EPSON
EXCEED YOUR VISION

MANUAL DA INTERFACE EPSON NÃO FISCAL

Ver. 2.1.0

INTERFACE DE ALTO NÍVEL PARA IMPRESSORAS NÃO FISCAIS EPSON

PROGRAMA EPSON PARCERIA DE SOFTWARE

A EPSON disponibiliza exemplos de programação em diversas linguagens e sistemas operacionais, para ter acesso a estes arquivos cadastre-se no **PEPS** (Programa Epson de Parcerias de Software). Basta acessar o site do EpsonStars e realizar sua inscrição, não demora mais do que 1 minuto.

Epson Stars

www.epsonstars.com.br

Índice

1 Convenções

Convenção de Símbolos

Símbolo	Significado...
	Este símbolo indica que o texto que vem logo em seguida é uma referência a outros tópicos deste documento.
	Este símbolo indica que em seguida encontra-se uma dica de como utilizar a interface.

Tabela 1 – Convenção de Símbolos

Tipos de Dados Suportados

Tipo de Dados	Abrev.	Valores permitidos
Alfanumérico	(A)	‘a’-‘z’, ‘A’-‘Z’, ‘0’-‘9’
Alfabético	(L)	‘a’-‘z’, ‘A’-‘Z’
Numérico	(N)	‘0’-‘9’
Binário	(B)	0x00-0xFF
Imprimível	(P)	0x20-0xFF
Hexadecimal	(H)	‘0’-‘9’, ‘a’-‘f’, ‘A’-‘F’
Data	(D)	ddmmaaaa (ex: “30012002)
Hora	(T)	hhmmss (ex: “113034”)
Booleano	(E)	’S’, ‘N’
Texto com atributos de impressão	(RT)	0x20-0xFF, aceitando atributos e códigos de barras.
Opcional	(O)	Campo opcional

Tabela 2 – Tipos de Dados

2 Introdução

Este documento descreve em detalhes a interface de alto nível para Impressoras Não Fiscais Epson. Esta interface pode ser usada em qualquer linguagem de desenvolvimento para o sistema operacional Windows 32/64-bits.

A Interface Epson de alto nível é uma API avançada com funções de máxima performance para a impressora não fiscal e foi concebida de maneira a permitir fácil integração entre a impressora e o aplicativo.

Nas seções seguintes encontram-se informações de como utilizar esta interface e uma descrição detalhada das funções, com seus protótipos e exemplos em diversas linguagens de desenvolvimento.

A InterfaceEpsonNF suporta os seguintes modelos de impressoras não fiscais Epson: TM-T81+, TM-T88 IV, TM-T88 V, TM-T20, TM-T20 II, TM-T70, TM-T70II, TM-P60II, TM-P80, TM-P20, TM-H6000 III, TM-H6000 IV.

As interfaces de comunicação suportadas pela InterfaceEpsonNF são: Serial, USB e Ethernet.

3 Retornos das Funções

A tabela abaixo lista os valores dos retornos das funções e seus respectivos significados.

Símbolo	Valor Hexa	Descrição
FUNC_SUCESSO	0x01	Operação realizada com sucesso.
FUNC_ERRO	0x00	Erro durante a execução.

Tabela 3 – Retornos das Funções

4 Funções da Interface

As funções da interface Epson foram definidas utilizando o seguinte protótipo:

```
function    Nome_Função(...)
```

A tabela abaixo define os tipos de dados utilizados como parâmetros nas funções e seus respectivos tamanhos em bits.

Tipo	Descrição	Declaração C/C++	Declaração VB6/VB.Net	Delphi
SHORTINT	16-bit	unsigned short	Short	ShortInt
INTEGER	32-bit	unsigned int	Integer	Integer
BOOLEAN	0 (FALSE) or ≠ 0 (TRUE)	int/bool	Boolean	LongBool

Tabela 4 – Tipos de Dados

A tabela abaixo define os tipos de ponteiros utilizados como parâmetros de retorno de dados nas funções.

Ponteiro	Tipo	Declaração C/C++	Declaração VB6/VB.Net	Delphi
@BOOLEAN	BOOL*	int*/bool *	Boolean	LongBool
PCHAR	char *	char *	String	Pchar / PAnsiChar

Tabela 5 – Tipos de Ponteiros

Por convenção, todas as tabelas que detalham as posições em buffers retornados pela InterfaceEpsonNF utilizam a posição "0" como sendo a posição inicial do mesmo (notação utilizada por linguagens como C/C++,

Delphi e Java). Caso a linguagem de programação utilizada utilize por convenção a posição "1" como sendo a posição inicial de um buffer, todas as posições das tabelas devem ser acrescidas de uma unidade.

4.1 Funções de Impressão

4.1.1 ConfiguraTaxaSerial

Esta função indica para a biblioteca qual deve ser a velocidade de comunicação utilizada para a porta serial.

Sintaxe:

function ConfiguraTaxaSerial (dwTaxa:Integer):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
dwTaxa	INTEGER	-	Velocidade da porta serial 115200 – 57600 – 38400 – 19200 – 9600

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

O canal de comunicação deve estar fechado.

Exemplo em C / C++ / C++ Builder / C#:

```
Retorno = ConfiguraTaxaSerial( 115200 );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ConfiguraTaxaSerial( 115200 )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ConfiguraTaxaSerial( 115200 );
```

4.1.2 IniciaPorta

Esta função abre a porta de comunicação com a impressora não fiscal. A execução bem sucedida desta função é necessária para o funcionamento de todos os demais comandos da interface.

Sintaxe:

Function IniciaPorta (pszPorta:Pchar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszPorta	PCHAR	-	Nome da Porta de Comunicação que será utilizada. Podemos utilizar: COM1, COM2, COM3, etc, para porta serial; USB ou o endereço I.P para Ethernet

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

O canal de comunicação deve estar fechado.

Exemplo em C / C++ / C++ Builder / C#:

```
Retorno = IniciaPorta( "USB" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = IniciaPorta( "COM1" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := IniciaPorta ( '192.168.192.168' );
```

4.1.3 FechaPorta

Esta função efetua o fechamento do canal de comunicação entre a impressora e a biblioteca. Após a execução deste comando será necessário reabrir o canal de comunicação para o envio de novos comandos.

Sintaxe:

function FechaPorta ():Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

O canal de comunicação deve estar aberto.

Exemplo em C / C++ / C++ Builder / C#:

```
Retorno = FechaPorta( );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = FechaPorta( )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := FechaPorta( );
```

4.1.4 ImprimeTexto

Esta função envia uma cadeia de caracteres para ser impresso. A quebra de linhas será efetuada caso seja inserido um caracter do tipo LINE FEED, ou caso seja atingido o limite máximo de caracteres suportados pela linha (limite variável, dependendo do tipo de fonte e impressora utilizada).

Sintaxe:

function ImprimeTexto(pszTexto:Pchar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszTexto	PCHAR	2048	Texto que será impresso

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

O canal de comunicação deve estar aberto.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeTexto("Linha 01\nLinha 02");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno =  ImprimeTexto("Linha 01" +  Chr(10) + "Linha 02")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno :=  ImprimeTexto('Linha 01' + #10 + 'Linha 02');
```

4.1.5 ImprimeTextoTag

Esta função envia uma cadeia de caracteres para ser impresso. A quebra de linhas será efetuada caso seja inserido um caracter do tipo LINE FEED, ou caso seja atingido o limite máximo de caracteres suportados pela linha (limite variável, dependendo do tipo de fonte e impressora utilizada). Nesta função podemos incluir TAGS que podem indicar:

- Formatação de texto
- Impressão de código de barras
- Impressão de códigos bi-dimensionais (QR-Code, PDF 417)
- Impressão de logotipo

Sintaxe:

```
function ImprimeTextoTag(pszTexto:Pchar):Integer; StdCall; External 'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
pszTexto	PCHAR	4096	Texto que será impresso

Tags suportadas:

Tag	Descrição
	Negrito
<ad></ad>	Alinhamento de texto à direita
<s></s>	Sublinhado
<e></e>	Expandido
<c></c>	Condensado
<n></n>	Fonte Padrão
<l></l>	Salta Linha
<ce></ce>	Centraliza o Texto • <ce> deve ser sempre utilizada no início da linha. • </ce> deve ser sempre utilizada antes da quebra de linha.
<da></da>	Altura Dupla
<xl></xl>	Altura e Largura Dupla
<g></g>	Aciona a Gaveta
<gui></gui>	Aciona a Guilhotina
<bmp>XX,XX</bmp>	Imprime logotipo pré carregado na impressora. Caso seja passado como parâmetro entre as tags o índice do logotipo, o mesmo será utilizado na impressão. Caso contrário será utilizada a primeira posição da tabela de logotipos (32,32).

<ibmp>XXX</ibmp>	Imprime uma imagem gravada em um arquivo BMP. Deve ser passado o path completo do arquivo, incluindo a extensão .bmp
<cespl>XX</cespl>	Configura o espaço entre as linhas utilizado pela impressora.
<upc-a> </upc-a>	Imprime Código de Barras padrão UPC-A
<ean13> </ean13>	Imprime Código de Barras padrão EAN-13
<ean8> </ean8>	Imprime Código de Barras padrão EAN-8
<code39> </code39>	Imprime Código de Barras padrão Code 39
<code93> </code93>	Imprime Código de Barras padrão Code 93
<codabar> </codabar>	Imprime Código de Barras padrão CODABAR
<i2of5> </i2of5>	Imprime Código de Barras padrão ITF
<code128> </code128>	Imprime Código de Barras padrão Code 128B
<code128c> </code128c>	Imprime Código de Barras padrão Code 128C
<pdf> </pdf>	Imprime Código Bidimensional padrão PDF 417
<qrcode> </qrcode>	Imprime Código Bidimensional padrão QR-Code
<pmqrcode> </pmqrcode>	Define o page mode do qrcode. Pode variar entre 150 a 255
<pmtextolateral> </pmtextolateral>	Define o page mode do texto lateral. Pode variar entre 150 a 255
<lmodulo> </lmodulo>	Define a largura do módulo utilizada no QR-Code
<correcao> </correcao>	Define o fator de correção utilizado no QR-Code
<texto_qrcode> </texto_qrcode>	Imprime o texto ao lado do QR-Code (as tags de formatação poderão ser utilizadas neste texto).

