

PRJ-DLL-001 – BIBLIOTECA VINCULO DINÂMICO
IMPRESSORAS NÃO FISCAIS
DOCUMENTO DE REQUISITOS

Versão <1.2>

Notas de revisão

Revisão	Data	Responsável	Motivo
1.0	12/02/2016	Antônio	Versão Inicial
1.1	26/02/2016	Antônio	Alterado o retorno de 0x00 e 0x01 para 0 e 1 Incluído nova funções: LeVersaoFirmware LeDataVersaoFirmware LinhasGuilhotina ConfigurarGuilhotina StatusGuilhotina ModoGaveta

Conteúdo

Objetivo.....	5
Perspectiva da solução.....	5
Glossário.....	5
Requisitos Funcionais.....	6
RF0001 – int ConfiguraTaxaSerial(int iBaudRate)	6
RF0002 – int IniciaPorta (char *szPortaSerial)	6
RF0003 – int FechaPorta ()	7
RF0004 – int ImprimeTexto (char *szTexto)	7
RF0005 – int ImprimeTextoTag (char *szTexto).....	8
RF0006 – int FormataTX (char *szTexto, int iTipoLetra, int iltalico, int iSublinhado, int iExpandido, int iEnfatizado)	9
RF0007 – int AcionaGuilhotina (int iTipoCorte)	10
RF0008 – int ComandoTX (char * szComandoESCPOS, int iTamanho)	10
RF0009 – int Le_Status ().....	11
RF0010 – int Le_Status_Gaveta()	11
RF0011 – int ConfiguraCodigoBarras (int iAltura, int iHRI, int iFonte, int iMargem)	12
RF0012 – int ImprimeCodigoBarrasCODABAR (char * szCodigo).....	13
RF0013 – int ImprimeCodigoBarrasCODE128 (char * szCodigo).....	13
RF0014 – int ImprimeCodigoBarrasCODE39 (char * szCodigo).....	13
RF0015 – int ImprimeCodigoBarrasCODE93 (char * szCodigo).....	14
RF0016 – int ImprimeCodigoBarrasEAN13 (char * szCodigo)	14
RF0017 – int ImprimeCodigoBarrasEAN8 (char * szCodigo)	15
RF0018 – int ImprimeCodigoBarrasITF (char * szCodigo)	15
RF0019 – int ImprimeCodigoBarrasUPCA (char * szCodigo).....	16
RF0020 – int ImprimeCodigoBarrasUPCE (char * szCodigo)	16

RF0021 – int ImprimeCodigoBarrasPDF417 (char * szCodigo)	16
RF0022 – int ImprimeCodigoQRCode (int iRestauracao, int iModulo, int iTipo, int iVersao, int iModo, char * szCodigo)	17
RF0023 – int GerarQRCodeArquivo (char * szNomeArquivoDestino, char * szDados)	18
RF0024 – int ImprimeBitmap (char * szPathArquivo)	18
RF0025 – int LeModelo (char * szModelo)	19
RF0025 – int AcionaGaveta ()	19
RF0026 – int ImprimirSAT (char *szPathArquivoXML, char szTipo)	20
RF0026 – int ImprimirSAT_Cancelamento (char *szPathArquivoXML, char *szQrCodeVenda, char *szModoProducao)	20
RF0027 – int LinhasGuilhotina (char *szQtdeLinhas)	21
RF0028 – int ConfiguraGuilhotina (int iHabilita, int qdteLinhas)	21
RF0029 – int StatusGuilhotina ()	22
RF0030 – int ModoGaveta (int iModo)	22
RF0030 – int ModoGaveta (int iModo)	23
RF0031 – int LeVersaoFirmware (char * szModelo)	23
RF0032 – int LeDataVersaoFirmware (char * szModelo)	24
Requisitos não funcionais	25
Software	25
RNF0001 – Definição de Retornos	25
RNF0002 – Captura de informações da impressora fiscal	25
RNF0003 – Desenvolver lib/algoritmo para geração do QRCODE	25

Objetivo

O objetivo deste documento é apresentar os requisitos para o desenvolvimento de uma biblioteca de vínculo dinâmico (DLL para Windows, SO para Linux) para maximizar o desenvolvimento da software house quando integrar seu produto as impressoras não fiscais da CIS e aproveitar o máximo possível dos recursos do equipamento e facilitar o desenvolvimento do integrador.

