas.....	15
Strings para mudança do tipo de letras (Font Face).....	15
TABELA DE CARACTERES: CHAR SET.....	16
Strings para mudança de cor da fonte.....	16
Strings para alinhamento do texto.....	17
Strings especiais.....	17
Função para imprimir gráficos.....	17
Função para espaçamento de linha personalizado.....	18
Função para Código de Barras.....	19
Boletos.....	20
Função para utilizar qualquer tipo de letra (Font Face).....	21
Função para utilizar qualquer tamanho de fonte.....	21
Função para utilizar efeito zebrado.....	21
ELEMENTOS GRÁFICOS BÁSICOS.....	21
[*SETPEN(<nStyle>, <nWidth>, <nColor>)].....	21
[LINE].....	22
[*LINE(<nX1>, <nY1>, <nX2>, <nY2>)].....	22
[BOX].....	22
[BOXO].....	22
[*BOX(<nTOP>, <nLEFT>, <nBOTTOM>, <nRIGHT>, [<nWidth>], [<nHeight>])].....	22
[*FILLRECT(<nX1>, <nY1>, <nX2>, <nY2>, <nRGBColor>)].....	23
[*ARC(<nX1>, <nY1>, <nX2>, <nY2>)].....	23
[*ELLIPSE(<nX1>, <nY1>, <nX2>, <nY2>)].....	23
Exibir mensagem enquanto roda relatório.....	23
PARTE III: CRIANDO PLANILHAS NO EXCEL.....	23
PADRÕES DA PLANILHA.....	24
DEFININDO UM CABEÇALHO NA PLANILHA.....	24
MESCLANDO CÉLULAS.....	25
FORMATANDO COLUNAS.....	25
COLOCANDO CONTEÚDO NA PLANILHA: COMANDO CÉLULA.....	26
COMANDOS CÉLULA DE 2 PARÂMETROS.....	26
COMANDOS CÉLULA DE 1 PARÂMETRO.....	29
PARTE IV: MODO <i>STAND ALONE</i>	30
O que é?.....	31
Sintaxe.....	31

Como funciona?.....	31
Importante.....	32
Arquivo de SCRIPT no STAND ALONE.....	32
Comandos do Script.....	32
Interpretando um exemplo de Script.....	33
Exemplos.....	33
PARTE V: PERSONALIZANDO O SIBRA.....	34
PARTE VI: THIRDY PARTY.....	35
PDFCreator.....	35
PARTE VII: TROUBLESHOOTING.....	36
Impressora Laser HP 1020.....	36
SIBRA não faz nada ou trava.....	36
1. Ausência do SIBRA.EXE.....	36
2. Ausência do interpretador de comandos.....	36
3. Propriedades do link do aplicativo.....	38
4. Permissões do usuário.....	38
5. Travou.....	39
Primeira página sai em branco.....	39
Demora para imprimir ou não imprime.....	40
Relatório não sai.....	40
Erro de programação / Run time error.....	40
Perdí as fontes e meu sistema não salva relatórios em arquivo.....	41

INTRODUÇÃO: Abrindo o pacote.zip

Ao adquirir o SIBRA, você recebe por email um arquivo chamado PACOTE.ZIP, ele deve conter os arquivos explicados a seguir, conforme o caso.

LIBs independentes

[SIBRAXH123.LIB](#): Biblioteca para xHarbour v1.2.3

[libSIBRAHB32.a](#): Biblioteca para Harbour v3.2

LIBs dependentes [do SIBRA.EXE]

[SIBRA50.LIB](#): Biblioteca para Clipper v5.01

[SIBRA52.LIB](#): Biblioteca para Clipper v5.2e

[SIBRA53.LIB](#): Biblioteca para Clipper v5.3

[SIBRA.EXE](#): Utilitário de impressão universal SIBRA necessário para Clipper ou modo *Stand*

Alone.

[MULTI.SCR](#): Exemplo de arquivo de *Script* para usar no modo **Stand Alone** (SIBRA @MULTI.SCR).

Demonstração do SIBRA: [TST1.PRG](#) com o respectivo arquivo de compilação.. Nesse PRG é exemplificado quase todos os comandos do SIBRA.

[TCHARAM.BMP](#): Imagem bitmap que é impressa no TST1.PRG.

[BOLCAI.BMP](#): Modelo de boleto da Caixa Econômica Federal.

[FUNCOES.TXT](#): Lista das funções contidas na LIB. Não pode haver funções com o mesmo nome em seu sistema.

[LAYOUT.CFG](#): Exemplo de personalização explicado na parte V deste manual.

Para testar no seu sistema, o teste para o Clipper deve ter o SIBRA.EXE na mesma pasta, enquanto que, o teste para [x]Harbour basta linkar a respectiva biblioteca no seu sistema.

1. Tenho que usar o SIBRA.LIB e SIBRA.EXE?

Não, você pode usar apenas o SIBRA.EXE para imprimir sem mudar nada nos seus relatórios, mas se limitará aos recursos que já conhece do Clipper. Essa modalidade do SIBRA, chamamos de **Stand Alone**. Saiba como funciona desse jeito na Parte III deste manual.

O Clipper não possui recursos para imprimir na USB, ele não conversa com o Windows. Os arquivos são impressos com recursos de outra linguagem de programação usada para criar o **SIBRA.EXE**. Esta outra linguagem conversa com o Windows e utiliza o **driver** da impressora instalada nele, por isso imprime em qualquer impressora!

A biblioteca (**SIBRA.LIB**) prepara o arquivo para ser devidamente enviado ao SIBRA.EXE para poder ser impresso. Quem faz o trabalho de impressão é o SIBRA.EXE. Com a LIB você tem expansão de recursos de impressão do Clipper, você poderá usar todos os recursos do SIBRA, tipo imprimir imagens, código de barras etc.

Obs.: Não tente criar sua própria LIB, pois não dará certo e não daremos suporte quanto a isso.

2. O que tem no FUNCOES.TXT ? Para que serve?

O arquivo FUNCOES.TXT lista o nome de todas as funções que existem na biblioteca SIBRA. O seu sistema não poderá conter função com nome duplicado, então mude o nome da função do seu sistema para não gerar conflitos. Você pode solicitar a mudança do nome da função em nossa LIB também!

3. Como compilar o seu sistema com o SIBRA

Para que você possa chamar a função RELATORIO() é preciso linkar a biblioteca do SIBRA com o seu sistema. Isso vai depender de qual linkeditor você está usando: RTLINK (nativo do Clipper); BLINKER; EXOSPACE; HBMAKE; HBMK2...

Vamos abordar somente o necessário aqui.

Compilando para o Clipper

Compilando com o RTLINK

O RTLINK é o linkeditor nativo do Clipper. Você pode linkeditar em "linha de comando" (command line) ou com um arquivo de script. Veja os comandos:

FILE ou **FI**: segue uma lista dos OBJetos a serem linkados

LIBRARY ou **LIB**: segue uma lista com as bibliotecas (LIB) a serem linkadas.

Digamos que você tenha um pequeno sistema com os programas: MENU.PRG; CADAstra.PRG; EDITA.PRG; CONSULTA.PRG e RELAT.PRG.

Você pode fazer o seguinte:

RTLINK FI MENU, CADAstra, EDITA, CONSULTA, RELAT.PRG **LIB SIBRA**

Ou

RTLINK FILE MENU, CADAstra, EDITA, CONSULTA, RELAT.PRG **LIBRARY SIBRA**

Ou, com um arquivo de script:

Script.Ink:

FI MENU, CADAstra, EDITA, CONSULTA, RELAT.PRG

LIB SIBRA

RTLINK @script.Ink

Com o BLINKER e o EXOSPACE também cabe a mesma coisa. Todos tem o mesmo parâmetro FILE e LIB. Coloque o nome da lib corretamente, tipo SIBRA52.LIB para Clipper v5.2e.

Em síntese:

Em um script **.LNK**: Você põe uma linha "LIB SIBRA" logo abaixo da linha que contém "FILE" ou "FI". Pode haver várias linhas FILE ou várias linhas LIB; pode linhas FILE ou LIB com vários nomes separados por vírgula, mas é claro que FILE é nome de programa e LIB é nome de LIB.

Compilando para o xHarbour

Como não vamos precisar do SIBRA.EXE para o **xHarbour** então teremos que atentar para algumas libs que o SIBRA precisa que esteja sendo compilado junto com o sistema.

O problema é que o HBMAKE usa libs padrões diferentes em cada versão do xHarbour, então pode ser que esteja faltando alguma que você precisa incluir manualmente.

Veja as seguintes libs externas que nossa biblioteca requer para serem compiladas junto com o seu sistema:

- libmisc.lib
- hbzebra.lib

Pegue o arquivo teste123.bc como exemplo.

Compilando para o Harbour

Além da LIB do SIBRA **libSIBRAHB32.a** você terá que compilar também:

- hbmisc.hbc
- hbwin.hbc
- hbhpp.hbc
- hbfoxpro.hbc
- xhb.hbc
- hbzebra.hbc
- hbtip.hbc

Pegue o arquivo **tst1hb32.hbp** como exemplo.

Compilando com o LETODBF

Use o *define* "**__LETODBF**" na compilação.

No Harbour insira uma linha com "-d__LETODBF" no arquivo .HBP

No xHarbour, chame o hbmake e deixe assim "User Defines: __XHARBOUR;__LETODBF". Deve aparecer em "HARBOURFLAGS" no arquivo .BC. Não esqueça de, no hbmake, fazer constar: "Libs de usuário: Ws2_32.lib Iphlpapi.lib".

Em nosso site tem um tutorial para instalar o LETODBF no Windows e no Linux na seção BLOGs: <http://linguagemclipper.com.br/blogs/tutoriais/database-server-letodbf-no-windows>

Importante: A pasta TEMP no LetoDBF deve ser informada sem o drive da unidade, ou seja, como o Leto roda no servidor, a pasta tem que estar lá no servidor também, não pode ser uma pasta local como "C:\TEMP", por exemplo, teria que ser "\\TEMP" (no raiz do servidor) ou "TEMP" (subdiretório dos DBFs no servidor).

4. Como funciona?

NÃO utilize comandos para imprimir do Clipper, tipo:

- SET PRINTER ON/OFF
- SET DEVICE TO PRINT/SCREEN
- SET PRINTER TO

Substitua estas linhas pela função RELATORIO()

Veja o exemplo que acompanha o SIBRA e entenda rapidamente o seu funcionamento.

Se o seu sistema salva o relatório em arquivo, você poderá utilizar o SIBRA no modo *Stand Alone* para imprimir sem mudar nenhuma linha do seu sistema. Sugerimos este modo se o seu sistema possui dezenas ou centenas de relatórios para não precisar alterar todos os PRGs.

PARTE I: PARÂMETROS DA FUNÇÃO

1a. Chamada: MARCA INICIO DO RELATORIO

RESUMO:

4 parâmetros: RELATORIO(1, N, C, C, N)

1o. Parâmetro: 1

2o. Parâmetro: (0-4) Saída do relatório

0 = Pergunta saída (1-4) ao usuário (padrão)

N(1-4) = Saída do relatório predefinida por este parâmetro

3o. Parâmetro: Nome do arquivo -Caso o segundo parâmetro seja 4.

4o. Parâmetro: Números das opções permitidas pelo usuário, exemplo: "124"

5o. Parâmetro: Opções disponíveis: 2 (Windows), 3 (LPT1), 4 (Auto) e 5 (Planilha do Excel).

1o. Parâmetro: 1

Este parâmetro é fixo, sempre será 1 na primeira chamada para determinar o início do relatório.

2o. Parâmetro: (0-4) Saída do relatório

Tipo numérico.

0 = Pergunta saída (1-4) ao usuário (padrão). Será aberto uma janela para o usuário escolher como vai querer o relatório. Na tela (1), impressora padrão (2), outra impressora (3) ou arquivo (4). Caso tenha escolhido tela e tenha instalado o **PDFCREATOR**, então será aberto o relatório **em PDF na tela** em modo gráfico como sairia na impressora.

N(1-4) = Saída do relatório predefinida por este parâmetro.

RELATORIO(1, 2) -Por exemplo, caso o parâmetro seja 2, o relatório será diretamente impresso na impressora padrão sem perguntar nada ao usuário.

RELATORIO(1,1,,,3) -Por exemplo, caso queira a saída direto na tela em texto puro.

RELATORIO(1,1,,,4) - Por exemplo, caso queira a saída direto na tela em PDF.
RELATORIO(1, 3) -Por exemplo, pergunta direto qual impressora quer imprimir.

3º. Parâmetro: Nome do arquivo

Tipo caracter.

Caso o segundo parâmetro seja 4, "Gravar em Arquivo", será perguntado ao usuário qual o nome que será dado ao arquivo de relatório gerado a não ser que seja informado um nome de arquivo aqui.

Caso o segundo parâmetro seja 3, "Imprimir em outra", será gerado um documento para a impressão com este nome. Útil caso queira gerar um PDF para enviar por e-mail.

Exemplo:

```
// CRIA UM NOME DE ARQUIVO COM NÚMERO ALEATÓRIO  
cARQUIVO := "ORCA"+ALLTRIM(STR(RAN(9999)))+".PDF"  
// DÁ NOME AO ARQUIVO E "IMPRIME EM OUTRA" (NO CASO, O PDFCREATOR)  
RELATORIO(1,3, cARQUIVO)
```

- 1) Vá em salvamento automático no PDFCREATOR
- 2) Em nome de arquivo: <REDMON_DOCNAME_FILE>

Marque "Usar este diretório para salvamento automático" e defina um diretório porque nossa linguagem trabalha apenas com o nome do documento. Não adianta colocar um caminho (path) junto com o nome do arquivo.

Por exemplo: C:\TEMP\

Daí, sabendo que a configuração do PDFCreator apontará para esta pasta [C:\TEMP](#) depois é só **copiar** o arquivo para outra pasta (o PDFCreator fica configurado para gerar o PDF e abrir então se tentar mover o arquivo dará erro). COPY FILE <cSourceFile> TO <cTargetFile>

4º. Parâmetro: Permissão de Usuário

Tipo caracter.

- a. Visualizar na tela: RELATORIO(1,,, "1")
- b. Imprimir na padrão: RELATORIO(1,,, "2")
- c. Imprimir em outra impressora: RELATORIO(1,,, "3")
- d. Gravar em arquivo: RELATORIO(1,,, "4")

"a" e "b": RELATORIO(1,,, "12")

"a" e "d": RELATORIO(1,,, "14")

Etc.

O padrão é "1234", ou seja, caso o 4º parâmetro seja nulo, o usuário terá permissão de escolher qualquer opção.