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum requisito é necessário.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeTextoTag("<ce><qrcode>www.epson.com.br</qrcode></ce>");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno =  ImprimeTextoTag("<ce><qrcode>www.epson.com.br</qrcode></ce>")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno :=  ImprimeTextoTag('<ce><qrcode>www.epson.com.br</qrcode></ce>');
```

4.1.6 FormataTX

Esta função envia uma cadeia de caracteres a ser impressa, com a possibilidade da escolha da formatação que será aplicada no texto.

Sintaxe:

function FormataTX(psxTexto:PChar , dwTipoLetra:Integer, dwItalico:Integer, dwSublinhado:Integer, dwExpandido:Integer, dwEnfatizado:Integer):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszTexto	PCHAR	-	Texto que será impresso
dwTipoLetra	INTEGER	-	Tipo de letra que será utilizada: **1** – Comprimido **2** - Normal
dwItalico	INTEGER	-	Flag que habilita itálico **1** – Habilita itálico **0** – Desabilita itálico
dwSublinhado	INTEGER	-	Flag que habilita sublinhado **1** – Habilita sublinhado **0** – Desabilita sublinhado
dwExpandido	INTEGER	-	Flag que habilita impressão expandida **1** – Habilita expandido **0** – Desabilita expandido
dwEnfatizado	INTEGER	-	Flag que habilita Negrito **1** – Habilita negrito **0** – Desabilita negrito

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = FormataTX( "Teste de impressão", 1, 0, 1, 1, 1 );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno =  FormataTX( "Teste de impressão", 1, 0, 1, 1, 1 )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno :=  FormataTX( 'Teste de impressão', 1, 0, 1, 1, 1 );
```

4.1.7 AcionaGuilhotina

Esta função executa o acionamento da guilhotina da impressora.

Sintaxe:

function AcionaGuilhotina (dwTipoCorte:Integer):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
dwTipoCorte	INT	-	Tipo do corte utilizado: **1** – Total

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = AcionaGuilhotina( 1 );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = AcionaGuilhotina( 1 )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := AcionaGuilhotina( 1 );
```

4.1.8 ComandoTX

Esta função deve ser utilizada para o envio de cadeias de bytes, contendo comandos do tipo ESC/POS.

Sintaxe:

function ComandoTX(pszComando:PChar; dwTamanho:Integer):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszComando	PCHAR	-	Cadeia de bytes do comando.
dwTamanho	INT	-	Quantidade de bytes utilizados no comando.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
char szComando[] = { 27, 112, 0, 25, 250}; //Aciona gaveta
Retorno = ComandoTX( szComando, 5 );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
szComando = Chr(27) + Chr(112) + Chr(0) + Chr(25) + Chr(250)
Retorno = ComandoTX( szComando, 5 )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
szComando := chr(#27) +  chr(#112) +  chr(#0) +  chr(#25) +  chr(#250);
Retorno := ComandoTX( szComando, 5 );
```

4.1.9 Le_Status

Esta função retorna o status da impressora.

Sintaxe:

function Le_Status():Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Nenhum

Retornos:

FUNC_ERRO	Erro durante a execução.
5	Impressora com pouco papel.
9	Tampa aberta.
24	Impressora “ONLINE”.
32	Impressora sem papel.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C#:

```
Retorno = Le_Status();
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = Le_Status()
```

Exemplo em Delphi/ Delphi XE2:

```
Retorno := Le_Status();
```

4.1.10 Le_Status_Gaveta

Esta função efetua a leitura do status da gaveta.

Sintaxe:

function Le_Status_Gaveta ():Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum

Saídas:

Nenhum.

Retornos:

FUNC_ERRO	Erro durante a execução.
1	Gaveta Aberta.
2	Gaveta Fechada.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = Le_Status_Gaveta();
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = Le_Status_Gaveta()
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := Le_Status_Gaveta();
```

4.1.11 ConfiguraCodigoBarras

Esta função efetua a configuração dos parâmetros utilizados na impressão do código de barras.

Sintaxe:

function ConfiguraCodigoBarras (dwAltura:Integer, dwLargura:Integer, dwHRI:Integer, dwFonte:Integer, dwMargem:Integer):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
dwAltura	INTEGER	-	Altura do código de barras (entre 1 e 255).
dwLargura	INTEGER	-	Largura da barra utilizada: **0** – Barras finas. **1** – Barras médias. **2** – Barras grossas.
dwHRI	INTEGER	-	Posição dos caracteres do código de barras: **0** – Não imprime os caracteres. **1** – Imprime os caracteres acima do código **2** – Imprime os caracteres abaixo do código **3** – Imprime os caracteres acima e abaixo do código
dwFonte	INTEGER	-	Tipo da fonte utilizada na impressão dos caracteres: **0** – Normal. **1** – Condensada.
dwMargem	INTEGER	-	Valor da margem utilizada (entre 0 e 575).

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ConfiguraCodigoBarras( 150, 0, 2, 1, 0 );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ConfiguraCodigoBarras( 150, 0, 2, 1, 0 )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ConfiguraCodigoBarras( 150, 0, 2, 1, 0 );
```

4.1.12 ImprimeCodigoBarrasCODABAR

Esta função efetua a impressão de um código de barras do tipo CODABAR.

Sintaxe:

function ImprimeCodigoBarrasCODABAR(pszCodigo:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

A função ConfiguraCodigoBarras deve ter sido executada.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasCODABAR( "A1234567B" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasCODABAR( "A1234567B")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasCODABAR( 'A1234567B');
```

4.1.13 ImprimeCodigoBarrasCODE128

Esta função efetua a impressão de um código de barras do tipo CODE128.

Sintaxe:

function ImprimeCodigoBarrasCODE128 (pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

A função ConfiguraCodigoBarras deve ter sido executada.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasCODE128( "Linha de Texto" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasCODE128( "Linha de Texto" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasCODE128( 'Linha de Texto' );
```

4.1.14 ImprimeCodigoBarrasCODE39

Esta função efetua a impressão de um código de barras do tipo CODE39.

Sintaxe:

function ImprimeCodigoBarrasCODE39(pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

A função ConfiguraCodigoBarras deve ter sido executada.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasCODE39("ABC-123");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasCODE39("ABC-123")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasCODE39('ABC-123');
```

4.1.15 ImprimeCodigoBarrasCODE93

Esta função efetua a impressão de um código de barras do tipo CODE93.

Sintaxe:

function ImprimeCodigoBarrasCODE93(pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

A função ConfiguraCodigoBarras deve ter sido executada.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasCODE93("123-ABC");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasCODE93("123-ABC")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasCODE93('123-ABC');
```

4.1.16 ImprimeCodigoBarrasEAN13

Esta função efetua a impressão de um código de barras do tipo EAN13.

Sintaxe:

function ImprimeCodigoBarrasEAN13(pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

A função ConfiguraCodigoBarras deve ter sido executada.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasEAN13("123456789012");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasEAN13("123456789012")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasEAN13('123456789012');
```

4.1.17 ImprimeCodigoBarrasEAN8

Esta função efetua a impressão de um código de barras do tipo EAN8.

Sintaxe:

function ImprimeCodigoBarrasEAN8(pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

A função ConfiguraCodigoBarras deve ter sido executada.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasEAN8("1234567");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasEAN8("1234567")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasEAN8('1234567');
```

4.1.18 ImprimeCodigoBarrasITF

Esta função efetua a impressão de um código de barras do tipo ITF.

Sintaxe:

function ImprimeCodigoBarrasITF(pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

A função ConfiguraCodigoBarras deve ter sido executada.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasITF("0123456789012345");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasITF("0123456789012345")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasITF('0123456789012345');
```

4.1.19 ImprimeCodigoBarrasUPCA

Esta função efetua a impressão de um código de barras do tipo UPC-A.

Sintaxe:

function ImprimeCodigoBarrasUPCA(pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

A função ConfiguraCodigoBarras deve ter sido executada.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasUPCA("12345678901");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasUPCA("12345678901")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasUPCA('12345678901');
```

4.1.20 ImprimeCodigoBarrasUPCE

Esta função efetua a impressão de um código de barras do tipo UPC-E.

Sintaxe:

function ImprimeCodigoBarrasUPCE(pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

A função ConfiguraCodigoBarras deve ter sido executada.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasUPCE("02354866");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasUPCE("02354866")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasUPCE('02354866');
```

4.1.21 ImprimeCodigoBarrasPDF417

Esta função efetua a impressão de um código bi-dimensional do tipo PDF-417.

Sintaxe:

function ImprimeCodigoBarrasPDF417(dwCorrecao:Integer, dwAltura:Integer, dwLargura:Integer, dwColunas:Integer, pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
dwCorrecao	INTEGER	-	Nível de correção de erros (entre 0 e 8).
dwAltura	INTEGER	-	Altura do caracter (entre 1 e 8).
dwLargura	INTEGER	-	Largura do caracter (entre 1 e 4).
dwColunas	INTEGER	-	Número de colunas utilizadas por linha (entre 0 e 30).
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoBarrasPDF417(4, 3, 2, 0, "Código PDF 417");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoBarrasPDF417(4, 3, 2, 0, "Código PDF 417")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoBarrasPDF417(4, 3, 2, 0, 'Código PDF 417');
```

4.1.22 ImprimeCodigoQRCODE

Esta função efetua a impressão de um código bi-dimensional do tipo QR-Code.

Sintaxe:

function ImprimeCodigoQRCODE(dwRestauracao:Integer, dwModulo:Integer, dwTipo:Integer, dwVersao:Integer, dwModo:Integer, pszCodigo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
dwRestauracao	INTEGER	-	Nível de restauração do código: **0** – 7% de restauração **1** – 15% de restauração **2** – 25% de restauração **3** – 30% de restauração
dwModulo	INTEGER	-	Tamanho do módulo do código (entre 1 e 127).
dwTipo	INTEGER	-	Tipo do código: **0** – Normal **1** – Reduzido
dwVersao	INTEGER	-	Versão do QR-Code (entre 1 e 40).
dwModo	INTEGER	-	Tipo dos dados que serão impressos: **0** – Somente números **1** – Alfanumérico **2** – Binário **3** – Kanji
pszCodigo	PCHAR	-	Código que será impresso.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCodigoQRCODE(3, 3, 1, 1, 1, "Código QR CODE");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCodigoQRCODE(3, 3, 1, 1, 1, "Código  QR CODE ")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := ImprimeCodigoQRCODE(3, 3, 1, 1, 1, 'Código  QR CODE');
```

4.1.23 GerarQRCodeArquivo

Esta função efetua a criação de um arquivo BMP com um código bi-dimensional do tipo QR-Code.

Sintaxe:

function GerarQRCodeArquivo(pszFileName:PChar, pszDados:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszFileName	PCHAR	-	Nome do arquivo de destino (Path completo, incluindo a extensão .BMP)
pszDados	PCHAR	-	Dados que serão gravados no QRCode.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno =  GerarQRCodeArquivo("C:\\Epson\\QRCode.bmp", "Código QR CODE");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno =  GerarQRCodeArquivo("C:\\Epson\\QRCode.bmp", "Código QR CODE")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno :=  GerarQRCodeArquivo('C:\\Epson\\QRCode.bmp', 'Código QR CODE');
```

4.1.24 ImprimeBmpEspecial

Esta função efetua a impressão de um arquivo BMP monocromático na impressora.