5º. Parâmetro: Modo de Impressão

Tipo numérico.

São 4 opções: 2, 3, 4 ou 5.

Opção 2 (Windows): Será aberto uma janela para escolher uma impressora dentre as que estão instaladas no Windows. A impressão será em modo gráfico e se dará intermediado pelo "driver" da impressora instalada no Windows, podendo imprimir em qualquer impressora instalada (Seja "Windows Only", USB, laser, deskjet, matricial, de FAX, de rede etc). Você pode imprimir gráficos, código de barras etc. No entanto, se for imprimir em uma impressora matricial a impressão será um pouco mais lenta (duas passadas por linha); assim, caso o relatório só possua texto, é aconselhável a opção 4 ou 3.

Opção 3 (LPT1): Será aberto uma janela para escolher uma impressora dentre dezenas de opções disponíveis. Imprime diretamente na impressora por meio da porta LPT1. Aconselhável apenas para impressoras matriciais prejudicadas com a velocidade de impressão na opção 2 assim como impressões em formulário contínuo sem a presença do comando EJECT. É impresso apenas texto em modo DRAFT, fazendo qualquer outro recurso do SIBRA ser ignorado. Não imprime nenhum tipo de gráfico. No entanto, estilos **negrito**, *itálico* e sublinhado são suportados, assim como controles de espaçamento de linhas 1/6, 1/8 e 7/72. É necessário a presença do arquivo PRINTER.DBF, fornecido no pacote. Nesta opção só é possível imprimir na porta **LPT1!**...

Opção 4 (Auto): É a opção padrão. Instale uma cópia da impressora no Windows e coloque o nome dela de acordo com o campo IMPRESS do arquivo PRINTER.DBF. Assim, o SIBRA irá pesquisar o nome da impressora no arquivo PRINTER.DBF pelo campo IMPRESS e, se encontrar, imprimirá com ela no modo **DRAFT** na porta LPT1 com apenas texto (equivalente à opção 3 com escolha automática).

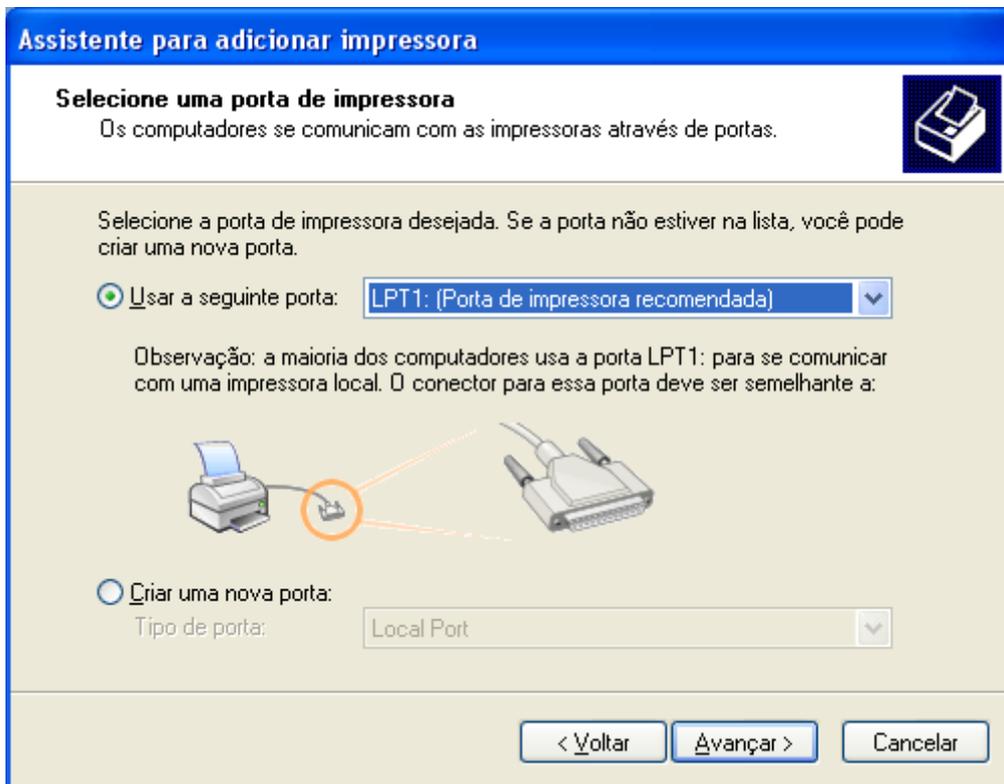
Opcão 5 (Planilha do Excel): Esta opção informa que será criado uma planilha do Excel. A forma de criar este tipo de relatório como planilha é totalmente diferente do convencional. Aprenda como na Parte III: Criando planilhas no EXCEL.

Você precisa instalar uma "nova impressora" igual a impressora matricial existente, uma "cópia". Coloque o nome dela conforme a coluna **MODELO DA IMPRESSORA** da tabela abaixo:

FABRICANTE	MODELO DA IMPRESSORA	EMULA (EQUIVALE)
Admate	100	Epson
Alps	ALQ 200/300	Epson FX
Alps	ALQ 200/300	Epson LQ
Alps	ALQ 200/300	IBM
Alps	ALQ 224e/324e	Epson
Alps	Allegro 24	Epson
Alps	P2000	Epson
Alps	P200G/2100	Epson
Amdek	5040	Diablo
Amstrad	3000/3160	Epson
Amstrad	4000	Epson
Amstrad	LQ 3500	Epson
Axiom	DX-2500/3500	Diablo
Brother	2024-L	DP Mode
Brother	BP5420	Epson FX
Brother	DM-5	Epson FX
Brother	HR-10/15/20/25	Diablo
Brother	HR-35	
Brother	HR-40	
Brother	IF-50/100/300	
Brother	M-1109	Mode I Epson
Brother	M-1509	Mode I Epson
Brother	M-1709	Mode I Epson
Brother	M-1724	Diablo Mode
Brother	M-1724	Epson LQ Mode
Brother	M-1724	IBM Proprinter
Brother	SP1000	
Brother	Twinwriter 5	Dot Matrnx only
Businessland	45LQ	Diablo
C. Itoh	C-310/15 X	Epson
C. Itoh	F10-40/1570F/	A10-20
Canon	AP-110	Diablo
Canon	AP-400/92	Diablo
Canon	AP-550	Diablo
Canon	AP-76	Diablo
Canon	LPB 8A1 / 8A2	Diablo Mode
Citizen	120-D/180-D	Epson
Citizen	120-D/180-D	IBM
Citizen	GSX-140	Epson
Citizen	MSP-10/15/20/5	Epson FX
Emilia	Emilia PS	Epson
Epson Laser	EPL-6000	HP LaserJet
Epson Laser	EPL-7000	HP LaserJet
Epson DFX	500	
Epson DX	10/20/35	
Epson EX	800/1000	Epson
Epson FX	80+/100/+	Epson
Epson FX	870/1170	Epson
Epson FX	FX-85/86/185	Epson
Epson FX	FX-850/1050	Epson
Epson FX	FX-850/1050	IBM
Epson FX	FX-86e/286e	IBM Mode
Epson JX	JX-80	
Epson LQ	All exc. 1500	
Epson LX	80/86/800/810	
Epson MX	MX-80/100	With Graftrax
Epson RX	80/100	
Epson RX	RX plus	
Epson T	750/1000	
Fujitsu	24C/24I/DL/DPL	Epson
Fujitsu	DL series	Epson
Fujitsu	DPL24, D	Diablo
Fujitsu	DX	Epson
Fujitsu	DotMax 9f/9i	
Fujitsu	SP-320/SP-320e	Diablo
Fujitsu Laser	RX series	Diablo 630
Fujitsu Laser	RX series	HP LaserJet
Fujitsu Laser	RX series	IBM Proprinter
Hewlett-Packard	LaserJet	+/II/III etc.
IBM	80 CPS Graphic	
IBM	Proprinter	
IBM Laser	4019	HP LaserJet
IBM Laser	4216	HP LaserJet
Okidata	390/391	Epson LQ
Okidata	390/391	IBM Proprinter
Okidata Laser	Laserline	HP LaserJet
Okidata Laser	Okilaser 400	HP LaserJet

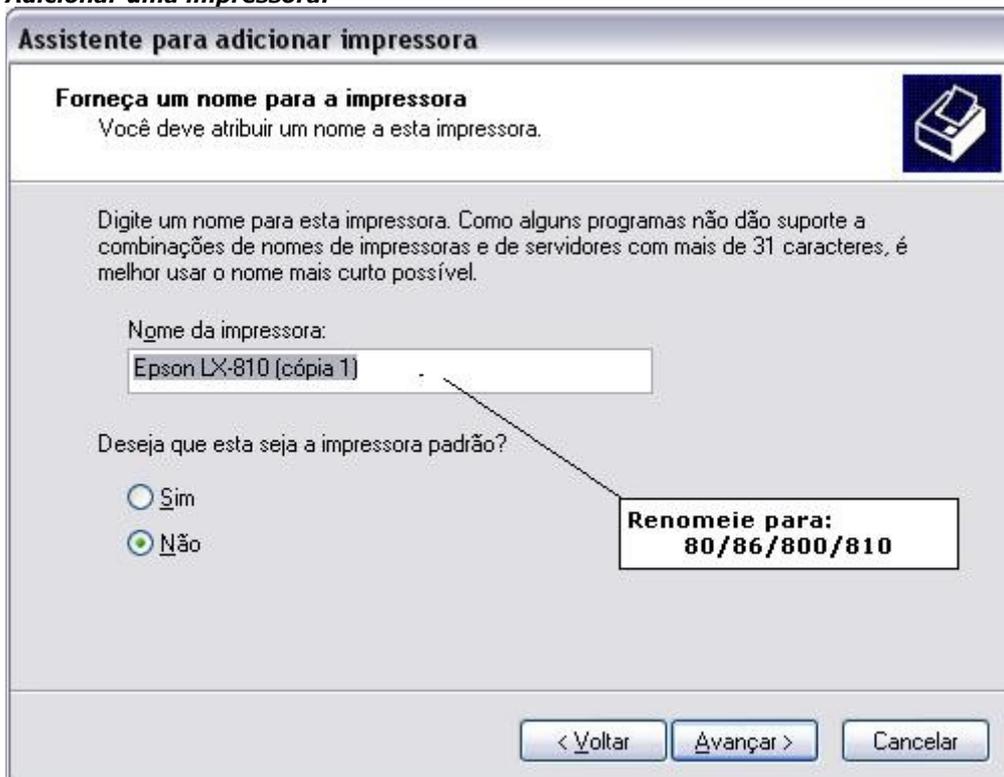
Okidata Laser	Okilaser 800	HP LaserJet
Panasonic Laser	4420 Laser	HP LaserJet
Panasonic Laser	4450 Laser	HP LaserJet
Panasonic Laser	4450i Laser	HP LaserJet
Rima Impressora	Rima XT220/300	Epson Fx 1050

Admita que você já tenha instalado a sua impressora com o nome "Epson LX-810". Agora você vai instalar de novo com outro nome. Observe as ilustrações abaixo:



Se for usar mais de uma impressora em modo DRAFT (texto) em um mesmo computador, precisa associar uma porta para cada impressora para não dar conflito.

Adicionar uma impressora:



Ficará com "duas" impressoras, assim:



Neste exemplo, quando quiser imprimir em modo DRAFT, apenas texto, é só escolher a impressora "80/86/800/810" e quando quiser imprimir gráficos, escolher a impressora "Epson LX-810".

Você verá a tela do SIBRA assim:



A partir da versão 3.8c, este parâmetro valor 4 (automático) tornou-se padrão. Você continua tendo que fazer tudo isto, mas basta fazer a 1ª chamada assim:

RELATORIO(1)

ATENÇÃO: Este procedimento é recomendado apenas para impressoras matriciais onde se queira imprimir apenas texto com velocidade em modo DRAFT.

2a. Chamada: MARCA FIM DO RELATORIO

8 parâmetros: RELATORIO(0, L, N, N, C, C, L, C, N)

1o. Parâmetro: 0

2o. Parâmetro: .T. ou .F. (.T. p/ impressão em modo paisagem)

Valor padrão: .F.

3o. Parâmetro: Nome da **impressora** quando o parâmetro 2 da primeira chamada for igual a 3 (imprimir em outra). Define a impressora a ser utilizada para imprimir o relatório. Use exatamente igual ao nome da impressora no Windows. Caso não seja informado (empty), o SIBRA abre uma janela para o usuário escolher qual impressora usar.

Ex.:

```
RELATORIO(1, 3)
```

```
RELATORIO(0, , "EPSON STYLUS COLOR 740")
```

```
RELATORIO(0,, "PDFCreator") // GERA UM PDF: RELATORIO(1,3, "ARQUIVO.PDF")
```

- 1) Vá em salvamento automático no PDFCREATOR
- 2) Em nome de arquivo: <REDMON_DOCNAME_FILE>

Marque "Usar este diretório para salvamento automático" e defina um diretório porque <REDMON_DOCNAME_PATH> do PDFCREATOR não funciona.

```
RELATORIO(0,, "Microsoft Print to PDF") // PERGUNTA O NOME DO PDF A SER GRAVADO
```

4o. Parâmetro: (10,12,17,20) Tamanho da fonte em CPI.
Valor padrão: 10

5o. Parâmetro: "nSUPERIOR, nINFERIOR, nESQUERDA, nDIREITA" Margem em caracteres.
String com parâmetros numéricos
Valor padrão: "0,0,0,0"

6o. Parâmetro: Qualidade de Impressão

Tipo numérico.

São 4 opções no SIBRA: 1, 2, 3 ou 4 (menor qualidade para maior).

Valor padrão: 1 (menor qualidade, porém mais rápido, tipo Draft).

São 2 opções no SIBRAXL (Excel): 1 (Normal) e 2 (Draft).

O modo Draft no Excel só deve ser utilizado quando só existir texto para imprimir por isso adotamos que o padrão seja Normal ao invés de Draft.

7o. Parâmetro: .T. ou .F. (.T. se for Windows XP/NT)

Valor padrão: .F. ou "NOSTART" \$ GETENV(RICOTXT)

8o. Parâmetro: Tamanho do papel personalizado.

"A3", "A4", "A5", "CARTA", "OFICIO" ou "PAG:<nPAPEL>".

Valor padrão: "A4".