Sintaxe:

function ImprimeBmpEspecial(pszFileName:PChar, dwX:Integer, dwY:Integer, dwAngulo:Integer,):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszFileName	PCHAR	-	Nome do arquivo de destino (Path completo, incluindo a extensão .BMP)
dwX	INTEGER	-	Escala Horizontal (não implementado).
dwY	INTEGER	-	Escala Vertical (não implementado).
dwAngulo	INTEGER	-	Ângulo de impressão (não implementado).

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno =  ImprimeBmpEspecial("C:\\Epson\\QRCode.bmp", 0,0,0);
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno =  ImprimeBmpEspecial("C:\\Epson\\QRCode.bmp", 0,0,0)
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno :=  ImprimeBmpEspecial('C:\\Epson\\QRCode.bmp', 0,0,0);
```

4.1.25 Habilita_Log

Esta função habilita o log da comunicação efetuada entre a biblioteca e a impressora.

Sintaxe:

function Habilita_Log(dwEstado:Integer, pszCaminho:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
dwEstado	INTEGER	-	Estado de execução do Log: **0** – Desabilitado **1** – Habilitado
pszCaminho	PCHAR	-	Nome do diretório de gravação do log.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno =  Habilita_Log(1, "C:\\Epson\\");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = Habilita_Log( 1, "C:\\Epson\\" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
iRetorno := Habilita_Log( 1, 'C:\\Epson\\');
```

4.1.26 ImprimeCheque

Esta função executa a impressão de um cheque. Comando disponível apenas para modelos TM-H6000.

Sintaxe:

function ImprimeCheque(szIndice:PChar, szValor:PChar, szData:PChar, szPara:PChar, szCidade:PChar, szAdicional:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	-	Índice das coordenadas do banco que serão utilizadas na impressão do cheque.
szValor	PCHAR	-	Valor do cheque, com 2 casas decimais.
szData	PCHAR	-	Data do cheque, no formato DD/MM/AAAA
szPara	PCHAR	-	Destinatário do cheque
szCidade	PCHAR	-	Cidade em que o cheque será emitido.
szAdicional	PCHAR	-	Texto adicional que será impresso no cheque.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
-1	Timeout de inserção do cheque.
-2	Indice do cheque não encontrado no arquivo InterfaceEpsonNF.xml
-3	Coordenada do cheque com valor zerado.
-4	Arquivo InterfaceEpsonNF.xml não encontrado
-5	Data do cheque com formato inválido.
-6	Mês com valor inválido.

Requisitos:

As coordenadas do cheque devem estar gravadas no arquivo InterfaceEpson.xml. Para maiores informações, verifique o capitulo 5 (Layout do cheque).

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeCheque("1", "12,23", "10/12/2014", "Epson do Brasil","São
Paulo", "");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeCheque("1", "12,23", "10/12/2014", "Epson do Brasil","São
Paulo", "")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
iRetorno :=  ImprimeCheque('1', '12,23', '10/12/2014', 'Epson do
Brasil','São Paulo', '');
```

4.1.27 ImprimeAutenticacao

Esta função executa a impressão de uma autenticação no verso de um cheque. Comando disponível apenas para modelos TM-H6000.

Sintaxe:

function ImprimeAutenticacao(pszPosX:PChar, pszPosY:PChar, pszLinhaTexto:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
pszPosX	PCHAR	3	Posição inicial do texto na horizontal.
pszPosY	PCHAR	2	Posição inicial do texto na vertical.
pszLinhaTexto	PCHAR	128	Texto da autenticação

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
-1	Timeout de inserção do cheque.

Requisitos:

As coordenadas da autenticação devem seguir o mesmo padrão de orientação de impressão do cheque. Para maiores informações, verifique o capitulo 5 (Layout do cheque).

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = ImprimeAutenticacao("120", "60", "Epson do Brasil");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = ImprimeAutenticacao("120", "60" "Epson do Brasil")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
iRetorno := ImprimeAutenticacao('120', '60', 'Epson do Brasil');
```

4.1.28 LeMICR

Esta função executa a leitura do código MICR do cheque. Comando disponível apenas para modelos TM-H6000.

Sintaxe:

function LeMICR(pszCodigo:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	-	Código do MICR lido do cheque.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
-1	Timeout de inserção do cheque.
-2	Erro de leitura do MICR.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder:

```
char pszCodigo[40];
Retorno = LeMICR( pszCodigo );
```

Exemplo em C#:

```
StringBuilder pszCodigo = new StringBuilder(40, 40);
Retorno = LeMICR( pszCodigo )
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Dim pszCodigo As String
pszCodigo = Space(40)
iRetorno := LeMICR( pszCodigo )
```

Exemplo em Delphi 7:

```
pszCodigo: array[0..40] of Char;
iRetorno := LeMICR( pszCodigo );
```

Exemplo em Delphi 2010 / Delphi XE:

```
pszCodigo := AnsiString(StringOfChar(' ', 40));
iRetorno := LeMICR( pszCodigo );
```

4.1.29 LeModelo

Esta função executa a leitura do modelo da impressora que está sendo utilizada.

Sintaxe:

function LeModelo(pszModelo:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Variável	Tipo	Tam.	Descrição
pszModelo	PCHAR	32	Modelo da impressora utilizada.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder:

```
char pszModelo[32];
Retorno = LeModelo(pszModelo);
```

Exemplo em C#:

```
StringBuilder pszModelo = new StringBuilder(32, 32);
Retorno = LeModelo( pszModelo )
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Dim pszModelo As String
pszModelo = Space(32)
iRetorno := LeModelo( pszModelo )
```

Exemplo em Delphi 7:

```
pszModelo: array[0..32] of Char;
iRetorno := LeModelo( pszModelo );
```

Exemplo em Delphi 2010 / Delphi XE:

```
pszModelo := AnsiString(StringOfChar(' ', 32));
iRetorno :=  LeModelo( pszModelo );
```

4.1.30 LeNumeroSerie

Esta função executa a leitura do número de série da impressora que está sendo utilizada.

Sintaxe:

function LeNumeroSerie(pszSerie:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Variável	Tipo	Tam.	Descrição
pszSerie	PCHAR	32	Numero de série da impressora utilizada.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder:

```
char pszSerie[32];
Retorno =  LeNumeroSerie( pszSerie );
```

Exemplo em C#:

```
StringBuilder pszSerie = new StringBuilder(32, 32);
Retorno = LeNumeroSerie(  pszSerie )
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Dim pszSerie As String
pszSerie = Space(32)
iRetorno := LeNumeroSerie( pszSerie )
```

Exemplo em Delphi 7:

```
pszSerie : array[0..32] of Char;
iRetorno :=  LeNumeroSerie( pszSerie );
```

Exemplo em Delphi 2010 / Delphi XE:

```
pszModelo := AnsiString(StringOfChar(' ', 32));
iRetorno :=  LeModelo( pszModelo );
```

4.1.31 Le_Status_Slip

Esta função executa a leitura do status dos sensores BOF e TOF da estação de cheque.

Sintaxe:

function Le_Status_Slip(pszFlags:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Variável	Tipo	Tam.	Descrição
pszFlags	PCHAR	4	Flas de status dos sensores.

Os dados serão retornados no seguinte formato:

Byte 0 :	Impressora aguardando inserção de papel (**1** – Sim, **0** – Não)
Byte 1 :	Sensor TOF detectando papel (**1** – Sim, **0** - Não)
Byte 2 :	Sensor BOF detectando papel (**1** – Sim, **0** - Não)

Todos os campos têm tamanho fixo.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder:

```
char pszFlags[4];
Retorno = Le_Status_Slip( pszFlags);
```

Exemplo em C#:

```
StringBuilder pszFlags = new StringBuilder(4, 4);
Retorno = Le_Status_Slip( pszFlags )
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Dim pszFlags As String
pszFlags = Space(4)
iRetorno := Le_Status_Slip( pszFlags )
```

Exemplo em Delphi 7:

```
pszFlags: array[0..4] of Char;
iRetorno := Le_Status_Slip( pszFlags );
```

Exemplo em Delphi 2010 / Delphi XE:

```
pszFlags := AnsiString(StringOfChar(' ', 4));
iRetorno := Le_Status_Slip( pszFlags );
```

4.1.32 AcionaGaveta

Esta função efetua o acionamento da gaveta de dinheiro conectada na impressora.

Sintaxe:

function AcionaGaveta():Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

O canal de comunicação deve estar aberto.

Exemplo em C / C++ / C++ Builder / C#:

```
Retorno = AcionaGaveta( );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = AcionaGaveta( )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := AcionaGaveta( );
```

4.1.33 EPSON_NFCe_Imprimir

Esta função efetua a impressão de um Danfe de NFC-e a partir de um XML.

Sintaxe:

function EPSON_NFCe_Imprimir(szXML:PChar, szTipo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szXML	PCHAR	1024	Caminho completo do arquivo XML.
szTipo	PCHAR	1	Flag que indica o tipo da Danfe que será impressa: **C** – Completa **S** – Simplificada

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

O canal de comunicação deve estar aberto.

As tags <MODELO>, <UrlFisco>, <cIdToken> e <CSC> devem estar configuradas no arquivo InterfaceEpsonNF.xml.

Para maiores informações, consulte o capítulo 9 (Configuração para impressão de Danfe NFC-e e Extrato do SAT-CFE a partir de um XML).

Exemplo em C / C++ / C++ Builder / C#:

```
Retorno = EPSON_NFCe_Imprimir("C:\\Temp\\nfce.xml", "C" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Imprimir("C:\\Temp\\nfce.xml", "C")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Imprimir('C:\\Temp\\nfce.xml', 'C');
```

4.1.34 EPSON_SAT_Imprimir

Esta função efetua a impressão de um Extrato de venda do SAT-CFE a partir de um XML.

Sintaxe:

function EPSON_SAT_Imprimir(szXML:PChar, szTipo:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szXML	PCHAR	1024	Caminho completo do arquivo XML.
szTipo	PCHAR	1	Flag que indica o tipo do extrato que será impresso: **C** – Completo **S** – Simplificado

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

O canal de comunicação deve estar aberto.
A tag <MODELO> deve estar configurada no arquivo InterfaceEpsonNF.xml.
Para maiores informações, consulte o capítulo 9 (Configuração para impressão de Danfe NFC-e e Extrato do SAT-CFE a partir de um XML).

Exemplo em C / C++ / C++ Builder / C#:

```
Retorno = EPSON_SAT_Imprimir("C:\\Temp\\sat.xml", "C");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_SAT_Imprimir("C:\\Temp\\sat.xml", "C")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_SAT_Imprimir('C:\\Temp\\sat.xml', 'C');
```

4.1.35 EPSON_SAT_Imprimir_Cancelamento

Esta função efetua a impressão de um Extrato de cancelamento de venda do SAT-CFE a partir de um XML.

Sintaxe:

function EPSON_SAT_Imprimir_Cancelamento(szXML:PChar, szQRCodeVenda:PChar, szTeste:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szXML	PCHAR	1024	Caminho completo do arquivo XML.
szQRCodeVenda	PCHAR	344	Campo <szassinaturaQRCODE> do XML de Venda que será cancelado.
szTeste	PCHAR	1	Flag que indica se o ambiente utilizado para envio do cancelamento será de testes ou produção: **S** – Sim (testes). **N** – Não (produção).

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

O canal de comunicação deve estar aberto.
A tag <MODELO> deve estar configurada no arquivo InterfaceEpsonNF.xml.
Para maiores informações, consulte o capítulo 9 (Configuração para impressão de Danfe NFC-e e Extrato do SAT-CFE a partir de um XML).

Exemplo em C / C++ / C++ Builder / C#:

```
Retorno = EPSON_SAT_Imprimir_Cancelamento("C:\\Temp\\canc_sat.xml",
"ELJl/osQTnFbCXTh9qvb9885yiRGMRSdS0Ll+BRYzDCDO3UjJ/7ddd9XJn7buzUKD4QDnHCqt
3IVSYOd8Z7zuQwiigpsU3nlgAbM7/kxIKrI5YExPdxqY6MlJmhKhT6ubUWHod8lPmANX+ayAVr
x+6DauqBYdz6n0GSGUxzhvwfBPESnBW6mFroMKAfRkd6zmLxf6t+KppqNf/sy8Bx103/sAzzZe
v+5lJR1lENdEUjobGnZyHeacWt7DWEttLGnKhNMtrTTyfTMHQzSi6Y1KzDH93yyeEqHIjfPKAA
AFmBQDcGve8rBA83ZBhB7u2cemn8UKSCVPpK1MzvBythGmg==", "S");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_SAT_Imprimir_Cancelamento("C:\\Temp\\canc_sat.xml",
"ELJl/osQTnFbCXTh9qvb9885yiRGMRSdS0Ll+BRYzDCDO3UjJ/7ddd9XJn7buzUKD4QDnHCqt
3IVSYOd8Z7zuQwiigpsU3nlgAbM7/kxIKrI5YExPdxqY6MlJmhKhT6ubUWHod8lPmANX+ayAVr
x+6DauqBYdz6n0GSGUxzhvwfBPESnBW6mFroMKAfRkd6zmLxf6t+KppqNf/sy8Bx103/sAzzZe
v+5lJR1lENdEUjobGnZyHeacWt7DWEttLGnKhNMtrTTyfTMHQzSi6Y1KzDH93yyeEqHIjfPKAA
AFmBQDcGve8rBA83ZBhB7u2cemn8UKSCVPpK1MzvBythGmg==", "S")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_SAT_Imprimir_Cancelamento('C:\\Temp\\canc_sat.xml',
'ELJl/osQTnFbCXTh9qvb9885yiRGMRSdS0Ll+BRYzDCDO3UjJ/7ddd9XJn7buzUKD4QDnHCqt
3IVSYOd8Z7zuQwiigpsU3nlgAbM7/kxIKrI5YExPdxqY6MlJmhKhT6ubUWHod8lPmANX+ayAVr
x+6DauqBYdz6n0GSGUxzhvwfBPESnBW6mFroMKAfRkd6zmLxf6t+KppqNf/sy8Bx103/sAzzZe
v+5lJR1lENdEUjobGnZyHeacWt7DWEttLGnKhNMtrTTyfTMHQzSi6Y1KzDH93yyeEqHIjfPKAA
AFmBQDcGve8rBA83ZBhB7u2cemn8UKSCVPpK1MzvBythGmg==', 'S');
```

4.2 Funções de Emissão de NFC-e

Antes de emitir uma NFC-e será necessário configurar qual será o provedor de messageria de documentos que será utilizado na integração com a InterfaceEpsonNF. Para maiores informações, verifique o capítulo 6 deste manual.

4.2.1 EPSON_NFCe_Abrir

Esta função efetua a abertura de uma NFC-e.

Sintaxe:

```
function EPSON_NFCe_Abrir (szCPF:PChar; szNome:PChar; szEndereco:PChar; szNumero:PChar;
                           szBairro:PChar; szCodMunicipio:PChar; szMunicipio:PChar; szUF:PChar;
                           szCep:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
szCPF	PCHAR	20	CPF ou CNPJ do consumidor.
szNome	PCHAR	60	Nome do consumidor.
szEndereco	PCHAR	60	Endereço do consumidor.
szNumero	PCHAR	60	Número do endereço.
szBairro	PCHAR	60	Bairro do endereço.
szCodMunicipio	PCHAR	7	Código do município do endereço (Tabela IBGE).
szMunicipio	PCHAR	60	Município do endereço.
szUF	PCHAR	2	Estado do endereço.
szCep	PCHAR	8	Cep do endereço.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / C++ / C++ Builder / C#:

```
Retorno = EPSON_NFCe_Abrir( "", "", "", "", "", "", "", "", "" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Abrir( "", "", "", "", "", "", "", "", "" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Abrir( '', '', '', '', '', '', '', '', '' );
```

4.2.2 EPSON_NFCe_AbrirEX

Esta função efetua a abertura de uma NFC-e. Utilizando esta função, o controle do número e da série da NFC-e ficam a cargo da aplicação.

Sintaxe:

```
function EPSON_NFCe_AbrirEX (szCPF:PChar; szNome:PChar; szEndereco:PChar; szNumero:PChar;
                             szBairro:PChar; szCodMunicipio:PChar; szMunicipio:PChar; szUF:PChar;
                             szCep:PChar; szNumero:PChar; szSerie:PChar):Integer; StdCall; External
                             'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
szCPF	PCHAR	20	CPF ou CNPJ do consumidor.
szNome	PCHAR	60	Nome do consumidor.
szEndereco	PCHAR	60	Endereço do consumidor.
szNumero	PCHAR	60	Número do endereço.
szBairro	PCHAR	60	Bairro do endereço.
szCodMunicipio	PCHAR	7	Código do município do endereço (Tabela IBGE).
szMunicipio	PCHAR	60	Município do endereço.
szUF	PCHAR	2	Estado do endereço.
szCep	PCHAR	8	Cep do endereço.
szNumero	PCHAR	9	Número da NFC-e que será iniciada
szSerie	PCHAR	3	Série da NFC-e que será iniciada.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_AbrirEX( "","","","","","","","","","100","1" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_AbrirEX( "","","","","","","","","","100","1" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_AbrirEX('','','','','','','','','','100','1' );
```

4.2.3 EPSON_NFCe_Vender_Item

Esta função efetua a venda de um item em uma NFC-e. O produto deve estar cadastrado no arquivo "produtos.xml". Para maiores informações a respeito deste arquivo, consulte o capítulo 7.

Sintaxe:

```
function EPSON_NFCe_Vender_Item(szCodigo:PChar; szDescricao:PChar; szQuantidade:PChar;
                                szCasasDecimaisQuantidade:PChar; szUnidadeDeMedida:PChar;
                                szPrecoUnidade:PChar; szCasasDecimaisPreco:PChar;
                                szAliquotas:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
szCodigo	PCHAR	60	Código do item.
szDescricao	PCHAR	120	Descrição do item.
szQuantidade	PCHAR	15	Quantidade do item.
szCasasDecimaisQuantidade	PCHAR	2	Número de casas decimais do valor informado no campo quantidade.
szUnidadeDeMedida	PCHAR	6	Símbolo da unidade.
szPrecoUnidade	PCHAR	15	Preço unitário.
szCasasDecimaisPreco	PCHAR	2	Número de casas decimais do valor informado no campo preço.
szAliquotas	PCHAR	4	Valor da alíquota em percentual com 2 casas decimais. (Exemplos: 1000, 1250, 0500).