A partir da versão 3.93 o SIBRA disponibiliza todos os tipos de papel para você imprimir com o parâmetro personalizado "PAG:<nPAPEL>", onde nPAPEL é um **Nº** correspondente ao **Tipo de Papel** que será utilizado conforme tabela abaixo:

<u>Nº</u>	<u>/*Tipo de Papel</u>	<u>*/</u>
1	/* Letter 8 1/2 x 11 in	*/
2	/* Letter Small 8 1/2 x 11 in	*/
3	/* Tabloid 11 x 17 in	*/
4	/* Ledger 17 x 11 in	*/
5	/* Legal 8 1/2 x 14 in	*/
6	/* Statement 5 1/2 x 8 1/2 in	*/
7	/* Executive 7 1/4 x 10 1/2 in	*/
8	/* A3 297 x 420 mm	*/
9	/* A4 210 x 297 mm	*/
10	/* A4 Small 210 x 297 mm	*/
11	/* A5 148 x 210 mm	*/
12	/* B4 (JIS) 250 x 354	*/
13	/* B5 (JIS) 182 x 257 mm	*/
14	/* Folio 8 1/2 x 13 in	*/
15	/* Quarto 215 x 275 mm	*/
16	/* 10x14 in	*/
17	/* 11x17 in	*/
18	/* Note 8 1/2 x 11 in	*/
19	/* Envelope #9 3 7/8 x 8 7/8	*/
20	/* Envelope #10 4 1/8 x 9 1/2	*/
21	/* Envelope #11 4 1/2 x 10 3/8	*/
22	/* Envelope #12 4 1/2 x 11	*/
23	/* Envelope #14 5 x 11 1/2	*/

24 /* C size sheet */
25 /* D size sheet */
26 /* E size sheet */
27 /* Envelope DL 110 x 220mm */
28 /* Envelope C5 162 x 229 mm */
29 /* Envelope C3 324 x 458 mm */
30 /* Envelope C4 229 x 324 mm */
31 /* Envelope C6 114 x 162 mm */
32 /* Envelope C65 114 x 229 mm */
33 /* Envelope B4 250 x 353 mm */
34 /* Envelope B5 176 x 250 mm */
35 /* Envelope B6 176 x 125 mm */
36 /* Envelope 110 x 230 mm */
37 /* Envelope Monarch 3.875 x 7.5 in */
38 /* 6 3/4 Envelope 3 5/8 x 6 1/2 in */
39 /* US Std Fanfold 14 7/8 x 11 in */
40 /* German Std Fanfold 8 1/2 x 12 in */
41 /* German Legal Fanfold 8 1/2 x 13 in */
42 /* B4 (ISO) 250 x 353 mm */
43 /* Japanese Postcard 100 x 148 mm */
44 /* 9 x 11 in */
45 /* 10 x 11 in */
46 /* 15 x 11 in */
47 /* Envelope Invite 220 x 220 mm */
48 /* RESERVED--DO NOT USE */
49 /* RESERVED--DO NOT USE */
50 /* Letter Extra 9 \275 x 12 in */
51 /* Legal Extra 9 \275 x 15 in */
52 /* Tabloid Extra 11.69 x 18 in */
53 /* A4 Extra 9.27 x 12.69 in */
54 /* Letter Transverse 8 \275 x 11 in */
55 /* A4 Transverse 210 x 297 mm */
56 /* Letter Extra Transverse 9\275 x 12 in */
57 /* SuperA/SuperA/A4 227 x 356 mm */
58 /* SuperB/SuperB/A3 305 x 487 mm */
59 /* Letter Plus 8.5 x 12.69 in */
60 /* A4 Plus 210 x 330 mm */
61 /* A5 Transverse 148 x 210 mm */
62 /* B5 (JIS) Transverse 182 x 257 mm */
63 /* A3 Extra 322 x 445 mm */
64 /* A5 Extra 174 x 235 mm */
65 /* B5 (ISO) Extra 201 x 276 mm */
66 /* A2 420 x 594 mm */
67 /* A3 Transverse 297 x 420 mm */
68 /* A3 Extra Transverse 322 x 445 mm */
69 /* Japanese Double Postcard 200 x 148 mm */
70 /* A6 105 x 148 mm */
71 /* Japanese Envelope Kaku #2 */
72 /* Japanese Envelope Kaku #3 */
73 /* Japanese Envelope Chou #3 */
74 /* Japanese Envelope Chou #4 */
75 /* Letter Rotated 11 x 8 1/2 11 in */
76 /* A3 Rotated 420 x 297 mm */
77 /* A4 Rotated 297 x 210 mm */
78 /* A5 Rotated 210 x 148 mm */
79 /* B4 (JIS) Rotated 364 x 257 mm */
80 /* B5 (JIS) Rotated 257 x 182 mm */
81 /* Japanese Postcard Rotated 148 x 100 mm */
82 /* Double Japanese Postcard Rotated 148 x 200 mm */
83 /* A6 Rotated 148 x 105 mm */
84 /* Japanese Envelope Kaku #2 Rotated */
85 /* Japanese Envelope Kaku #3 Rotated */
86 /* Japanese Envelope Chou #3 Rotated */
87 /* Japanese Envelope Chou #4 Rotated */
88 /* B6 (JIS) 128 x 182 mm */
89 /* B6 (JIS) Rotated 182 x 128 mm */
90 /* 12 x 11 in */
91 /* Japanese Envelope You #4 */
92 /* Japanese Envelope You #4 Rotated*/
93 /* PRC 16K 146 x 215 mm */
94 /* PRC 32K 97 x 151 mm */
95 /* PRC 32K(Big) 97 x 151 mm */
96 /* PRC Envelope #1 102 x 165 mm */
97 /* PRC Envelope #2 102 x 176 mm */
98 /* PRC Envelope #3 125 x 176 mm */
99 /* PRC Envelope #4 110 x 208 mm */
100 /* PRC Envelope #5 110 x 220 mm */

```

101 /* PRC Envelope #6 120 x 230 mm */
102 /* PRC Envelope #7 160 x 230 mm */
103 /* PRC Envelope #8 120 x 309 mm */
104 /* PRC Envelope #9 229 x 324 mm */
105 /* PRC Envelope #10 324 x 458 mm */
106 /* PRC 16K Rotated */
107 /* PRC 32K Rotated */
108 /* PRC 32K(Big) Rotated */
109 /* PRC Envelope #1 Rotated 165 x 102 mm */
110 /* PRC Envelope #2 Rotated 176 x 102 mm */
111 /* PRC Envelope #3 Rotated 176 x 125 mm */
112 /* PRC Envelope #4 Rotated 208 x 110 mm */
113 /* PRC Envelope #5 Rotated 220 x 110 mm */
114 /* PRC Envelope #6 Rotated 230 x 120 mm */
115 /* PRC Envelope #7 Rotated 230 x 160 mm */
116 /* PRC Envelope #8 Rotated 309 x 120 mm */
117 /* PRC Envelope #9 Rotated 324 x 229 mm */
118 /* PRC Envelope #10 Rotated 458 x 324 mm */

```

Por exemplo, se quiser um envelope de **165x102mm** em **modo paisagem** ("rotated") que corresponde ao número **109 (grifado na tabela acima)**, utiliza-se assim: **"PAG:109"**.

9o. Parâmetro: Número de cópias a serem impressas do relatório.

Este parâmetro só funciona se mandar direto para a uma impressora real. Se mandar para uma impressora virtual (FAX, PDF etc) não funciona. Daí você imprime pelo próprio visualizador quantas cópias quiser. O usuário também pode mudar a quantidade de cópias no menu do SIBRA.

Valor padrão: 1

* Tipos de parâmetros: N = Numérico / C = Caracter / L = Lógico / A = Array

Os parâmetros em cinza são do Texto Rico que não possuem correspondente no SIBRA, portanto, deixe-os vazios (empty). O SIBRA foi feito para ser compatível com o TEXTO RICO, permitindo sua fácil migração de um para o outro.

PARTE II: CRIANDO RELATÓRIOS NO SIBRA

O SIBRA permite imprimir com estilos de fonte, gráficos e tamanhos de caracteres diferentes, em 5, 6, 10, 12, 17 ou 20CPI, controle de espaçamento de linhas e código de barras.

Estes comandos devem ser usados como se fosse uma *string* no seu relatório.

Exemplo:

```

RELATORIO (1)

LL = 0
lPAISAGEM := .F.
@ LL++,00 SAY "[*BMP('SIBRA.BMP', {200,0,2000,193})]"
LL += 5
@ LL++,00 SAY "\10cpi TESTE COM 10 CPI"
@ LL++,00 SAY "1234567890123456789012345678901234567890123456789012345"
@ LL++,00 SAY "\12cpi TESTE COM 12 CPI"
@ LL++,00 SAY "1234567890123456789012345678901234567890123456789012345"
@ LL++,00 SAY "\17cpi TESTE COM 17 CPI"
@ LL++,00 SAY "1234567890123456789012345678901234567890123456789012345"
@ LL++,00 SAY "\20cpi TESTE COM 20 CPI"
@ LL++,00 SAY "1234567890123456789012345678901234567890123456789012345"

LL++
cTXT = "TESTE DE ESTILOS DE FONTE"
@ LL++, (80-LEN(cTXT))/2 SAY "[10CPI] [FS] [B] [I]"+cTXT+" [\B] [\I] [\FS]"
LL++
@ LL++,00 SAY "TESTE [FS] [B] NEGRITO [\B] [\FS] NESTA LINHA"
@ LL++,00 SAY "TESTE [FS] [i] ITALICO [\i] [\FS] NESTA LINHA"
@ LL++,00 SAY "TESTE [FS] [U] SUBLINHADO [\U] [\FS] NESTA LINHA"
@ LL++,00 SAY "[*BMP('TCHARAM.BMP', {1200,1500,300,179})]"

```

Strings para tamanho de caracteres

O SIBRA possui os tamanhos 10, 12, 17 e 20 CPI. As strings para ativar estes tamanhos são:

```

"[5CPI]" = 5 CPI
"[6CPI]" = 6 CPI
"[10CPI]" = 10 CPI
"[12CPI]" = 12 CPI
"[17CPI]" = 17 CPI
"[20CPI]" = 20 CPI

```

OBS.: A formatação opcional "\17cpi ", por exemplo, continua valendo.

Uma vez alterado o tamanho, ele permanecerá até o fim do relatório caso não haja nova alteração de tamanho informada. Informe as strings sempre em letras maiúsculas.

Strings para estilo de fonte

O SIBRA permite alterar o estilo de fonte para NEGRITO, ITÁLICO e SUBLINHADO e suas combinações. Ao ativar um estilo, ele permanecerá até a linha em que o mesmo for desativado. Veja as strings disponíveis abaixo:

Ativando estilo

```

"[B]" = negrito (bold)
"[I]" = itálico (italic)
"[U]" = sublinhado (underline)
"[BI]" = negrito + itálico
"[BU]" = negrito + sublinhado
"[IU]" = itálico + sublinhado
"[BIU]" = negrito + itálico + sublinhado
"[SOB]" = Sobrescrito
"[SUB]" = Subscrito

```

Desativando estilo

Basta colocar a **mesma string** antecedida pelo caracter "\", exemplos:

```

"\[B]" = negrito (bold)
"\[I]" = itálico (italic)
"\[U]" = sublinhado (underline)

```

Exemplo: @ LL++,00 SAY "TESTE [B]NEGRITO[\B] NESTA LINHA"

Resultará: TESTE NEGRITO NESTA LINHA

Desativação automática de estilo (>versão 4.01)

Quando o SIBRA encontra um outro comando de estilo na mesma linha, ele desativa o anterior automaticamente. Portanto, utilize o comando de estilo apropriado para o caso que desejar.

Exemplo: @ LL++,00 SAY "TESTE [B]NEGRITO [I]ITALICO [BI]AMBOS"

Resultará: TESTE NEGRITO ITALICO AMBOS

(Mudou desde a versão 3.7)

Sempre que quiser informar alteração de estilo de fonte, deve-se usar a string “[FS]” (abreviação de “Font Style”) para marcar o início do texto estiloso e a string “[\FS]” para marcar o fim do texto estiloso. Depois usa-se as strings relativas ao estilo que você quer ativar depois de “[FS]” e desativar antes de “[\FS]”.

Ativando estilo

“[FS]” seguido de uma ou mais opções abaixo:

“[B]” = negrito (bold)

“[I]” = itálico (italic)

“[U]” = sublinhado (underline)

Desativando estilo

“[\FS]” seguido de uma ou mais opções abaixo:

“[\B]” = negrito (bold)

“[\I]” = itálico (italic)

“[\U]” = sublinhado (underline)

Exemplo: ~~e LL+,00 SAY "TESTE [FS][B]NEGRITO[\B][\FS] NESTA LINHA"~~

Strings para espaçamento de linhas

O espaçamento de linha é essencial para imprimir em Notas Fiscais ou outros formulários pré-impresos.

Enquanto no Clipper com os PCL da impressora isso era muito limitado e, às vezes, era necessário horas de trabalho para encaixar as informações nos campos do papel; com o SIBRA este trabalho ficou muito mais prático e o que era impossível foi alcançado, pois podemos utilizar qualquer espaçamento possível.

O SIBRA possui controle de espaçamento de linhas pré-definido de 4 a 9, mas, caso precise de um controle mais preciso, use a string especial [**LSn*].

“[LS4]” = 1/4"

“[LS5]” = 1/5"

“[LS6]” = 1/6"

“[LS7]” = 1/7"

“[LS8]” = 1/8"

“[LS9]” = 1/9"

Strings para mudança do tipo de letras (Font Face)

O SIBRA possui 7 tipos de fonte, a **Courier New** é a padrão. As strings disponíveis são:

[COURIER] = Courier New

[TIMES] = Times New Roman

[LUCIDA] = Lucida Console

[VERDANA] = Verdana

[ARIAL] = Arial

[WINGDINGS] =  (símbolos)

[WEBDINGS] =  (símbolos)

OBS.: Em MODO TEXTO, só a fonte [TIMES] funciona, enquanto que, o [COURIER] equivale ao DRAFT.

TABELA DE CARACTERES: CHAR SET

O SIBRA trabalha com o **CHAR SET = 255** que equivale ao OEM de acordo com a tela console do MS-DOS, portanto o padrão não é 1, mas 255.

Se precisar alterar o charset padrão defina a **variável de ambiente STDCHARSET** com um dos valores da tabela abaixo, *exemplo*:

```
SET STDCHARSET=0
```

Na tabela foi destacado de amarelo as opções mais prováveis de utilização:

Constante	Valor
ANSI_CHARSET	0
DEFAULT_CHARSET *)	1
SYMBOL_CHARSET	2
SHIFTJIS_CHARSET	128
HANGEUL_CHARSET	129
HANGUL_CHARSET	129
GB2312_CHARSET	134
CHINESEBIG5_CHARSET	136
OEM_CHARSET	255
JOHAB_CHARSET	130
HEBREW_CHARSET	177
ARABIC_CHARSET	178
GREEK_CHARSET	161
TURKISH_CHARSET	162
VIETNAMESE_CHARSET	163
THAI_CHARSET	222
EASTEUROPE_CHARSET	238
RUSSIAN_CHARSET	204

*) default

Lembrando que:

- MODO CONSOLE = OEM = 255;
- MODO GRÁFICO = ANSI = 0;
- Fonte de símbolos = SYMBOL CHARSET = 2 (Wingdings, Webdings etc).