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Vender_Item( "78912341234", "Produto XYZ", "1000",
"2", "UNI", "100", "2", "1800" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Vender_Item( "78912341234", "Produto XYZ", "1000",
"2", "UNI", "100", "2", "1800" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Vender_Item( '78912341234', 'Produto XYZ', '1000',
'2', 'UNI', '100', '2', '1800' );
```

4.2.4 EPSON_NFCe_Vender_ItemEX

Esta função efetua a venda de um item em uma NFC-e.

Sintaxe:

function EPSON_NFCe_Vender_ItemEX(szCodigo:PChar; szDescricao:PChar; szQuantidade:PChar; szCasasDecimaisQuantidade:PChar; szUnidadeDeMedida:PChar; szPrecoUnidade:PChar; szCasasDecimaisPreco:PChar; szAliquotas:PChar; szNCM: PChar; szCST: PChar; szPIS: PChar; szCOFINS: PChar; szCFOP: PChar; szCEST: PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szCodigo	PCHAR	60	Código do item.
szDescricao	PCHAR	120	Descrição do item.
szQuantidade	PCHAR	15	Quantidade do item.
szCasasDecimaisQuantidade	PCHAR	2	Número de casas decimais do valor informado no campo quantidade.
szUnidadeDeMedida	PCHAR	6	Símbolo da unidade.
szPrecoUnidade	PCHAR	15	Preço unitário.
szCasasDecimaisPreco	PCHAR	2	Número de casas decimais do valor informado no campo preço.
szAliquotas	PCHAR	4	Valor da alíquota em percentual com 2 casas decimais. (Exemplos: 1000, 1250, 0500).
szNCM	PCHAR	8	Número do NCM do item.
szCST	PCHAR	2	CST de PIS/COFINS do item.
szPIS	PCHAR	4	Alíquota de PIS do Item.
szCOFINS	PCHAR	4	Alíquota de COFINS do item.
szCFOP	PCHAR	4	CFOP do item.
szCEST	PCHAR	7	CEST do item.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Vender_ItemEX( "78912341234", "Produto XYZ", "1000",
"2", "UNI", "100", "2", "1800", "12", "00", "1000", "1200", "1201",
"0100100" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Vender_ItemEX( "78912341234", "Produto XYZ", "1000",
"2", "UNI", "100", "2", "1800", "12", "00", "1000", "1200", "1201",
"0100100" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Vender_ItemEX( '78912341234', 'Produto XYZ', '1000',
'2', 'UNI', '100', '2', '1800', '12', '00', '1000', '1200', '1201',
'0100100' );
```

4.2.5 EPSON_NFCe_Desconto_Acrescimo_Item

Esta função efetua um desconto ou acréscimo em um item vendido na NFC-e.

Sintaxe:

```
function EPSON_NFCe_Desconto_Acrescimo_Item (szNumeroItem: PChar; szOperacao: PChar; szTipo: PChar;
                                             szValor: PChar; szCasasDecimaisValor: PChar):Integer;
                                             StdCall; External 'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
szNumeroItem	PCHAR	3	Número do item registrado na NFC-e.
szOperacao	PCHAR	1	Operação que será realizada: **0** – Desconto. **1** – Acréscimo.
szTipo	PCHAR	1	Tipo da operação: **0** – Percentual. **1** – Valor.
szValor	PCHAR	15	Valor da operação.
szCasasDecimaisValor	PCHAR	2	Casas decimais utilizadas no valor da operação.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.
O item selecionado deve ter sido registrado na NFC-e.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Desconto_Acrescimo_Item("1", "0", "0", "1000", "2");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Desconto_Acrescimo_Item("1", "0", "0", "1000", "2")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Desconto_Acrescimo_Item('1', '0', '0', '1000', '2');
```

4.2.6 EPSON_NFCe_Cancelar_Item

Esta função efetua o cancelamento de um item registrado na NFC-e.

Sintaxe:

function EPSON_NFCe_Cancelar_Item(szNumeroItem:PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szNumeroItem	PCHAR	3	Número do item.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.
O item selecionado deve ter sido registrado na NFC-e.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Cancelar_Item("1");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Cancelar_Item("1")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Cancelar_Item('1');
```

4.2.7 EPSON_NFCe_Cancelar_Item_Parcial

Esta função efetua o cancelamento parcial de um item registrado na NFC-e.

Sintaxe:

```
function EPSON_NFCe_Cancelar_Item_Parcial(szNumeroItem:Pchar; szQuantidade:Pchar;
                                          szCasasDecimaisQuantidade:PChar ):Integer; StdCall; External
                                          'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
szNumeroItem	PCHAR	3	Número do item.
szQuantidade	PCHAR	15	Quantidade que será cancelada.
szCasasDecimaisQuantidade	PCHAR	2	Casas decimais utilizadas na quantidade.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.
O item selecionado deve ter sido registrado na NFC-e.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Cancelar_Item_Parcial("1", "100", "2");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Cancelar_Item_Parcial("1", "100", "2")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Cancelar_Item_Parcial('1', '100', '2');
```

4.2.8 EPSON_NFCe_Cancelar_Desconto_Acrescimo_Item

Esta função efetua o cancelamento de um desconto ou acréscimo de um item registrado na NFC-e.

Sintaxe:

function EPSON_NFCe_Cancelar_Desconto_Acrescimo_Item(szNumeroItem:Pchar; szOperacao:Pchar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szNumeroItem	PCHAR	3	Número do item.
szOperacao	PCHAR	1	Operação que será cancelada: **0** – Desconto. **1** – Acréscimo.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.
O item selecionado deve ter sido registrado na NFC-e.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Cancelar_Desconto_Acrescimo_Item("1", "0");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Cancelar_Desconto_Acrescimo_Item("1", "0")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Cancelar_Desconto_Acrescimo_Item('1', '0');
```

4.2.9 EPSON_NFCe_Pagamento

Esta função efetua o registro de um meio de pagamento na NFC-e.

Sintaxe:

```
function EPSON_NFCe_Pagamento(szFormaPagamento:Pchar; szValor:Pchar;
                              szCasasDecimaisValor:Pchar):Integer; StdCall; External
                              'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
szFormaPagamento	PCHAR	3	Forma de pagamento que será registrada: **1** – Dinheiro. **2** – Cheque. **3** – Cartão de Crédito. **4** – Cartão de Débito. **5** – Crédito Loja. **10** – Vale Alimentação. **11** – Vale Refeição. **12** – Vale Presente. **13** – Vale Combustível. **99** – Outros.
szValor	PCHAR	15	Valor da forma de pagamento.
szCasasDecimaisValor	PCHAR	2	Número de casas decimais utilizadas no valor.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Pagamento("1", "1000", "2");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Pagamento("1", "1000", "2")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Pagamento('1', '1000', '2');
```

4.2.10 EPSON_NFCe_Estorno_Pagamento

Esta função efetua o estorno de um meio de pagamento na NFC-e.

Sintaxe:

```
function EPSON_NFCe_Estorno_Pagamento(szEstornada:Pchar; szEfetivada:Pchar; szValor:Pchar;
                                      szCasasDecimaisValor:Pchar):Integer; StdCall; External
                                      'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
szEstornada	PCHAR	3	Forma de pagamento que será estornada.
szEfetivada	PCHAR	3	Forma de pagamento que será registrada.
szValor	PCHAR	15	Valor da forma de pagamento.
szCasasDecimaisValor	PCHAR	2	Número de casas decimais utilizadas no valor.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Estorno_Pagamento("1", "2", "1000", "2");
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Estorno_Pagamento("1", "2", "1000", "2")
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Estorno_Pagamento('1', '2', '1000', '2');
```

4.2.11 EPSON_NFCe_Dados_Consumidor

Esta função efetua a inclusão dos dados do consumidor na NFC-e.

Sintaxe:

```
function EPSON_NFCe_Dados_Consumidor( szCPF: PChar; szNome: PChar; szEndereco: PChar; szNumero:
                                      PChar; szBairro: PChar; szCodMunicipio: PChar; szMunicipio:
                                      PChar; szUF: PChar; szCep: PChar; szEmail: PChar ):Integer;
                                      StdCall; External 'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
szCPF	PCHAR	20	CPF ou CNPJ do consumidor.
szNome	PCHAR	60	Nome do consumidor.
szEndereco	PCHAR	60	Endereço do consumidor.
szNumero	PCHAR	60	Número do endereço.
szBairro	PCHAR	60	Bairro do endereço.
szCodMunicipio	PCHAR	7	Código do município do endereço (Tabela IBGE).
szMunicipio	PCHAR	60	Município do endereço.
szUF	PCHAR	2	Estado do endereço.
szCep	PCHAR	8	Cep do endereço.
szEmail	PCHAR	60	Email do consumidor.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Dados_Consumidor( "52106911000100", "Epson do
Brasil", "Avenida Tucunare", "720", "Tambore", "3505708", "Barueri", "SP",
"06460020", "" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Dados_Consumidor( "52106911000100", "Epson do
Brasil", "Avenida Tucunare", "720", "Tambore", "3505708", "Barueri", "SP",
"06460020", "" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Dados_Consumidor('52106911000100', 'Epson do
Brasil', 'Avenida Tucunare', '720', 'Tambore', '3505708', 'Barueri', 'SP',
'06460020', '');
```

4.2.12 EPSON_NFCe_Dados_Lei_Imposto

Esta função efetua a inclusão dos dados da Lei "De olho no imposto" na NFC-e.

Sintaxe:

function EPSON_NFCe_Dados_Lei_Imposto (szValorMunicipal: PChar; szPercMun: PChar; szValorEstadual: PChar; szPercEst: PChar; szValorFederal: PChar; szPercFed: PChar; szUF: PChar; szTabela: PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szValorMunicipal	PCHAR	15	Valor total do imposto municipal (com 2 casas decimais).
szPercMun	PCHAR	5	Percentual total do imposto municipal (com 2 casas decimais).
szValorEstadual	PCHAR	15	Valor total do imposto estadual (com 2 casas decimais).
szPercEst	PCHAR	5	Percentual total do imposto estadual (com 2 casas decimais).
szValorFederal	PCHAR	15	Valor total do imposto federal (com 2 casas decimais).
szPercFed	PCHAR	5	Percentual total do imposto federal (com 2 casas decimais).
szUF	PCHAR	2	Unidade federal do estabelecimento.
szTabela	PCHAR	10	Versão da tabela IBPT utilizada.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Dados_Lei_Imposto( "420", "42", "1700", "170",
"1200", "120", "SP", "14.2.a" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Dados_Lei_Imposto( "420", "42", "1700", "170",
"1200", "120", "SP", "14.2.a" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Dados_Lei_Imposto('420', '42', '1700', '170',
'1200', '120', 'SP', '14.2.a');
```

4.2.13 EPSON_NFCe_Dados_Lei_ImpostoEX

Esta função a inclusão dos dados da Lei "De olho no imposto" na NFC-e de acordo com o buffer passado como parâmetro. A biblioteca não faz nenhum tratamento na mensagem recebida, apenas imprime a mesma.

Sintaxe:

```
function EPSON_NFCe_Dados_Lei_ImpostoEX (szMensagem:Pchar):Integer; StdCall; External
                                                    'InterfaceEpsonNF.dll';
```

Entradas:

Variável	Tipo	Tam.	Descrição
szMensagem	PCHAR	256	Mensagem da Lei "De olho no imposto.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Dados_Lei_ImpostoEX( "Trib aprox R$ 4,45 Federal, R$
5,40 Estadual" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Dados_Lei_ImpostoEX( "Trib aprox R$ 4,45 Federal, R$
5,40 Estadual" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Dados_Lei_ImpostoEX('Trib aprox R$ 4,45 Federal, R$
5,40 Estadual');
```

4.2.14 EPSON_NFCe_Fechar

Esta função executa o fechamento e envio de uma NFC-e.