Strings para mudança de cor da fonte

O SIBRA possui 5 cores diferentes para as letras e para sua maior comodidade, é representado da mesma forma que os parâmetros de SETCOLOR() do Clipper. Veja:

```
[COLOR:N] = Preto  
[COLOR:B] = Azul  
[COLOR:G] = Verde  
[COLOR:BG] = Ciano  
[COLOR:R] = Vermelho  
[COLOR:RB] = Magenta  
[COLOR:GR] = Marrom  
[COLOR:GR+] = Amarelo
```

[COLOR:W] = **Branco**

Strings para alinhamento do texto

Agora ficou fácil alinhar o texto no modo gráfico com o SIBRA. Veja:

[C] = **Centralizado**
[R] = **Alinhado à direita**
[L] = **Alinhado à esquerda**

Strings especiais

Strings especiais começam sempre por “[*” e terminam com “]”, elas chamam uma função auxiliar do SIBRA específica para se obter o resultado desejado.

As strings especiais são usadas como se você fosse imprimir esta string, mas a linha não será impressa; pois é uma ordem de comando ao SIBRA e não um informação do relatório.

Os parâmetros das funções contidas nas strings especiais são sempre strings, mas devem ter formatação de um parâmetro do tipo informado em cada caso. Observe os exemplos para maior elucidação.

Os parâmetros do tipo caracter devem se apresentar em aspas simples (‘’) para não conflitar com as aspas duplas (“”).

Função para imprimir gráficos

BMP(cARQUIVO, aINFO, nZOOM, Lmm)

Onde:

cARQUIVO = nome do arquivo bitmap. Tipo caracter.

aINFO = informações de posição e tamanho do bitmap. Tipo array.

nZOOM = aumenta ou diminui o tamanho da figura preservando sua proporção de tamanho (“aspect ratio”) [Opcional]. Tipo numérico. *Utilizado para compatibilidade com versões anteriores a 4.2 XL.*

Lmm = Parâmetro criado para não criar incompatibilidade com os clientes que vinham utilizando versões anteriores do SIBRA 4.2 XL. Tipo lógico (*boolean*). Utilize F. (*default*) para compatibilidade com versões anteriores do SIBRA 4.2 XL, enquanto que, T. a partir do SIBRA 4.2 XL para parametrizar em milímetros, conforme explicações a seguir.

aINFO é um array de 4 posições, onde:

Caso Lmm= <u>F.</u> (compatibilidade)	Caso Lmm= <u>T.</u> (recomendado)
aINFO[1] = posição horizontal (X pos) em <i>pixels</i> da margem esquerda onde será impresso o gráfico. Varia de acordo com a resolução (DPI*) de cada impressora.	aINFO[1] = posição horizontal da margem esquerda em milímetros onde será impresso a imagem. O SIBRA assume a posição PCol() por padrão (default).
aINFO[2] = posição vertical (Y pos) em <i>pixels</i> da margem do cabeçalho onde será impresso o gráfico. Varia de acordo com a resolução (DPI) de cada impressora.	aINFO[2] = posição vertical da margem do cabeçalho em milímetros onde será impresso a imagem. O SIBRA assume a posição PRow() por padrão (default).
aINFO[3] = tamanho horizontal (largura) em <i>pixels</i> do gráfico.	aINFO[3] = tamanho horizontal (largura) do gráfico em milímetros .

aINFO[4] = tamanho vertical (altura) em <i>pixels</i> do gráfico.	aINFO[4] = tamanho vertical (altura) do gráfico em milímetros .
--	---

*DPI = Dots per inch (Pontos por polegada). Pontos é o mesmo que *pixels*.

Observação: O posicionamento no papel em milímetros da margem superior e esquerda assim como o tamanho do gráfico em milímetros foi implantado a partir do SIBRA 4.2 XL. Anteriormente a posição era em *pixels* e ambos dependiam da resolução de cada impressora, tornando tanto a posição quanto o tamanho do gráfico diferente em impressoras com resoluções (DPI) diferentes. Agora, a posição e o tamanho dos gráficos não mudam em impressoras com resolução (DPI) diferentes entre si.

Exemplo 1:

```
@ LL++,00 SAY "[*BMP('TCHARAM.BMP', {10,5,30,18},,.T.)]"
```

Você pode esticar ou encolher o gráfico com os elementos 3 e 4 do array.

O nome do arquivo tem que estar com **aspas simples** (para não conflitar com as aspas duplas da linha). No exemplo acima, será impresso a imagem com 30mm de largura e 18mm de altura posicionada no eixo Y a 10mm do topo da página e no eixo X a 5mm da esquerda da página.

No exemplo abaixo, a imagem é impressa com o dobro do tamanho da imagem original (Zoom 2x) no eixo de 10px da margem superior e 20px da margem esquerda.

Exemplo 2:

```
@ LL++,00 SAY "[*BMP('TCHARAM.BMP', {10,20,300,179}, 2)]"
```

Exemplo 3:

```
@ LL++,00 SAY "[*BMP('TCHARAM.BMP', {,,30,18},,.T.)]"
```

Caso as posições no gráfico sejam omitidas, o SIBRA posiciona o gráfico automaticamente na posição equivalente a **PROW()** e **PCOL()**.

Você pode esticar ou encolher o gráfico com os elementos 3 e 4 do array. Para aumentar ou diminuir a imagem sem alterar as proporções da imagem original use o parâmetro **nZOOM**.

Função para espaçamento de linha personalizado

Você já tem strings de espaçamento de 4 a 9, mas se quiser um controle mais preciso, com decimais, use esta função especial.

LS(nES)

Onde:

nES = Tamanho de espaçamento personalizado, pode conter decimais.

Tipo numérico.

Exemplo: @ LL++,00 SAY "[*LS(7.3)]"

PS.1: LS é uma abreviatura de *Line Spacing*.

Função para Código de Barras

O SIBRA pode imprimir em **14 tipos** de código de barras: EAN13; EAN8; UPCA; UPCE; CODE39 (4 tipos); ITF (2 tipos); MSI (3 tipos); CODABAR (2 tipos); CODE93 (2 tipos); CODE11 (2 tipos); CODE128 (3 tipos); PDF417, DATAMATRIX e QRCODE!.

Com ou sem numeração e descrição nas barras, ajuste de tamanho e altura.

Funciona com a lib **hbzebra** disponível tanto no xHarbour v1.2.3 quanto o Harbour v3.2.

CB(cBARRAS, aXY, cTIPO, nZOOM, lCODIGO, nALTURA, cDES, lmm, lDESC, nFLAGS)

Onde:

cBARRAS = É o número a ser representado pelo código de barras. Tipo caracter.

aXY = Matriz de 4 (quatro) posições. aXY[1]=posição do código de barras no eixo X (horizontal) em pixels ou milímetros; aXY[2]=posição do código de barras no eixo Y (vertical) em pixels ou milímetros; aXY[3]=largura do código de barras em pixels ou milímetros; aXY[4]=altura do código de barras em pixels ou milímetros. O padrão é onde está a linha (row()) e coluna (col()).

cTIPO = Tipo do código de barras: 'EAN13', 'EAN8' etc. O padrão é 'EAN13'.

nZOOM = aumenta ou diminui o tamanho da imagem preservando suas proporções ao tamanho original. Número maior, aumenta; número menor, diminui. O Zoom padrão é 1. Este parâmetro é multiplicado pelo tamanho das barras.

lCODIGO = Imprime ou não, a numeração do código de barras. O padrão é .T. (imprime).

nALTURA = Aumenta a altura do código de barras. Este parâmetro é multiplicado pela altura padrão das barras. O padrão é 1. Por exemplo, caso queira que as barras sejam 50% mais altas, use 1.5; caso queira que seja o dobro da altura, use 2.

cDES = Descrição do objeto referente ao código de barras. Texto que aparecerá no canto superior esquerdo do código de barras. Este parâmetro é opcional.

lmm = .T. ou .F. Define se os parâmetros serão em milímetros (.T.) ou em pixels (.F.). O padrão é .F. Normalmente deixado em branco.

nFLAGS = Zebra flag. Para facilitar, esse parâmetro poderá variar de 0 a 5 conforme a seguir:

- 0 (PADRÃO) = NIL
- 1=HB_ZEBRA_FLAG_CHECKSUM
- 2=HB_ZEBRA_FLAG_CHECKSUM + HB_ZEBRA_FLAG_WIDE2_5

- 3=HB_ZEBRA_FLAG_CHECKSUM + HB_ZEBRA_FLAG_WIDE3
- 4=HB_ZEBRA_FLAG_WIDE3
- 5=HB_ZEBRA_FLAG_WIDE2_5

O valor padrão (*default*) é **0** exceto para CODABAR e CODE11 que é **4**. Se colocar o valor correspondente ao DEFINE também funcionará, ou seja, 0=0, 1=1, 2 ou 65, 3 ou 129, 4 ou 128 e 5 ou 64.

Exemplo:

```
cEAN13 := "789136043666" // código de barras
cDESC := "Camisa Branca" // objeto que o código identificará
@ LL++,00 SAY "EAN13: [*CB('"+cEAN13+"',, 'EAN13',,,, '"+cDESC+"')]"
```

Veja todos os exemplos de código de barras no arquivo **TST1.PRG**

Boletos

O tipo de código de barras para boleto no Brasil é o ITF-14. Observe que a composição da linha digitável e do código de barras pode ser diferente, ou seja, as informações podem ficar embaralhadas. Veja, por exemplo, o guia do **Banco Santander**:



Linha Digitável:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0	3	3	9	9	2	1	3	6	3	1	6	3	0	0	0	0	0	0	2	9	8	2	9	9	8	3	0	1	0	1	2	9	7	3	2	5	0	0	0	0	0	5	6	6	2	4

- Código do Banco
- Moeda
- Código do Convênio/Identificação do Beneficiário
- Nosso Número
- Fixo
- Modalidade

Código de Barras (representação numérica):

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
0	3	3	9	9	7	3	2	5	0	0	0	0	0	5	6	6	2	4	9	2	1	3	6	1	6	3	0	0	0	0	0	0	2	8	2	9	9	8	3	0	1	0	1

Função para utilizar qualquer tipo de letra (Font Face)

[*FON('Nome da fonte', CharSet)]

Nome da fonte: Coloque o nome completo da fonte em aspas simples.

CharSet padrão é **255** para fontes de letras normais no modo Console OEM. Consulte a tabela no tópico CHAR SET deste manual para consultar os valores possíveis.

Observe que fontes de símbolos como Wingdings e Webdings o CharSet é **2** (Symbol Charset). O SIBRA já vai colocar o CharSet **2** para qualquer fonte que contenha “dings”.

Observe o TST1.PRG que acompanha o pacote e serve de exemplo. Observe também que isso só funciona no modo gráfico, no modo texto (Draft) este comando será ignorado.

Exemplo: @ prow(),pcol() SAY "[*FON('Monotype Corsiva')]"

Você pode baixar fontes de códigos de barras também. Veja: http://www.netfontes.com.br/fontes.php/categ_7.10.htm

Função para utilizar qualquer tamanho de fonte

[*CPI(nCPI)]

Onde nCPI é um número correspondente ao CPI do tamanho da fonte. Como é um número, não leva aspas simples. Observe o TST1.PRG que acompanha o pacote e serve de exemplo. Observe também que isso só funciona no modo gráfico, no modo texto (Draft) este comando será ignorado.

Exemplo: @ prow(),pcol() SAY "[*CPI(15)]"

Função para utilizar efeito zebrado

[*ZEBRADO]

Função que pinta linha sim e linha não de cinza. Técnica de diagramação bastante utilizada atualmente para facilitar a leitura dos relatórios. Use em cada linha que desejar o efeito. *Exemplo:*

```
DO WHILE !CAD->(EOF())
    @ LL++,00 SAY "[*ZEBRADO]" + CAD->NOME + " - " + CAD->TELEFONE
    CAD->(DBKIP())
ENDDO
```

ELEMENTOS GRÁFICOS BÁSICOS

[*SETPEN(<nStyle>, <nWidth>, <nColor>)]

Define o estilo, largura e cor do tracejado dos elementos gráficos básicos. A caneta padrão é sólida, 1px e preta. Pode ser combinado com todos os elementos gráficos.

nSTYLE: estilo do tracejado.

nWIDTH: largura do tracejado em pixels.

nCOR: Cor do tracejado.

Estilo	Tipo	Exemplo
0	Caneta desenha linha sólida	
1	Caneta desenha tracejado	(-----)
2	Caneta desenha pontilhado	(.....)
3	Caneta desenha traços e pontos	(-.-.-.-)
4	Caneta desenha traços e 2 pontos	(-.-.-.-)
5	Caneta invisível	

Exemplo: @ prow(),pcol() SAY "[*SETPEN(0, 1, 65793)" // caneta padrão

[LINE]

Desenha uma linha reta de ponta a ponta na borda superior.

[*LINE(<nX1>, <nY1>, <nX2>, <nY2>)]

Desenha uma linha as coordenadas gráficas informadas. X=horizontal, Y=vertical. X1, Y1=Canto superior. X2, Y2=Canto inferior.

[BOX]

Desenha um quadro na mesma linha de ponta a ponta.

[BOXO]

BOX com as extremidades arredondadas.

[*BOX(<nTOP>, <nLEFT>, <nBOTTOM>, <nRIGHT>, [<nWidth>], [<nHeight>])]

~~Desenha uma caixa, um box, quadrado, nas coordenadas gráficas informadas.~~

~~Os 2 últimos parâmetros são arredondamentos dos cantos: horizontal e vertical. Desenha uma caixa com cantos arredondados, se quiser, caso informe o 5º e 6º parâmetro.~~

~~**Exemplo:** @ LL, 30 SAY "[*BOX(10,100,4940,200)]"~~

Essa função foi reformulada facilitar o seu uso. Não precisa informar as coordenadas gráficas, mas sim as posições de linhas e colunas como é feito com o "@ TO" ou "@ BOX" que desenha um quadro na tela, mas da mesma forma na impressora.

- **nTOP** e **nLEFT** = coordenadas do canto superior esquerdo do quadro.
- **nBOTTOM** e **nRIGHT** = coordenadas do canto inferior direito do quadro.
- **nWIDTH** e **nHEIGHT** = curvatura radial das bordas, horizontal e vertical.

Caso precise apenas de um quadro na mesma linha basta chamar "[*BOX()]" sem parâmetros e caso precise que o quadro se prolongue mais algumas linhas abaixo basta informar o número de linhas (informe só 1 parâmetro), por exemplo: "[*BOX(3)]" desenha um quadro da linha atual até 3 linhas para baixo.

Muito mais fácil e prático!

[*FILLRECT(<nX1>, <nY1>, <nX2>, <nY2>, <nRGBColor>)]

Desenha uma caixa preenchida com determinada cor. Tem que informar o número da cor.

Exemplo: @ LL,00 SAY "[*FILLRECT(10,100,4940,200,8716288)]"

[*ARC(<nX1>, <nY1>, <nX2>, <nY2>)]

Desenha um arco.

[*ELLIPSE(<nX1>, <nY1>, <nX2>, <nY2>)]

Desenha uma elipse ou círculo dependendo das coordenadas informadas. Se as coordenadas equivalem a um quadrado, então ela desenha um círculo dentro dessas coordenadas.

Exibir mensagem enquanto roda relatório

Copie a inteligência do exemplo a seguir:

```
#include "Set.ch" // COLOQUE ESTA LINHA NO INÍCIO DO PROGRAMA

// SALVA AS CONFIGURAÇÕES DO SIBRA
Ldevice := SET(_SET_DEVICE)
Lprinter := SET(_SET_PRINTER)
Cfile := SET(_SET_PRINTFILE)

// ALTERA A SAÍDA PARA A TELA
SET DEVICE TO SCREEN
SET PRINTER OFF

// EXIBE MENSAGEM ENQUANTO RODA O RELATÓRIO
@ 10,10 SAY "OLÁ! ESTA MENSAGEM PODE SER VISTA ENTRE AS CHAMADAS DA
FUNÇÃO RELATORIO()" // SUGESTÃO: PONHA UMA BARRA DE PROGRESSO

// RETORNA AS CONFIGURAÇÕES DO SIBRA
SET(_SET_DEVICE, Ldevice)
SET(_SET_PRINTER, Lprinter)
SET(_SET_PRINTFILE, Cfile)
```

PARTE III: CRIANDO PLANILHAS NO EXCEL

Quando você determina que o relatório será uma planilha do Excel com o 5º parâmetro da 2ª chamada, a função RELATORIO() do SIBRA.LIB vai chamar o SIBRAXL.EXE ao invés do SIBRA.EXE. Tenha certeza de que o SIBRAXL.EXE está na mesma pasta do sistema para criar a planilha.

O SIBRAXL.EXE vai detectar a presença do MS-EXCEL na sua máquina para que possa continuar. Caso você não o tenha instalado, você não conseguirá criar uma planilha.

Criar uma planilha do Excel pelo SIBRAXL significa que você terá que criar um relatório totalmente do padrão que você está acostumado. Não dá pra sair colocando o texto na tela como um relatório convencional, o Excel precisa saber em que célula você vai colocar as informações e qual será a formatação dela.

Desta forma, o relatório que você vai criar não vai ficar legível para um ser humano, ou seja, será um monte de comandos que só o SIBRAXL poderá ler e transformar aquilo tudo em uma planilha do Excel legível para o usuário. Portanto, não estranhe e tenha fé! Não deixe de verificar o PRG de exemplo que acompanha o pacote para se familiarizar com a forma de criar uma planilha.

Observe que os comandos explicados a seguir são textos de um relatório. Você vai chamar a função RELATORIO() e colocar eles como se estivesse fazendo um relatório com o comando @ nL,nC SAY. Desta forma, para saber utilizar bem tais comandos você tem que dominar a arte de concatenar strings com variáveis e campos para poder trabalhar no seu relatório, conforme ensinamos em nosso programa TST1.PRG que acompanha o produto e exemplifica de forma prática seu funcionamento.

Os parâmetros da função RELATÓRIO() serão os mesmos. O SIBRAXL vai aproveitá-los na criação da planilha, exceto os que doravante vamos apresentar.

PADRÕES DA PLANILHA

O tamanho das fontes e o modo retrato ou paisagem já temos definido pela função RELATORIO(). Vamos precisar apenas definir o tipo de letra (font face) padrão (default) da planilha, conforme a seguir:

[FONT: nome da fonte]

Com o comando [FONT: nome da fonte] você vai definir o tipo de letra (font face) que o Excel adotará para todas as células, caso você não especifique qual letra terá determinada célula.

Exemplos:

[FONT:Verdana]

[FONT:Arial Narrow]

DEFININDO UM CABEÇALHO NA PLANILHA

Isto é o que mais temos de interessante aqui. Você vai dizer as primeiras linhas que farão parte do cabeçalho, mostrar número de páginas e logotipo. Estas informações serão exibidas automaticamente em cada início de página como cabeçalho.

[LINHAS:n]

Exemplos:

Cabeçalho de 5 linhas: [LINHAS:5]

Cabeçalho de 7 linhas: [LINHAS:7]

[PAG]

Mostra o número da página e a quantidade de páginas total do relatório no canto direito do cabeçalho, no formato “Pág.: 1/10”. Onde 1 é a página atual e 10 é o número total de páginas, por exemplo. Caso não queira mostrar, basta não colocar [PAG] no relatório.

[LOGO(cALIGN):cArquivo]

Mostra uma imagem ou logotipo no cabeçalho. **cAlign** é o alinhamento horizontal da imagem que poderá ser E (canto esquerdo), C (centralizada) e D (canto direito).

cArquivo é o nome do arquivo de imagem com a extensão. Se a imagem não estiver no diretório (pasta) corrente, terá que informar o PATH (caminho) completo da imagem.

Exemplos:

Colocar logotipo bitmap no canto esquerdo do cabeçalho: **[LOGO(E):logo.bmp]**

Colocar imagem do centro do cabeçalho:

[LOGO(C):C:\SISTEMA\IMG\IMAGEM.JPG]

[BORDAS:C1nC2]

Numa planilha sempre tem os títulos de cada coluna com as bordas fechadas fazendo um quadrado. Aqui basta você dizer qual linha conterà as bordas e qual vai começar do lado esquerdo e terminar do lado direito.

Exemplos:

Linha 5, começa pela célula A e termina na célula G: **[BORDAS:A5G]**

Linha 7, começa pela célula B e termina na célula E: **[BORDAS:B7E]**

MESCLANDO CÉLULAS

[MESCLA(cFAIXA)]

Esse comando vai mesclar as células informadas na string **cFAIXA**.

Onde **cFAIXA** é uma string no formato A1:M1 como no Excel. As colunas são informadas como letras e linhas como números. O intervalo é separado por dois pontos (“:”).

Exemplos:

Para mesclar as células A1 (coluna A, Linha 1) até F1 (coluna F, linha 2), utilize:

[MESCLA(A1:F1)]

FORMATANDO COLUNAS

Temos a formatação de tamanho de coluna e alinhamento horizontal padrão de coluna.

[TCOL(nCOL,...):nSize]

O comando **TCOL** foi feito para determinar uma certa largura de uma ou mais colunas, onde **nCOL** é o número de uma ou mais colunas separadas por vírgula, enquanto que, **nSize** é o tamanho da largura da coluna.

É equivalente ao Excel: **FORMATAR > COLUNA > LARGURA > LARGURA DA COLUNA.**

Exemplo:

Se você quiser que a coluna 3 e 4 (C e D) tenha 30 de largura, então:

[TCOL(3,4):30]

[HCOL(cAlignment):nCOL,...]

O comando **HCOL** foi feito para determinar que uma certa coluna tenha um determinado alinhamento horizontal. Dependendo do tipo da informação que ficará nela, é melhor que fique do lado esquerdo da célula, centralizado ou do lado direito.

cAlignment pode ser C (centralizado), E (esquerda), D (direita), J (Justificado), R (DistRibuído), M (meio, centralizar seleção), P (Preenchido) e G (Geral), enquanto que, **nCols** poderá ser uma certa coluna ou várias colunas separadas por vírgula que terão o mesmo alinhamento horizontal.

Exemplo:

Se você quiser que a coluna 3 e 4 (C e D) tenha alinhamento no lado horizontal esquerdo de todas as células desta coluna:

[HCOL(E):3,4]

COLOCANDO CONTEÚDO NA PLANILHA: COMANDO CÉLULA

Todo texto é inserido na planilha com o comando :

[CEL(nL,nC):<xPAR1>:<xPAR2>]

Onde **nL,nC** é a linha e a coluna que representa a posição da célula do Excel. Você sabe que o Excel trabalha com letra para colunas e número para linhas, mas aqui neste comando você vai tratar apenas com números para a linha e coluna também. Você vai interpretar as letras do alfabeto das colunas como se fossem números, por exemplo: A =1; B=2; C=3; D=4; E=5; F=6... e assim por diante.

Exemplo: Célula A1 = (1,1); Célula C5 = (5,3); F10 = (10,6)

Todos os comandos [CEL(nL,nC)] você tem que determinar qual é a célula pela posição da linha e coluna conforme explicado acima. Vamos explicar o restante do comando a seguir.

Alguns comandos tem 2 parâmetros, o segundo vai passar o valor para o primeiro, ou seja, o segundo vai dizer o que o primeiro tem. Vamos tentar explicar melhor com os exemplos a seguir.

Observe que os comandos sempre começam ([]) e terminam (]) com colchetes.

COMANDOS CÉLULA DE 2 PARÂMETROS

TEXTO – DATA - NUMERO – TAM – FONT – AV – AH – FORMULA – COR – INTERIOR – FORMATO - BORDA

Estes são os principais comandos. Eles vão colocar conteúdo na célula como texto; tamanho; fonte; alinhamento vertical e horizontal etc.

[CEL(nL,nC):TEXTO:cText]

TEXTO é o comando que você vai utilizar para colocar texto na célula.

Exemplo: [CEL(1,1):TEXTO:Esta é minha primeira linha do Excel pelo Clipper!]

[CEL(nL,nC):DATA:cText]

DATA é o comando que você vai utilizar para colocar data na célula.

Exemplo: [CEL(1,1):DATA:31/03/2011]

*OBS.: As células de datas são automaticamente formatadas com SET(4), ou seja, é automático tomando por base o SET DATE TO do Clipper. Se você utilizou SET DATE TO BRITISH, o formato das datas já serão dd/mm/aaaa automaticamente!

[CEL(nL,nC):NUMERO:cText]

NUMERO é o comando que você vai utilizar para colocar número na célula.

Exemplo: [CEL(1,1):NUMERO:1000.00]

[CEL(nL,nC):TAM:nSize]

TAM é o comando que você vai utilizar para definir o tamanho das letras, o *font size*. O valor de **nSize** sempre será um número, mas você tratará como caracter. Observe os exemplos.

Exemplo:

[CEL(1,1):TAM:22]

[CEL(1,1):VALOR:Esta linha terá o tamanho de fonte 22]

[CEL(nL,nC):FONT:<cFontFace>]

FONT é o comando que você vai utilizar para definir o tipo de letra que você vai utilizar em determinada célula, o *font face*.

Exemplo:

[CEL(1,1):FONT:Monotype Corsiva]

[CEL(1,1):VALOR:Esta linha terá a letra do Monotype Corsiva]

[CEL(nL,nC):AV:cAlignment]

AV é o comando que você vai utilizar para definir o Alinhamento Vertical de determinada célula. Os parâmetros de **cAlignment** são T (Topo), F (Fim), C (Centro), J (Justificado) e D (Distribuído).

Exemplos:

[CEL(1,1):AV:F]

[CEL(1,1):VALOR:Esta linha estará verticalmente no fim da célula.]

[CEL(1,1):RETORNO]

[CEL(nL,nC):AH:cAlignment]

AV é o comando que você vai utilizar para definir o Alinhamento Horizontal de determinada célula. Os parâmetros de **cAlignment** para o alinhamento são C (centralizado), E (esquerda), D (direita), J (Justificado), R (DistRibuído), M (meio, centralizar seleção), P (Preenchido) e G (Geral).

Exemplo:

[CEL(1,1):AH:C]

[CEL(1,1):VALOR:Esta linha estará horizontalmente centralizada na célula.]

[CEL(1,1):RETORNO]

[CEL(nL,nC):FORMULA:cFORMULA]

FORMULA é o comando que você vai utilizar para colocar uma fórmula na célula. O parâmetro **cFORMULA** é a fórmula em formato texto conforme se coloca no Excel tendo as colunas por letras e as faixas de células separadas por dois pontos (:).

Exemplo:

Para somar as células de E3 até E10 e colocar o resultado na célula E11.

[CEL(11,5):FORMULA:SUM(E3:E10)]

[CEL(nL,nC):COR:cFONTCOLOR]

COR é o comando que você vai utilizar para colocar uma determinada cor no texto da célula. Os parâmetros de **cFONTCOLOR** são PRETO, BRANCO, AZUL, AZUL CLARO, AZUL ESCURO, VERDE, VERMELHO, AMARELO, MARRON, VIOLETA, LARANJA e ROSA.

Exemplo:

Colocar a cor das letras em vermelho:

[CEL(1,1):COR:VERMELHO]

[CEL(nL,nC):INTERIOR:cFONTCOLOR]

INTERIOR é o comando que você vai utilizar para preencher o interior de uma célula com uma determinada cor. Os parâmetros de **cFONTCOLOR** são PRETO, BRANCO, AZUL, AZUL CLARO, AZUL ESCURO, VERDE, VERMELHO, AMARELO, MARRON, VIOLETA, LARANJA e ROSA.

Exemplo:

Preencher a célula de azul escuro com as letras brancas:

[CEL(1,1):INTERIOR:AZUL ESCURO]

[CEL(1,1):COR:BRANCO]

[CEL(nL,nC):FORMATO:cCATEGORIA]

FORMATO é o comando que você vai utilizar para formatar a categoria do número da célula. É o equivalente às configurações que você faz em FORMATAR CÉLULAS > NÚMERO > CATEGORIA no Excel.