Sintaxe:

function EPSON_NFCe_Fechar(szImprime:Pchar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szImprime	PCHAR	1	Imprime DANFE em caso de sucesso de envio: **S** – Sim. **N** – Não.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Uma NFC-e deve estar aberta.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Fechar( "S" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Fechar( "S" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Fechar('S');
```

4.2.15 EPSON_NFCe_Reimprimir

Esta função executa a reimpressão da última NFC-e autorizada.

Sintaxe:

function EPSON_NFCe_Reimprimir():Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Reimprimir();
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Reimprimir()
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Reimprimir();
```

4.2.16 EPSON_NFCe_Cancelar

Esta função executa o cancelamento de uma NFC-e.

Sintaxe:

function EPSON_NFCe_Cancelar(szNumero: PChar; szSerie: PChar; szChave: PChar; szProtocolo: PChar; szJustificativa: PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szNumero	PCHAR	9	Número da NFC-e que será cancelada.
szSerie	PCHAR	3	Série da NFC-e que será cancelada.
szChave	PCHAR	44	Chave de acesso da NFC-e.
szProtocolo	PCHAR	15	Protocolo de autorização da NFC-e.
szJustificativa	PCHAR	255	Justificativa do cancelamento.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Cancelar( "1", "1",
"11112222333344445555666677778888999900001111", "123456789012345",
"Desistencia da compra" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Cancelar( "1", "1",
"11112222333344445555666677778888999900001111", "123456789012345",
"Desistencia da compra" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Cancelar( '1', '1',
'11112222333344445555666677778888999900001111', '123456789012345',
'Desistencia da compra' );
```

4.2.17 EPSON_NFCe_Inutilizar_Numeros

Esta função executa a inutilização de números de NFC-e não utilizados.

Sintaxe:

function EPSON_NFCe_Inutilizar_Numeros(szInicial: PChar; szFinal: PChar; szSerie: PChar; szJustificativa: PChar):Integer; StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szInicial	PCHAR	9	Número inicial que será inutilizado.
szFinal	PCHAR	9	Número final que será inutilizado.
szSerie	PCHAR	3	Série dos números que serão inutilizados.
szJustificativa	PCHAR	255	Justificativa da inutilização.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Inutilizar_Numeros( "100", "110", "1", "Erro de sincronia" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Inutilizar_Numeros( "100", "110", "1", "Erro de sincronia" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Inutilizar_Numeros( '100', '110', '1', 'Erro de sincronia' );
```

4.2.18 EPSON_NFCe_Obter_Erro

Esta função executa a leitura do status da última comunicação com o servidor da SEFAZ.

Sintaxe:

function EPSON_NFCe_Obter_Erro (szCodigo:PChar; szDescricao:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Variável	Tipo	Tam.	Descrição
szCodigo	PCHAR	5	Código do erro retornado.
szDescricao	PCHAR	601	Descrição do erro.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder:

```
char szCodigo[5], szDescricao[601];
Retorno = EPSON_NFCe_Obter_Erro( szCodigo, szDescricao);
```

Exemplo em C#:

```
StringBuilder szCodigo = new StringBuilder(5, 5);
StringBuilder szDescricao = new StringBuilder(601, 601);
Retorno = EPSON_NFCe_Obter_Erro( szCodigo, szDescricao);
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Dim  szCodigo As String
Dim  szDescricao As String
szCodigo = Space(5)
szDescricao = Space(601)
iRetorno = EPSON_NFCe_Obter_Erro( szCodigo, szDescricao)
```

Exemplo em Delphi 7:

```
szCodigo: array[0..5] of Char;
szDescricao: array[0..601] of Char;
iRetorno := EPSON_NFCe_Obter_Erro( szCodigo, szDescricao);
```

Exemplo em Delphi 2010 / Delphi XE:

```
szCodigo := AnsiString(StringOfChar(' ', 5));
szDescricao := AnsiString(StringOfChar(' ', 601));
iRetorno := EPSON_NFCe_Obter_Erro( szCodigo, szDescricao);
```

4.2.19 EPSON_NFCe_Obter_Estado

Esta função executa a leitura do status da NFC-e atual.

Sintaxe:

function EPSON_NFCe_Obter_Estado(szEstado:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Variável	Tipo	Tam.	Descrição
szEstado	PCHAR	2	Estado atual da NFC-e.

Os valores possíveis de estado são:

0	:	NFC-e Fechada.
1	:	NFC-e Aberta mas sem registro de itens.
2	:	NFC-e Aberta e com itens registrados.
3	:	NFC-e Totalizada.
4	:	NFC-e Paga.
5	:	NFC-e Encerrada.
6	:	NFC-e Cancelada.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder:

```
char  szEstado[2];
Retorno = EPSON_NFCe_Obter_Estado( szEstado );
```

Exemplo em C#:

```
StringBuilder szEstado = new StringBuilder(2,2);
Retorno = EPSON_NFCe_Obter_Estado( szEstado );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Dim szEstado As String
szEstado = Space(2)
iRetorno = EPSON_NFCe_Obter_Estado (  szEstado )
```

Exemplo em Delphi 7:

```
szEstado : array[0..2] of Char;
iRetorno := EPSON_NFCe_Obter_Estado( szEstado );
```

Exemplo em Delphi 2010 / Delphi XE:

```
szEstado := AnsiString(StringOfChar(' ', 2));
iRetorno := EPSON_NFCe_Obter_Estado( szEstado );
```

4.2.20 EPSON_NFCe_Verifica_Servidor

Esta função executa a verificação do estado do servidor da SEFAZ.

Sintaxe:

function EPSON_NFCe_Verifica_Servidor(szEstado:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Nenhum.

Saídas:

Variável	Tipo	Tam.	Descrição
szEstado	PCHAR	4	Código de retorno do servidor.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder:

```
char  szEstado[4];
Retorno = EPSON_NFCe_Verifica_Servidor( szEstado );
```

Exemplo em C#:

```
StringBuilder szEstado = new StringBuilder(4,4);
Retorno =  EPSON_NFCe_Verifica_Servidor( szEstado );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Dim szEstado As String
szEstado = Space(4)
iRetorno = EPSON_NFCe_Verifica_Servidor(  szEstado )
```

Exemplo em Delphi 7:

```
szEstado : array[0..4] of Char;
iRetorno := EPSON_NFCe_Verifica_Servidor( szEstado );
```

Exemplo em Delphi 2010 / Delphi XE:

```
szEstado := AnsiString(StringOfChar(' ', 4));
iRetorno := EPSON_NFCe_Verifica_Servidor( szEstado );
```

4.2.21 EPSON_NFCe_Imprimir_Mensagem

Esta função executa a impressão da mensagem promocional na NFC-e.

Sintaxe:

function EPSON_NFCe_Imprimir_Mensagem(szMensagem:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szMensagem	PCHAR	4096	Mensagem Promocional.

Saídas:

Nenhum.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder / C# :

```
Retorno = EPSON_NFCe_Imprimir_Mensagem( "Obrigado e Volte Sempre" );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Retorno = EPSON_NFCe_Imprimir_Mensagem( "Obrigado e Volte Sempre" )
```

Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:

```
Retorno := EPSON_NFCe_Imprimir_Mensagem( 'Obrigado e Volte Sempre' );
```

4.2.22 EPSON_NFCe_Obter_Informacao

Esta função executa a recuperação de informações da NFC-e atual (caso exista uma em andamento) ou da última NFCe fechada.

Sintaxe:

function EPSON_NFCe_Obter_Informacao (szIndice:PChar; szInformacao:PChar):Integer;StdCall; External 'InterfaceEpsonNF.dll';

Entradas:

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	3	Índice da informação que será recupeda.

Os índices possíveis de informações são:

1	:	Chave de acesso da NFC-e.
2	:	Protocolo de autorização da NFC-e.

Saídas:

Variável	Tipo	Tam.	Descrição
szInformacao	PCHAR	128	Ínformação que será recupeda.

Retornos:

FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

Requisitos:

Nenhum.

Exemplo em C / Visual C++ / C++ Builder:

```
char  szInformacao[128];
Retorno = EPSON_NFCe_Obter_Informacao( "1", szInformacao );
```

Exemplo em C#:

```
StringBuilder szInformacao = new StringBuilder(128,128);
Retorno = EPSON_NFCe_Obter_Informacao( "1", szInformacao );
```

Exemplo em Visual Basic 6.0 / Visual Basic.NET:

```
Dim szInformacao As String
szInformacao = Space(128)
iRetorno = EPSON_NFCe_Obter_Informacao( "1", szInformacao )
```

Exemplo em Delphi 7:

```
szInformacao : array[0..128] of Char;
iRetorno := EPSON_NFCe_Obter_Informacao( '1', szInformacao );
```