Os valores possíveis para **cCATEGORIA** são vários:

'0'
'0.00'
'#,##0'
'#,##0.00'
'#,##0\ "pta";\-#,##0\ "pta"
'#,##0\ "pta";[Red]\-#,##0\ "pta"
'#,##0.00\ "pta";\-#,##0.00\ "pta"
'#,##0.00\ "pta";[Red]\-#,##0.00\ "pta"
'0%'
'0.00%'
'0.00E+00'
'dd/mm/yy'
'dd\ -mmm\ -yy'
'dd\ -mmm'
'mmm\ -yy'
'h:mm\ a.m./p.m.'
'h:mm:ss\ a.m./p.m.'
'h:mm'
'h:mm:ss'
'dd/mm/yy\ h:mm'
'mm:ss'

Exemplo:

Definir que a célula é do tipo porcentagem com duas casas decimais:

[CEL(1,1):FORMATO:0.00%]

Definir que a célula é do tipo numérico com separador de milhares e duas casas decimais:

[CEL(1,1):FORMATO: #,##0.00]

Definir que a célula é do tipo data com a hora do lado:

[CEL(1,1):FORMATO:dd/mm/yy\ h:mm]

*OBS.: As células de datas são automaticamente formatadas com SET(4), ou seja, é automático tomando por base o SET DATE do Clipper. Se você utilizou SET DATE TO BRITISH, o formato das datas já serão dd/mm/aaaa automaticamente!

COMANDOS CÉLULA DE 1 PARÂMETRO

RETORNO – NEGRITO – ITALICO – SUBLINHADO – SOBRESCRITO – SUBSCRITO – SOMBRA – CONTORNO - TACHADO - BORDA

[CEL(nL,nC):RETORNO]

RETORNO é o equivalente à formatação de célula: Alinhamento > Controle de texto > Retorno automático de texto.

Faz com que o conteúdo da célula caiba dentro da mesma célula, deixando-a mais “gordinha”, verticalmente alongada.

Se você não definir esta formatação da célula, o texto que você for inserir poderá ultrapassar os limites da célula para as células vizinhas da direita.

Exemplo:

[CEL(10,1):RETORNO]

[CEL(10,1):TEXTO:Este texto vai estar dentro da mesma célula]

[CEL(nL,nC):NEGRITO]

NEGRITO coloca o texto da célula em **negrito**.

Exemplo:

[CEL(10,1):NEGRITO]

[CEL(10,1):TEXTO:Este texto vai estar em NEGRITO]

[CEL(nL,nC):ITALICO]

ITALICO coloca o texto da célula em *itálico*.

Exemplo:

[CEL(10,1):ITALICO]

[CEL(10,1):TEXTO:Este texto vai estar em ITALICO]

[CEL(nL,nC):SUBLINHADO]

SUBLINHADO coloca o texto da célula em sublinhado.

Exemplo:

[CEL(10,1):SUBLINHADO]

[CEL(10,1):TEXTO:Este texto vai estar em SUBLINHADO]

[CEL(nL,nC):SOBRESCRITO]

SOBRESCRITO coloca o texto da célula em ^{sobrescrito}.

Exemplo:

[CEL(10,1):SOBRESCRITO]

[CEL(10,1):TEXTO:Este texto vai estar em SOBRESCRITO]

[CEL(nL,nC):SUBSCRITO]

SUBSCRITO coloca o texto da célula em subscrito.

Exemplo:

[CEL(10,1):SUBSCRITO]

[CEL(10,1):TEXTO:Este texto vai estar em SUBSCRITO]

[CEL(nL,nC):SOMBRA]

SOMBRA coloca sombra no texto da célula.

Exemplo:

[CEL(10,1):SOMBRA]

[CEL(10,1):TEXTO:Este texto vai estar com SOMBRA]

[CEL(10,1):TAM:22]

[CEL(nL,nC):CONTORNO]

CONTORNO coloca contorno no texto da célula (OutLine).

Exemplo:

[CEL(10,1):CONTORNO]

[CEL(10,1):TEXTO:Este texto vai estar com CONTORNO]

[CEL(10,1):TAM:22]

[CEL(nL,nC):TACHADO]

TACHADO coloca o texto da célula riscado.

Exemplo:

[CEL(10,1):TACHADO]

[CEL(10,1):TEXTO:Este texto vai estar TACHADO, ou seja, riscado]

[CEL(10,1):TAM:22]

[CEL(nL,nC):BORDA]

BORDA coloca as 4 bordas em uma célula.

Exemplo:

[CEL(10,1):BORDA]

PARTE IV: MODO *STAND ALONE*



Imagem do SIBRA em **Stand Alone** com um arquivo de script aguardando arquivos



Imagem do SIBRA em **Stand Alone** com um arquivo de script imprimindo um arquivo

O que é?

O modo *Stand Alone* é a modalidade de impressão onde o SIBRA controla a impressão sozinho, sem auxílio da função RELATORIO(), contida no sibra.lib.

Sintaxe

SIBRA [arquivo | mascara | @script] [saida] [impressora] [cpi] [/S]

Onde:

Arquivo|Máscara|@Script = nome, mascara ou arquivo de script =>STAND ALONE
Arquivo = imprime apenas o arquivo especificado na impressora padrão.
Máscara = imprime todos os arquivos que equivalerem à máscara especificada na impressora padrão. Ex.: *.TXT
@Script = imprime os arquivos conforme orientação do arquivo de Script que pode determinar uma impressora para cada tipo de arquivo. Trataremos com mais detalhes adiante. Ex.: @multi.scr

saida = /P para imprimir
 /V para visualizar

impressora = 1 imprime na impressora padrao
 0 escolhe a impressora de uma lista
 2 escolhe a impressora WINDOWS de uma lista
 3 escolhe a impressora MATRICIAL/LPT1 de uma lista
 4 Detecta automaticamente uma MATRICIAL e imprime na LPT1. Vide o 5º parâmetro da 1ª chamada da função RELATORIO() para entender melhor.

cpi = /10CPI
 /12CPI
 /17CPI
 /20CPI

/S = Executa sem pedir confirmação para prosseguir ou cancelar.

Como funciona?

Funciona assim: execute o SIBRA.EXE, volte ao Windows e **execute o seu sistema em outra janela (pois, o SIBRA não é TSR, não fica residente)**. O importante é que o Windows esteja executando o SIBRA.EXE e seu sistema ao mesmo tempo!

Caso você tenha perdido os fontes do seu sistema em Clipper, se ele tiver a função de gravar os relatórios em arquivo, você os grava em arquivo que assim que o SIBRA detectar um arquivo de relatório, ele o imprime, tipo um *spool* de impressão.

O SIBRA pesquisa o diretório corrente a cada segundo para encontrar arquivos para imprimir. Ele imprime todos eles e depois os apaga!!! Portanto, cuidado! Não deixe arquivos que você não quer que imprima no mesmo diretório do SIBRA.EXE, porque além de imprimir, ele o apagará! Mesmo se você escolher visualizar na tela, ele apaga!

O SIBRA apaga os arquivos depois de imprimir porque assume que os arquivos são temporários e foram criados só para serem impressos naquela hora e depois apagados. Assim, evita que muitos arquivos temporários se acumulem no diretório.

Você pode informar a extensão dos arquivos que quiser que o SIBRA imprima, assim como a impressora que vai imprimir.

Importante

Para que o SIBRA funcione corretamente no modo *Stand Alone*, selecione uma impressora do tipo **EPSON LX** na configuração do seu sistema, pois o SIBRA substitui os códigos da impressora da EPSON pelos códigos universais do SIBRA, ou seja, quando o SIBRA encontra CHR(15)+CHR(27)+"M", ele entende como se estivesse encontrado "[20cpi]"; portanto não importa qual impressora você vai imprimir efetivamente, escolha uma **EPSON LX** no seu sistema e depois defina a impressora com o SIBRA.

Arquivo de SCRIPT no STAND ALONE

Você pode imprimir em qualquer impressora de rede por meio de um arquivo de Script, ensinando o SIBRA quais arquivos e quais impressoras.

Instale todas as impressoras de rede no Servidor; defina uma extensão de arquivo para os relatórios de cada estação da rede; determine que arquivos com determinada extensão imprimam em determinada impressora. Isto fará seu sistema imprimir na USB sem mudar nenhuma linha do seu programa!

Comandos do Script

O arquivo de Script é um arquivo de texto comum contendo comandos com parâmetros. Os parâmetros sempre vêm em COLCHETES "[]"; os comandos são separados um do outro por um espaço em branco. Cada linha do Script é uma instrução diferente. Os comandos existentes são os seguintes, sempre na ordem:

ARQUIVOS > IMPRESSORA > CPI > LS > [PAISAGEM] > SP

Apenas os comandos **ARQUIVOS** e **IMPRESSORA** são obrigatórios. Os demais comandos são opcionais; se omitidos, usa-se o valor padrão.

ARQUIVOS – Este comando leva a máscara dos arquivos a serem impressos, como parâmetro. Ex.: *.TXT; *.IP1; *.IP2 etc.

Determine que cada computador da rede grave arquivos com a extensão relacionada à máscara que vai usar a **IMPRESSORA** que vai imprimi-los.

O básico é que cada computador da rede grave arquivos com uma extensão de arquivo diferente para imprimir com a sua **IMPRESSORA**.

IMPRESSORA – Este comando leva o nome da impressora batizada no Windows que vai imprimir os **ARQUIVOS**. O nome tem que ser exatamente igual ao nome da

impressora no Windows! Não é *case-sensitive*! Tanto faz maiúsculas como minúsculas. Ex.: Epson Stylus Color 740; 80/86/800/810; Epson LX-810 etc.

CPI – Caracteres Por Polegada que serão impressos os **ARQUIVOS** na **IMPRESSORA**. Aceita os parâmetros: 10 (padrão), 12, 17 e 20.

LS – Espaçamento de linha que será dado na impressão dos **ARQUIVOS** na **IMPRESSORA**. Aceita os parâmetros: 6 (padrão), 8 e 9 no modo Draft (impressora matricial) e 4, 5, 6 (padrão), 7, 8 e 9 no modo gráfico (demais impressoras).

[PAISAGEM] – Imprime os **ARQUIVOS** na **IMPRESSORA** em modo paisagem (Landscape). Este comando não possui parâmetros e é opcional. Sua existência determina Modo Paisagem; sua ausência, Modo Retrato (padrão).

SP – Abreviatura de *Speed* (velocidade). Define a velocidade e a qualidade de impressão. Valores possíveis 1, 2, 3 e 4. Sendo o valor “1” o mais rápido (menor qualidade) e “4” o mais lento (maior qualidade). O valor padrão é “1”.

Interpretando um exemplo de Script

Vamos analisar e interpretar o conteúdo do arquivo de Script abaixo contido no DEMO.ZIP, o MULTI.SCR:

```
ARQUIVOS[*].LOG] IMPRESSORA[80/86/800/810] CPI[12] LS[6]  
ARQUIVOS[*].IP2] IMPRESSORA[EPSON Stylus COLOR 740] CPI[20] LS[8] [PAISAGEM]
```

Linha 1: Todos os arquivos com a extensão .LOG serão impressos na impressora 80/86/800/810 (nome de impressora matricial, imprimirá em modo DRAFT) com 12CPI e espaçamento de linha de 1/6”.

Linha 2: Todos os arquivos com a extensão .IP2 serão impressos na impressora Epson Stylus COLOR 740 com 20CPI, espaçamento de linha de 1/8” e em modo paisagem.

MULTI.SCR, é correto afirmar que:

1. O *Computador Alfa* tem uma impressora Epson LX-810 e grava arquivos com a extensão .LOG;
2. O *Computador Beta* tem uma impressora Epson Stylus Color 740 e grava os arquivos com a extensão .IP2;
3. O SIBRA está rodando em modo **Stand Alone** com o script MULT.SCR no *Computador Servidor da Rede*, assim: SIBRA @MULTI.SCR /S;
4. O *Computador Servidor da Rede* possui duas impressoras de rede instaladas. Uma é do *Computador Alfa* e chama-se 80/86/800/810, é uma Epson LX-810. A outra impressora é do *Computador Beta* e chama-se Epson Stylus COLOR 740, é uma impressora com mesmo nome.

Exemplos

Exemplos no modo stand alone:

SIBRA *.TXT /P 4 /10CPI /S *A máscara deve conter um "*" (asterisco).*

SIBRA @multi.scr /S *O arquivo de script deve preceder de um "@" (arroba).*

Exemplos no modo individual:

SIBRA NATELA.TXT

SIBRA TESTE.TXT "/P:80/86/800/810" 4 /20CPI

SIBRA LISTA.TXT "/P:EPSON Stylus COLOR 740" 4 /10CPI

O 1º exemplo visualiza na tela o arquivo NATELA.TXT;

O 2º imprime o arquivo TESTE.TXT com 20CPI em modo Draft, na impressora chamada 80/86/800/810;

O 3º imprime o arquivo LISTA.TXT com 10CPI na impressora Epson Stylus Color 740

PRONTO! Agora você vai imprimir de todo jeito! Mesmo que você tenha perdido os seus fontes do Clipper, você vai imprimir!!!

PARTE V: PERSONALIZANDO O SIBRA

A partir da versão 3.22 é possível personalizar a janela de opções de relatórios do SIBRA. Você pode mudar a cor de tudo para que se adapte em conformidade com o padrão do seu sistema.

Para isso, siga fielmente as instruções a seguir:

1. Crie um arquivo de texto chamado "LAYOUT.CFG" com o seguinte conteúdo:

nMOVX=0

nMOVY=0

cCORJ=W+/B

cCORF=W+/B, N/W

lSOMBRA=.F.

2. Salve o arquivo na mesma pasta do seu sistema, onde está o SIBRA.EXE.

Analise cada linha:

nMOVX : Move a posição horizontal da janela. Use um valor positivo para mover para a direita e negativo para a esquerda.

nMOVY : Move a posição vertical da janela. Use um valor positivo para mover para baixo e negativo para cima.

cCORJ : Muda a cor da janela

cCORF : Muda as cores do conteúdo da janela executando um SETCOLOR(cCORF).

lSOMBRA : Liga ou desliga a sombra das janelas.

*Não utilize aspas e não mude o nome das variáveis (palavras antes do sinal de igual), altere apenas o conteúdo após o sinal de igual "=".

ARQUIVOS TEMPORÁRIOS

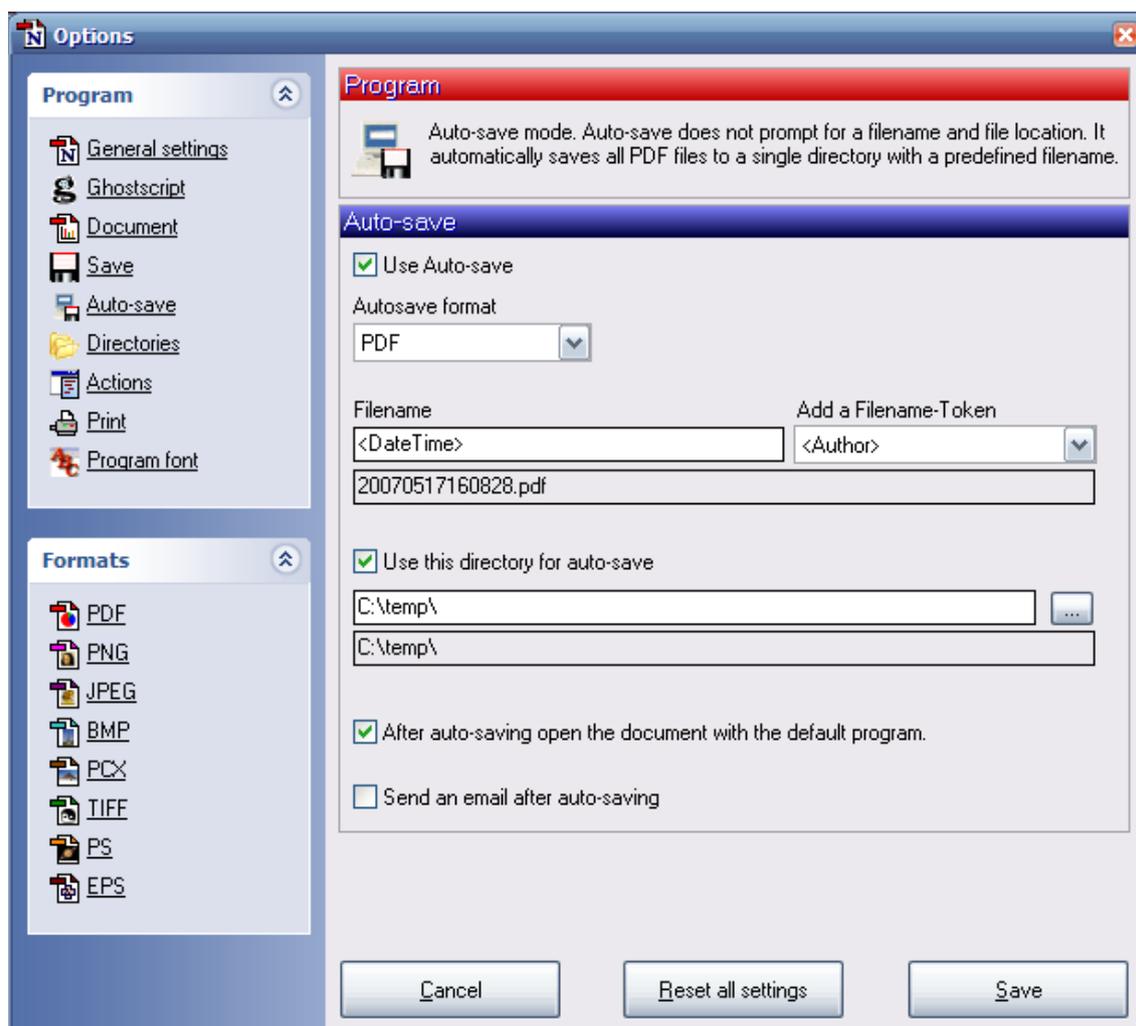
A partir da versão 3.8 o sistema passou a gravar os arquivos temporários na pasta C:\TEMP para dar maior celeridade aos processos reduzindo a utilização da rede.

É possível mudar esta pasta definindo uma variável de ambiente chamada "SIBRATMP". Exemplo: SET SIBRATMP=C:\SIBRA

PARTE VI: THIRDY PARTY

PDFCreator

Instale o **PDFCreator** (Freeware) e crie relatórios em PDF a partir do SIBRA e inclusive, poderá visualizar o PDF criado automaticamente, ou seja, com isto você pode visualizar o relatório em modo gráfico do SIBRA, exatamente como seria no papel!



Site oficial: <http://www.pdfforge.org/products/pdfcreator/download>

Depois de instalar o PDFCreator, siga as instruções seguintes:

OPTIONS > AUTOSAVE

[x] Use autosave
(Habilitar salvamento automático)

[x] After auto-saving open the document with the default program.
(Após o salvamento automático, abrir o documento com o programa padrão)

Se você instalar o PDFCreator e pedir para visualizar o relatório na tela, automaticamente será aberto o PDF do relatório na tela.

*Precisa ter instalado o **Adobe Acrobat Reader** ou outro visualizador de arquivos PDF como o Sumatra.

PARTE VII: TROUBLESHOOTING

Impressora Laser HP 1020

Problemas com a impressão de mais de uma página foram detectados com o driver para Windows 98 desta impressora. Há problemas com outros aplicativos do Windows 98 também, não só o SIBRA.

Aconselha-se procurar um *driver de impressora* mais recente no site do fabricante. Este problema desaparece no Windows XP.

SIBRA não faz nada ou trava

1. Ausência do SIBRA.EXE

Verifique se o arquivo SIBRA.EXE está na mesma pasta do seu sistema.

2. Ausência do interpretador de comandos

Verifique se o interpretador de comandos do ms-dos (COMMAND.COM do Windows 98 ou CMD.EXE no Windows XP) está no caminho (path) especificado na variável de ambiente COMSPEC (veja seu AUTOEXEC.BAT).

Exemplo:

```
SET COMSPEC=C:\WINDOWS\COMMAND.COM
```

Neste exemplo, o arquivo “command.com” precisa existir no diretório “c:\windows” que foi especificado!

Você pode executar o comando: DIR COMMAND.COM /S/P no diretório raiz do seu computador (c:\) descobrir onde está localizado o arquivo. Depois você inclui a variável COMSPEC no seu autoexec.bat especificando o local do arquivo como no exemplo dado.

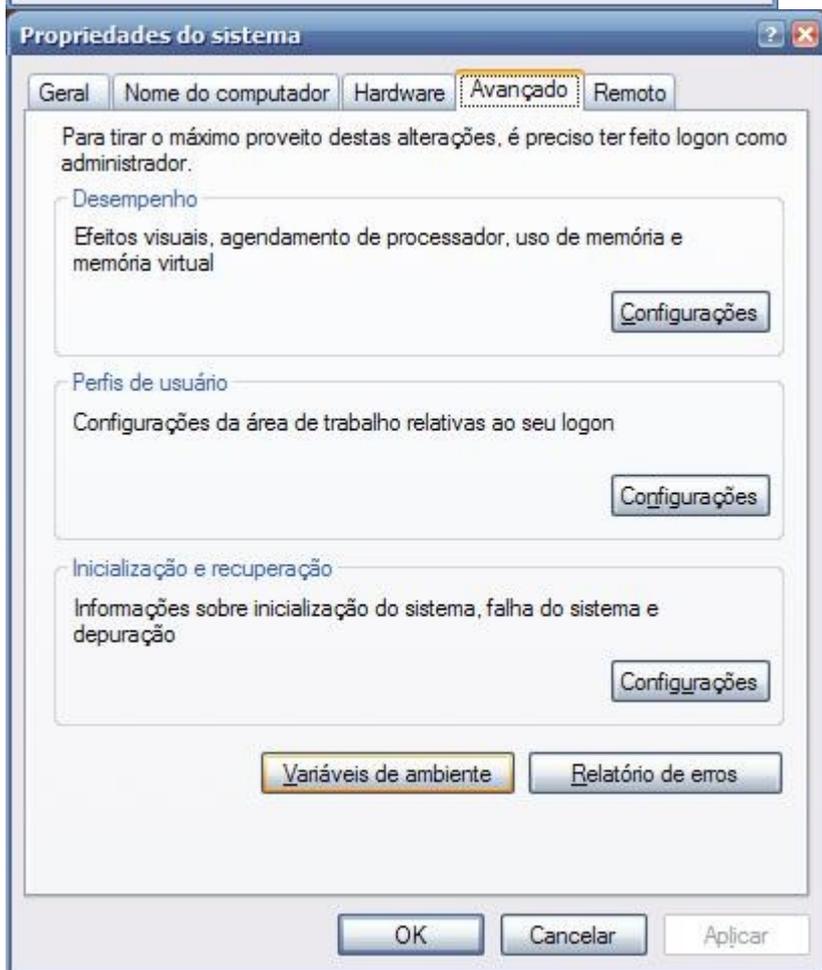
No caso do Windows XP, você altera as variáveis de ambiente em:

Painel de Controle > Sistema



Ou com as teclas [Windows] + [Pause/Break]

Veja as imagens na próxima página a seguir:



3. Propriedades do link do aplicativo

Observe em *propriedades do link* (imagem ao lado) que “Linha de comando:” é o caminho completo do arquivo.

Observe também que “Processando:” é o caminho onde o aplicativo será executado, ou seja, como se ele estivesse sendo chamado daquela pasta.

A imagem ao lado foi tirada do Windows XP, mas funciona da mesma forma no Windows 98.

Lembre que você precisa MAPEAR A UNIDADE DE REDE para trabalhar com o Clipper em rede.

No nosso exemplo:
//servidor/contábil
está mapeado para a unidade **H:**



4. Permissões do usuário

Pode ser que o usuário da máquina não tenha privilégios de acesso suficientes para trabalhar na rede.

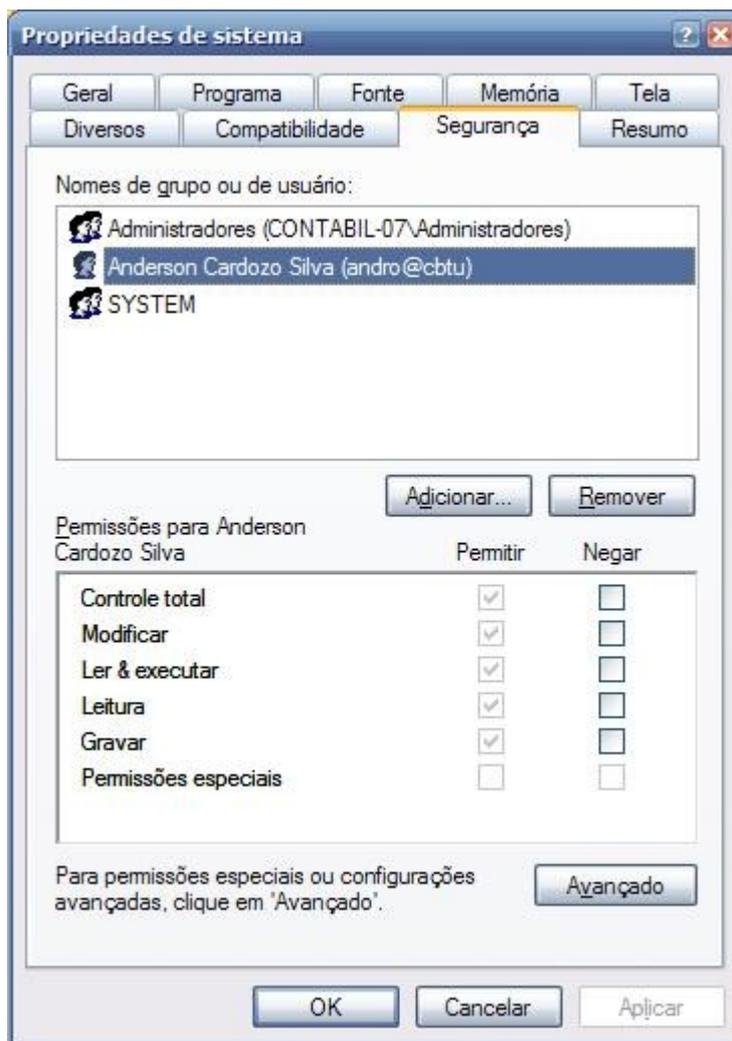
Verifique a conta do usuário em:

PAINEL DE CONTROLE > CONTAS DE USUÁRIO



Observe que na aba “segurança” em *Propriedades do link* (Windows XP) pode-se verificar as permissões do usuário também.

O usuário deve ter acesso completo de rede à pasta do sistema, já que o acesso às informações é controlado pelo próprio sistema e não pela rede.



Caso não tenha resolvido o problema, procure o administrador da rede.

5. Travou

Caso você tente utilizar uma string do SIBRA com parâmetro inexistente, como, por exemplo: “[LS1]”, o sistema trava. Verifique se está passando as strings corretamente.

Cada tipo de código de barras tem uma quantidade exata de números, se ela é de 13, não passe 12 ou 14. Seu leitor não conseguirá ler o código de barras se fizer isso, portanto não esqueça de passar a quantidade certa de números que constitui o código.

Algo errado também pode acontecer se você estiver chamando a função RELATORIO() com os parâmetros incorretos.

Primeira página sai em branco

O SIBRA detecta automaticamente o caracter CHR(12), “♀”, que toda impressora interpreta como salto de página e, inclusive, equivale ao comando EJECT do Clipper.

Se a primeira página sai em branco, analise a rotina de seu relatório. Salve o relatório em arquivo .STM e verifique se existe a presença do caracter “♀”, CHR(12), no início do relatório. O salto de página ocorre quando existe o comando EJECT ou é

emitido uma linha @ nL, nC SAY onde a posição da linha atual (nL) é menor que a posição da linha anterior. *Exemplo:*

```
LL = 0
@ LL++,00 SAY "LINHA 0"
@ LL++,00 SAY "LINHA 1"
@ LL++,00 SAY "LINHA 2"
@ LL++,00 SAY "LINHA 3"
LL = 0
@ LL++,00 SAY "LINHA 0" // 0 menor que 3 = ♀ (EJECT)
@ LL++,00 SAY "LINHA 1"
@ LL++,00 SAY "LINHA 2"
@ LL++,00 SAY "LINHA 3"
```

Talvez você encontre a linha do seu relatório assim:

```
LINHA 3♀LINHA 0
```

É claro que o exemplo acima não vai dar a primeira página em branco, é só um exemplo didático. Cada caso precisa ser analisado individualmente e geralmente é muito fácil descobrir. A maioria das pessoas que relataram este problema, o resolveram sozinho.

Demora para imprimir ou não imprime

Basta instalar o *driver* da impressora de rede no computador local, ou seja, mesmo que não tenha a impressora naquele computador onde está mandando imprimir, instale o *driver* nela.

Assim, mesmo que não precise por se tratar de uma impressora ligado em paralelo via porta LPT, isto deve resolver o problema.

Execute primeiro o **PDFCreator** e deixe-o minimizado para evitar problemas com o *spooler* de impressão do Windows.

Relatório não sai

Não use **SET FIXED ON** no seu sistema porque atrapalha a saída da impressora para arquivo. Isto é uma falha do Clipper. O SIBRA trabalha com criação de arquivos.

Ao invés disso, use máscaras com @...GET...PICTURE ou TRANSFORM().

Erro de programação / Run time error

Envie o arquivo ERROR.LOG para nosso email para que possamos corrigir o problema. Informe as opções utilizadas pelo usuário quando o erro aparece. Se possível, mande cópia apenas do PRG que está dando este problema para verificar como você está chamando a função.