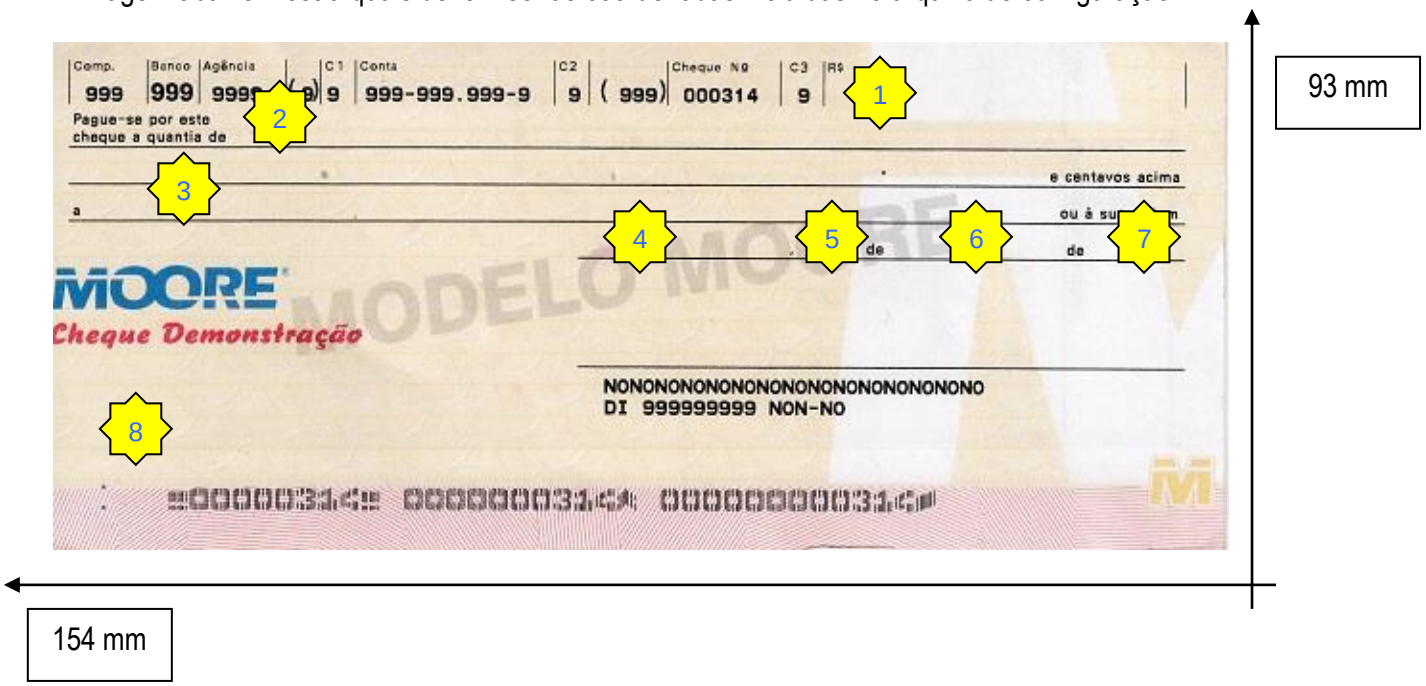
Exemplo em Delphi 2010 / Delphi XE:

```
szInformacao := AnsiString(StringOfChar(' ', 128));
iRetorno := EPSON_NFCe_Obter_Informacao( '1', szInformacao );
```

5 Layout do cheque

Devido a não padronização do layout dos cheques utilizados no país, é necessário a inclusão das principais coordenadas de impressão dos cheques que serão impressos. Estas coordenadas devem ser incluídas no arquivo de configurações da biblioteca, chamado "InterfaceEpsonNF.xml"

A imagem abaixo mostra quais devem ser as coordenadas incluídas no arquivo de configuração:



Campo	Tag no arquivo XML	Descrição
1	<VALOR>	Coordenadas para a impressão do valor do cheque.
2	<EXTENSO>	Coordenadas para a impressão do valor por extenso do cheque.
3	<PARA>	Coordenadas para a impressão do destinatário do cheque.
4	<CIDADE>	Coordenadas para a impressão da cidade onde o cheque foi emitido.
5	<DIA>	Coordenadas para a impressão do dia da emissão do cheque.
6	<MES>	Coordenadas para a impressão do mês da emissão do cheque.
7	<ANO>	Coordenadas para a impressão do ano da emissão do cheque.
8	<ADICIONAL>	Coordenadas para a impressão de um texto adicional no cheque.

Dentro do grupo das coordenadas, devem ser incluídas as tags <HORIZONTAL> e <VERTICAL> contendo as coordenadas de impressão em milímetros, a partir da margem inferior direita do cheque.

Além das tags de coordenadas, devem ser incluídas também uma tag referente ao índice do cheque no arquivo, e outra com a descrição do tipo do cheque.

Opcionalmente, outras configurações podem ser incluídas no arquivo XML. Na tabela abaixo são mostradas as tags opcionais que podem ser incluídas no arquivo:

Tag no arquivo XML	Descrição
<TIMEOUT>	Tempo de espera para a inserção do cheque. Caso este tempo seja expirado, a função retornará erro. O tempo deve ser incluído em segundos.
<MOEDA>	Grupo contendo as configurações de moeda.
<SINGULAR>	Descrição da moeda utilizada na impressão do cheque no singular.
<PLURAL>	Descrição da moeda utilizada na impressão do cheque no plural.
<CENTAVOS>	Flag que indica se a palavra “CENTAVOS” deve ser incluída na impressão do valor por extenso do cheque. Devem ser utilizados os valores ‘S’ para Sim e ‘N’ para Não.
<MICR>	Grupo contendo as configurações de leitura do MICR do cheque.
<TIMEOUT>	Tempo de espera para a inserção do cheque. Caso este tempo seja expirado, a função retornará erro. O tempo deve ser incluído em segundos.
<TIPO>	Tipo do MICR que será lido. Deverá ser utilizado o valor ‘0’ para leitura do padrão CMC7.

A imagem abaixo mostra a configuração de um cheque, gravada no arquivo InterfaceEpsonNF.xml:

```
<EPSON>
    <NAO_FISCAL>
        <CHEQUES>
            <TIMEOUT>25</TIMEOUT>
            <MOEDA>
                <SINGULAR>REAL</SINGULAR>
                <PLURAL>REAIS</PLURAL>
                <CENTAVOS>S</CENTAVOS>
            </MOEDA>
            <CHEQUE>
                <CODIGO>1</CODIGO>
                <DESCRICAO>BANCO MOORE</DESCRICAO>
                <VALOR>
                    <HORIZONTAL>55</HORIZONTAL>
                    <VERTICAL>65</VERTICAL>
                </VALOR>
                <EXTENSO>
                    <HORIZONTAL>145</HORIZONTAL>
                    <VERTICAL>60</VERTICAL>
                </EXTENSO>
                <PARA>
                    <HORIZONTAL>165</HORIZONTAL>
                    <VERTICAL>50</VERTICAL>
                </PARA>
                <CIDADE>
                    <HORIZONTAL>95</HORIZONTAL>
                    <VERTICAL>43</VERTICAL>
                </CIDADE>
                <DIA>
                    <HORIZONTAL>60</HORIZONTAL>
                    <VERTICAL>43</VERTICAL>
                </DIA>
                <MES>
                    <HORIZONTAL>49</HORIZONTAL>
                    <VERTICAL>43</VERTICAL>
                </MES>
                <ANO>
                    <HORIZONTAL>19</HORIZONTAL>
                    <VERTICAL>43</VERTICAL>
                </ANO>
                <ADICIONAL>
                    <HORIZONTAL>160</HORIZONTAL>
                    <VERTICAL>10</VERTICAL>
                </ADICIONAL>
            </CHEQUE>
        </CHEQUES>
        <MICR>
            <TIMEOUT>25</TIMEOUT>
            <TIPO>0</TIPO>
        </MICR>
    </NAO_FISCAL>
</EPSON>
```

Configuração do timeout para impressão do cheque.

Configuração da moeda utilizada na impressão do cheque.

Coordenadas utilizadas na impressão do cheque.

Configurações utilizadas na leitura do MICR do cheque.

6 Configuração do provedor de messageria NFC-e

Atualmente a InterfaceEpsonNF está integrada com 4 provedores de messageria de NFC-e, listados abaixo em ordem alfabética:

- IT Works
- Oobj
- Paperless
- Tecnospeed

Após escolher o provedor que será utilizado na emissão de NFC-e, o arquivo InterfaceEpsonNF.xml deverá ser configurado, sendo incluídos os dados necessários para correta emissão das notas.

Será necessário incluir no arquivo InterfaceEpsonNF.xml a TAG raiz <NFCE> que irá guardar todas as configurações relativas a emissão das notas.

Como TAGS filhas da <NFCE> termos 6 TAGS simples e mais 2 grupos de configurações, conforme a tabela abaixo:

Tag no arquivo XML	Descrição
<CST_DEFAULT>	Valor de CST padrão que será utilizado, caso este valor não seja informado na venda do item.
<PIS_DEFAULT>	Alíquota padrão de PIS que será utilizado, caso este valor não seja informado na venda do item.
<COFINS_DEFAULT>	Alíquota padrão de COFINS que será utilizado, caso este valor não seja informado na venda do item.
<SimplesNacional>	Flag que indica se o estabelecimento faz parte do Simples Nacional (S – sim, N – não).
<cStat>	Tag que irá guardar o código da última comunicação com o servidor da SEFAZ.
<xMotivo>	Tag que irá guardar a resposta da última comunicação com o servidor da SEFAZ.
<ide>	Grupo de informações relativas ao grupo “ide” que será utilizado na emissão da NFC-e.
<emit>	Grupo de informações relativas ao grupo “emit” que será utilizado na emissão da NFC-e.

Abaixo segue a tabela com os campos que fazem parte do grupo “ide”:

Tag no arquivo XML	Descrição
<cUF>	Código da UF do emitente do Documento Fiscal.
<natOp>	Descrição da Natureza da Operação.
<indPag>	Indicador da forma de pagamento.
<mod>	Código do Modelo do Documento Fiscal.
<serie>	Série do Documento Fiscal.
<nNF>	Número do Documento Fiscal.
<dhEmi>	Timezone que será utilizado na data de emissão do Documento Fiscal.

<idDest>	Identificador de local de destino da operação.
cMunFG	Código do Município de Ocorrência do Fato Gerador.
tpImp	Formato de Impressão do DANFE.
tpEmis	Tipo de Emissão da NFC-e
tpAmb	Identificação do Ambiente.
finNFe	Finalidade de emissão da NFC-e.
indFinal	Indica operação com Consumidor final.
indPres	Indicador de presença do comprador no estabelecimento comercial no momento da operação.
procEmi	Processo de emissão da NFC-e.
verProc	Versão do Processo de emissão da NFC-e.

Abaixo segue a tabela com os campos que fazem parte do grupo “emit”:

Tag no arquivo XML	Descrição
CNPJ	CNPJ do emitente.
xNome	Razão Social ou Nome do emitente.
IE	Inscrição Estadual do emitente.
CRT	Código de Regime Tributário.
enderEmit	Grupo do Endereço do emitente.
xLgr	Logradouro do emitente.
nro	Número do logradouro.
xBairro	Bairro do logradouro.
cMun	Código do município.
xMun	Nome do município.
UF	Sigla da UF.
CEP	Código do CEP.