Perdí as fontes e meu sistema não salva relatórios em arquivo

O Windows permite você criar uma impressora imaginária!

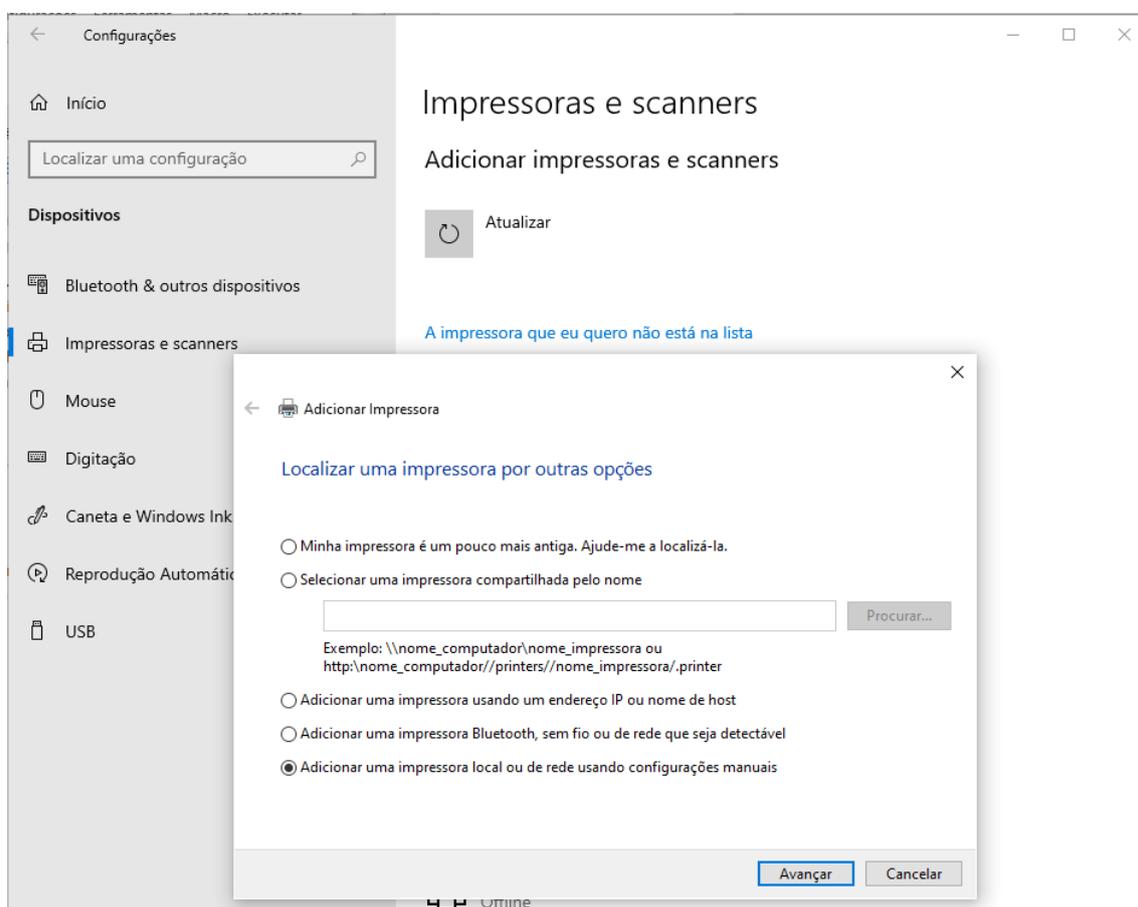
Adicione uma nova impressora local do tipo “Genérico / somente texto” e ao selecionar a porta marque “Imprimir para arquivo”. Nomeie a impressora como **PRINTFILE**, por exemplo.

Depois disso é só fazer o truque com o NET USE, por exemplo:

```
NET USE LPT1: \\COMPUTADOR\PRINTFILE /PERSISTENT: YES
```

No Windows 10 seria assim:

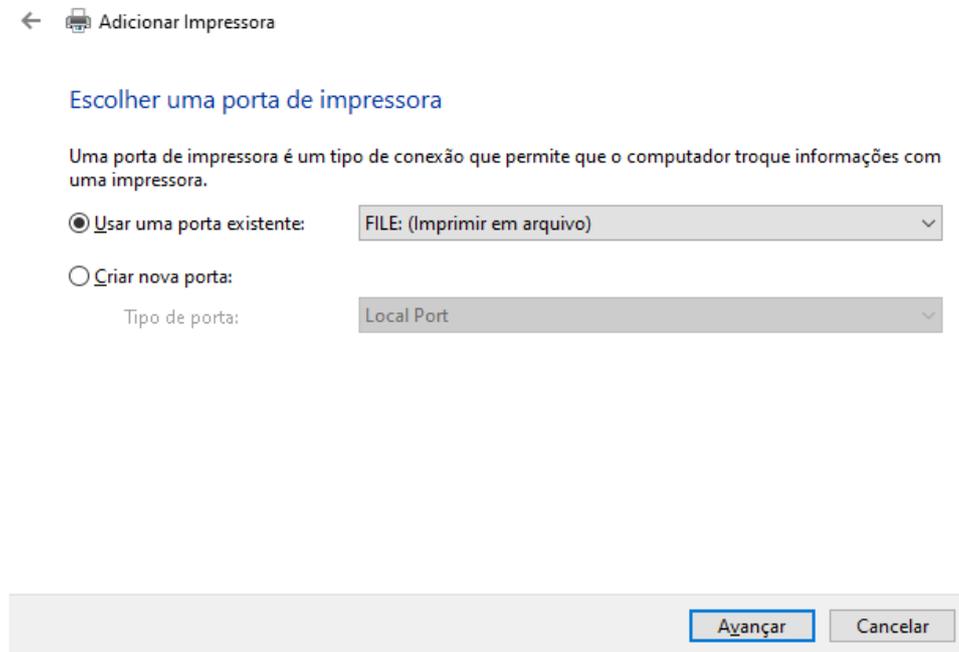
Adicionar impressoras e scanners > A impressora que eu quero não está na lista > Adicionar uma impressora local ou de rede usando configurações manuais > Avançar



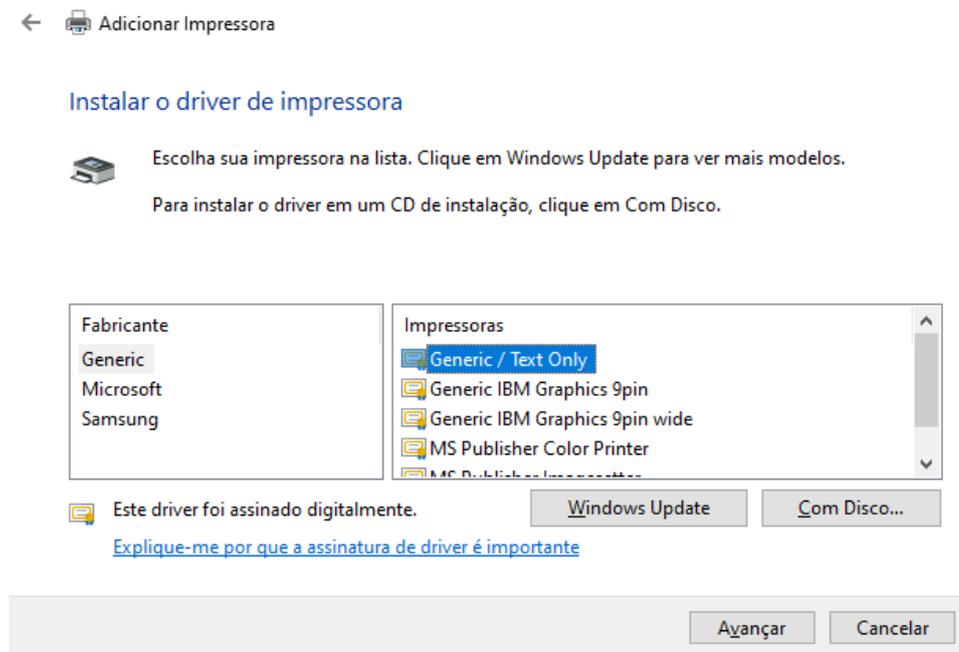
Depois disso irá aparecer outra janela para escolher a porta de impressora.

Escolha **FILE: (Imprimir em arquivo)**

Veja a foto ilustrativa a seguir.

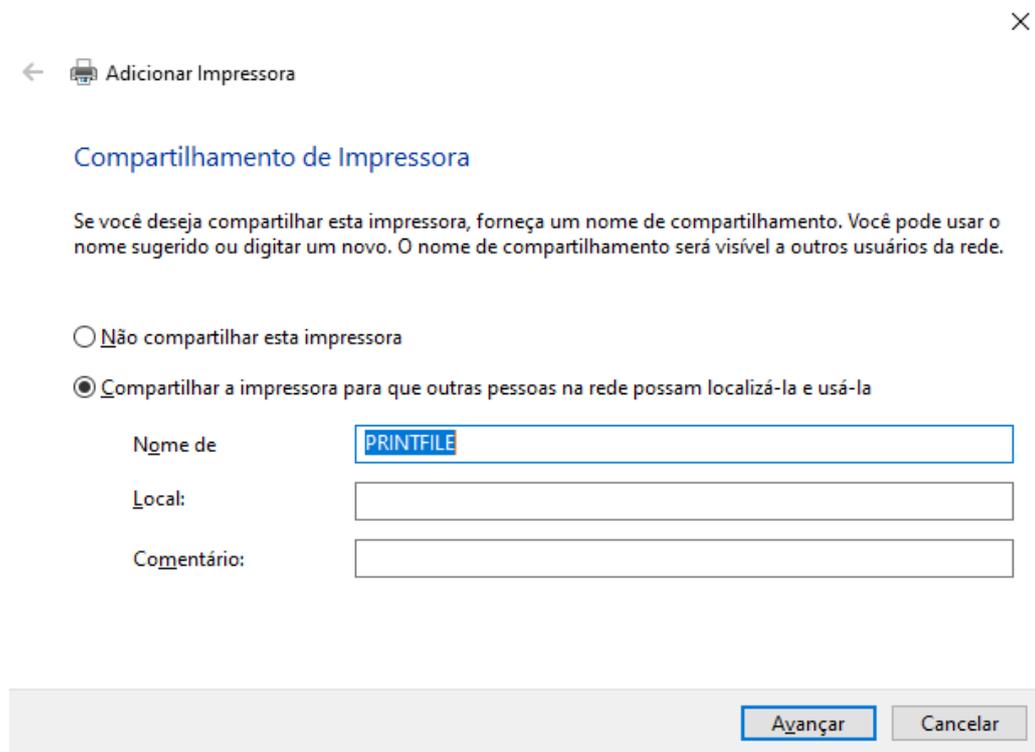


Depois escolha o fabricante **Generic** e a impressora **Generic / Text Only** conforme a foto abaixo.



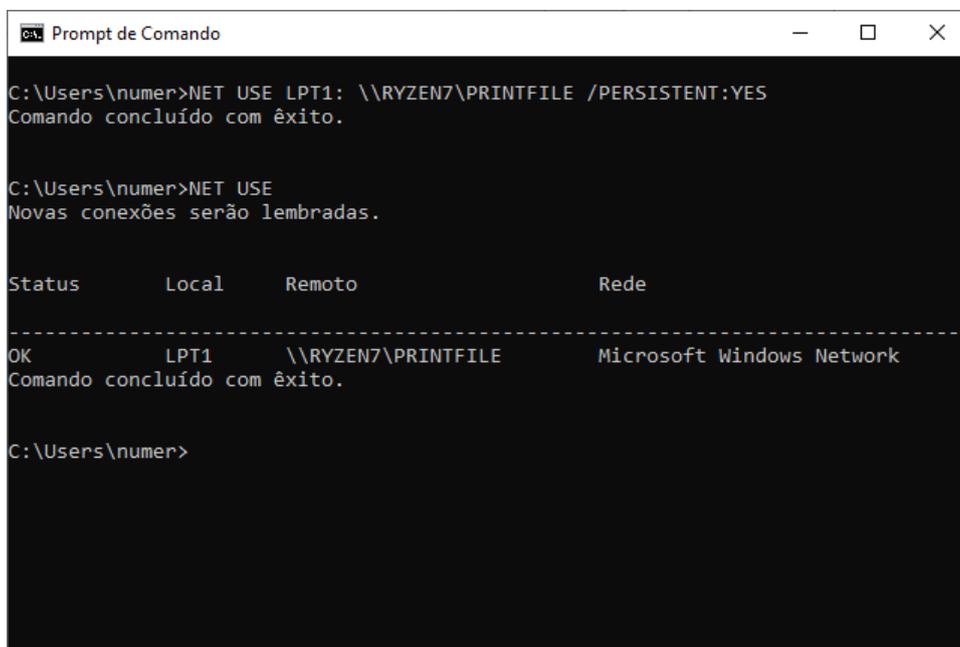
Depois disso é só digitar um nome para a impressora: **PRINTFILE**

Compartilhe a impressora para poder usar o **NET USE**, conforme a foto a seguir.



The screenshot shows the 'Adicionar Impressora' (Add Printer) dialog box in Windows. The 'Compartilhamento de Impressora' (Printer Sharing) tab is selected. The dialog has a title bar with a back arrow, a printer icon, and the text 'Adicionar Impressora'. A close button (X) is in the top right corner. Below the title bar, the text reads: 'Se você deseja compartilhar esta impressora, forneça um nome de compartilhamento. Você pode usar o nome sugerido ou digitar um novo. O nome de compartilhamento será visível a outros usuários da rede.' There are two radio buttons: 'Não compartilhar esta impressora' (unselected) and 'Compartilhar a impressora para que outras pessoas na rede possam localizá-la e usá-la' (selected). Below the radio buttons are three text input fields: 'Nome de' (containing 'PRINTFILE'), 'Local:', and 'Comentário:'. At the bottom right, there are two buttons: 'Avançar' (highlighted) and 'Cancelar'.

Para saber o nome do seu computador pressione as teclas: **Windows+PAUSE/BREAK**. É o nome do dispositivo.



```
cmd Prompt de Comando
C:\Users\numer>NET USE LPT1: \\RYZEN7\PRINTFILE /PERSISTENT:YES
Comando concluído com êxito.

C:\Users\numer>NET USE
Novas conexões serão lembradas.

Status      Local      Remoto      Rede
-----
OK          LPT1      \\RYZEN7\PRINTFILE      Microsoft Windows Network
Comando concluído com êxito.

C:\Users\numer>
```

Atenção: faça o teste antes de comprar!