A imagem abaixo mostra as configurações iniciais, gravadas no arquivo InterfaceEpsonNF.xml:

```
<NFCE>
    <CST_DEFAULT></CST_DEFAULT>
    <PIS_DEFAULT></PIS_DEFAULT>
    <COFINS_DEFAULT></COFINS_DEFAULT>
    <SimplesNacional>N</SimplesNacional>
    <cStat></cStat>
    <xMotivo></xMotivo>
    <ide>
        <cUF>41</cUF>
        <natOp>Venda</natOp>
        <indPag>0</indPag>
        <mod>65</mod>
        <serie>146</serie>
        <nNF>11</nNF>
        <dhEmi>-02:00</dhEmi>
        <idDest>1</idDest>
        <cMunFG>4115200</cMunFG>
        <tpImp>4</tpImp>
        <tpEmis>1</tpEmis>
        <tpAmb>2</tpAmb>
        <finNFe>1</finNFe>
        <indFinal>1</indFinal>
        <indPres>1</indPres>
        <procEmi>0</procEmi>
        <verProc>1.0</verProc>
    </ide>
    <emit>
        <CNPJ>xxxxxxxxx</CNPJ>
        <xNome>xxxxxxxxx</xNome>
        <IE>ISENTO</IE>
        <CRT>3</CRT>
        <enderEmit>
            <xLgr>xxxxxxxxxx</xLgr>
            <nro>xxxxx</nro>
            <xBairro>xxxxxxx</xBairro>
            <cMun>xxxxxxx</cMun>
            <xMun>xxxxxxx</xMun>
            <UF>xxxxx</UF>
            <CEP>xxxxxx</CEP>
        </enderEmit>
    </emit>
</NFCE>
```

Após a inclusão da configuração padrão no arquivo de configuração, deverão ser incluídas as informações relativas ao provedor de messageria de NFC-e.

Abaixo seguem as tabelas para cada um dos provedores:

1. IT Works

Tag no arquivo XML	Descrição
ITWORKS	Grupo das configurações IT Works.
IP	IP do serviço IT Works.
PORTA	Porta de comunicação que deve ser utilizada.
TIMEOUT_SOCKET	Time-out que será utilizado na comunicação com o serviço da IT Works.
Centralizador	Número do centralizador.
CaixaEntrada	Número da caixa de entrada.
PDV	Número do PDV.
AssinarXML	Flag que indica se o XML deve ser assinado pela IT Works (1 – Sim, 0 – Não).
UrlFisco	URL de consulta do fisco.
cIdToken	Identificador do CSC – Código de Segurança do Contribuinte no Banco de Dados da SEFAZ.
CSC	Código de Segurança do Contribuinte (antigo Token).

2. Oobj

Tag no arquivo XML	Descrição
OOBJ	Grupo das configurações Oobj.
PASTA_ENTRADA	Pasta no PDV utilizada como entrada para comunicação com o motor periférico Oobj.
PASTA_SAIDA	Pasta no PDV utilizada como saida para comunicação com o motor periférico Oobj.
PASTA_TEMPORARIA	Pasta no PDV temporária utilizada para comunicação com o motor periférico Oobj.
UrlConsulta	URL de consulta do fisco.
cIdToken	Identificador do CSC – Código de Segurança do Contribuinte no Banco de Dados da SEFAZ.
CSC	Código de Segurança do Contribuinte (antigo Token).

3. Paperless

Tag no arquivo XML	Descrição
PAPERLESS	Grupo das configurações Paperless.
IP	IP do serviço Paperless.
PORTA	Porta de comunicação que deve ser utilizada.
TIMEOUT_SOCKET	Time-out que será utilizado na comunicação com o serviço da Paperless.
Modo	Flag de modo de treinamento (1 – Sim, 0 – Não).
CNPJ	CNPJ do emitente.
Loja	Número da Loja.
PDV	Número do PDV.
Sessao	Número da última sessão utilizada.

4. Tecnospeed

Tag no arquivo XML	Descrição
TECNOSPEED	Grupo das configurações TecnoSpeed.
IP	IP do serviço TecnoSpeed.
PORTA	Porta de comunicação que deve ser utilizada.
TIMEOUT_SOCKET	Time-out que será utilizado na comunicação com o serviço da TecnoSpeed.
Usuario	Usuário cadastrado na TecnoSpeed.
Senha	Senha cadastrada na TecnoSpeed.
Grupo	Grupo cadastrado na TecnoSpeed.
CNPJ	CNPJ do emitente.
NeverStop	Flag que indica se o NeverStop está sendo utilizado na comunicação (S – Sim, N – Não).
UrlConsulta	URL de consulta do fisco.
cIdToken	Identificador do CSC – Código de Segurança do Contribuinte no Banco

	de Dados da SEFAZ.
CSC	Código de Segurança do Contribuinte (antigo Token).

Os arquivos base de configuração para cada provedor de messageria estão disponíveis na página de suporte a impressoras não fiscais térmicas da Epson.

7 Configuração do arquivo produtos.xml

O arquivo produtos.xml deve ser utilizado para guardar informações complementares dos itens, necessárias para a emissão da Nota Fiscal do consumidor eletrônica (NFC-e).

A tabela abaixo mostra a descrição dos campos necessários:

Tag no arquivo XML	Descrição
<PRODUTO>	Grupo de informações relativas ao produto.
<CODIGO>	Código que será passado no registro do item.
<EAN>	Valor do código EAN do item (caso exista).
<DESCRICAO>	Descrição do produto.
<NCM>	Código NCM do produto.
<CST>	CST do item.
<PIS>	Alíquota de PIS do item.
<COFINS>	Alíquota de COFINS do item.
<COMBUSTIVEL>	Flag que indica se o item é do tipo combustível (S – Sim, N – Não).

A imagem abaixo mostra um exemplo de configuração, gravado no arquivo produtos.xml:

```
<PRODUTOS>
    <PRODUTO>
        <CODIGO>1</CODIGO>
        <EAN>1</EAN>
        <DESCRICAO>Produto 01</DESCRICAO>
        <NCM>34052000</NCM>
        <CST>01</CST>
        <PIS>1,00</PIS>
        <COFINS>1,00</COFINS>
        <COMBUSTIVEL>N</COMBUSTIVEL>
    </PRODUTO>
</PRODUTOS>
```

8 Resolvendo importação dos nomes das funções

Caso você esteja tendo problemas na importação dos nomes das funções da InterfaceEpsonNF (assinaturas já exportadas por outras bibliotecas), você poderá utilizar o prefixo “EPSON_” em todas as funções para resolver este problema.

Abaixo listamos como ficam os nomes alterados das funções:

Nome da função	Nome da função equivalente
ConfiguraTaxaSerial	EPSON_ConfiguraTaxaSerial
IniciaPorta	EPSON_IniciaPorta

FechaPorta	EPSON_FechaPorta
ImprimeTexto	EPSON_ImprimeTexto
ImprimeTextoTag	EPSON_ImprimeTextoTag
FormataTX	EPSON_FormataTX
AcionaGuilhotina	EPSON_AcionaGuilhotina
ComandoTX	EPSON_ComandoTX
Le_Status	EPSON_Le_Status
Le_Status_Gaveta	EPSON_Le_Status_Gaveta
ConfiguraCodigoBarras	EPSON_ConfiguraCodigoBarras
ImprimeCodigoBarrasCODABAR	EPSON_ImprimeCodigoBarrasCODABAR
ImprimeCodigoBarrasCODE128	EPSON_ImprimeCodigoBarrasCODE128
ImprimeCodigoBarrasCODE39	EPSON_ImprimeCodigoBarrasCODE39
ImprimeCodigoBarrasCODE93	EPSON_ImprimeCodigoBarrasCODE93
ImprimeCodigoBarrasEAN13	EPSON_ImprimeCodigoBarrasEAN13
ImprimeCodigoBarrasEAN8	EPSON_ImprimeCodigoBarrasEAN8
ImprimeCodigoBarrasITF	EPSON_ImprimeCodigoBarrasITF
ImprimeCodigoBarrasUPCA	EPSON_ImprimeCodigoBarrasUPCA
ImprimeCodigoBarrasUPCE	EPSON_ImprimeCodigoBarrasUPCE
ImprimeCodigoBarrasPDF417	EPSON_ImprimeCodigoBarrasPDF417
ImprimeCodigoQRCODE	EPSON_ImprimeCodigoQRCODE
GerarQRCodeArquivo	EPSON_GerarQRCodeArquivo
ImprimeBmpEspecial	EPSON_ImprimeBmpEspecial
Habilita_Log	EPSON_Habilita_Log
ImprimeCheque	EPSON_ImprimeCheque
LeMICR	EPSON_LeMICR
LeModelo	EPSON_LeModelo
Le_Status_Slip	EPSON_Le_Status_Slip
AcionaGaveta	EPSON_AcionaGaveta

9 Configuração para impressão de Danfe NFC-e e Extrato do SAT-CFE a partir de um XML

Para que os comprovantes da Danfe NFC-e e/ou Extrato do SAT-CFE sejam impressos com sucesso, será necessário a inclusão de algumas tags de configuração no arquivo InterfaceEpsonNF.xml

A tabela abaixo mostra a descrição dos campos necessários:

Tag no arquivo XML	Descrição
<MODELO>	Modelo da impressora que será utilizada.
<NFCE>	Grupo de informações relativas a impressão da Danfe NFC-e.
<UrlFisco>	Url do fisco que será utilizada para efetuar a verificação do QR Code impresso na Danfe NFC-e.
<cIdToken>	Token do estabelecimento que será utilizado para a composição do QR Code impresso na Danfe NFC-e.
<CSC>	CSC do estabelecimento que será utilizado para a composição do QR Code impresso na Danfe NFC-e.

Para a impressão exclusiva de Extratos do SAT-CFE somente a tag <MODELO> precisa ser preenchida. Para a impressão da Danfe NFC-e, todos os campos devem ser preenchidos.

A imagem abaixo mostra um exemplo de configuração somente para SAT-CFE, gravado no arquivo InterfaceEpsonNF.xml:

```
<EPSON>
    <NAO_FISCAL>
        <MODELO>T20</MODELO>
    </NAO_FISCAL>
</EPSON>
```

A imagem abaixo mostra um exemplo de configuração completa, gravado no arquivo InterfaceEpsonNF.xml:

```
<EPSON>
    <NAO_FISCAL>
        <MODELO>T20</MODELO>
        <NFCE>
            <UrlFisco>http://homnfce.sefaz.am.gov.br/nfceweb/consultarNFCe.jsp</UrlFisco>
            <cIdToken>000001</cIdToken>
            <CSC>C1774291-A86A-1234-B247-79129999CF50</CSC>
        </NFCE>
    </NAO_FISCAL>
</EPSON>
```