Perspectiva da solução

- Interface amigável
- Maior agilidade no processo de integração
- Melhor aproveitamento dos recursos do equipamento
- Multiplataformas (Windows e Linux)
- Controle da porta serial
- Facilidade da impressão de QR Code
- Facilidade de impressão dos Códigos de barras
- Facilidade da impressão de textos com formatações
- Leitura de status do equipamento
- Impressão do XML SAT
- Impressão do XML NFCe *

Glossário

A lista abaixo descreve os termos técnicos usados no documento e durante o projeto

- SAT – Sistema Autenticador e Transmissor de Documentos fiscais utilizados atualmente no estado de São Paulo. Maiores informações sobre esse sistema acesse: <http://www.fazenda.sp.gov.br/sat/>
- NFCe - Nota Fiscal de Consumidor Eletrônica é o projeto que está substituindo o tradicional Cupom Fiscal emitido por ECF e a Nota Fiscal de venda a consumidor - modelo 2 - emitida manualmente em lojas, supermercados, drogarias e comércio varejista em geral na grande maioria dos estados brasileiros.

Requisitos Funcionais

RF0001 – int ConfiguraTaxaSerial(int iBaudRate)

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

Entradas:

115200 – 57600 – 38400 – 19200 – 9600

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

*RF0002 – int IniciaPorta (char *szPortaSerial)*

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.

Entradas:

COM1, COM2 (Windows) /dev/ttyS0, /dev/ttyS1 (Linux)

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

RF0003 – int 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.

Entradas:

Nenhuma

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

*RF0004 – int ImprimeTexto (char *szTexto)*

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.

Entradas:

Ponteiro para char, tamanho máximo de 2048 bytes.

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

RF0005 – *int ImprimeTextoTag (char *szTexto)*

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. Nesta função podemos incluir TAGS que podem indicar:

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

Entradas:

Ponteiro para char, tamanho máximo de 4096 bytes.

Tag suportadas pela função:

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 Texto
<da></da>	Altura Dupla
<xl></xl>	Altura e Largura Dupla
<g></g>	Aciona a Gaveta
<gui></gui>	Aciona a Guilhotina
<bmp>XX,XX</bmp>	Redefinir, verificar a qtde de logo que é possível ter na impressora (checar suporte)
<ibmp>XXX</ibmp>	Imprime imagem BMP a partir do path completo XXX e com extensão .bmp
<cespl></cespl>	Configura espaço entre linhas
<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 Code39
<code93></code93>	Imprime código de barras padrão 93
<codabar></codabar>	Imprime código de barras CODABAR
<i2of5></i2of5>	Imprime código de barras padrão interleave 2&5

<code128></code128>	Imprime código de barras padrão CODE 128B
<code128c></code128c>	Imprime código de barras padrão Code 128C
<pdf></pdf>	Checar suporte
<qrcode></qrcode>	Imprime código de barras padrão QR-Code
<lmodulo></lmodulo>	Define largura do módulo do QR-Code
<correcao></correcao>	Define o fator de correção do QR-Code

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Alta	Alta	Alta

*RF0006 – int FormataTX (char *szTexto, int iTipoLetra, int iltalico, int iSublinhado, int iExpandido, int iEnfatizado)*

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

Entradas:

Variável	Descrição
szTexto	Texto a ser impresso
iTipoLetra	1 – Comprimido 2 – Normal
iltalico	0 – Desativa itálico 1 – Ativa itálico
iSublinhado	0 – Desativa sublinhado 1 - Ativa sublinhado
iExpandido	0 – Desativa expandido 1 – Ativa expandido
iEnfatizado	0 – Desativa enfatizado 1 – Ativa enfatizado

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

RF0007 – int AcionaGuilhotina (int iTipoCorte)

Esta função executa o acionamento da guilhotina

Entradas:

iTipoCorte: 1 – Total / 0 - Parcial

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

RF0008 – int ComandoTX (char * szComandoESCPOS, int iTamanho)

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

Entradas:

szComandoESCPOS : sequencia de comandos ESC/POS

iTamanho : Total de bytes do comando enviado em szComandoESCPOS

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

RF0009 – int Le_Status ()

Retorna o status da impressora

Entradas:

Nenhuma

Saídas:

Nenhuma

Retornos: CHECAR CAPTURA DE RETORNO

CIS_ERRO	
5	Impressora com pouco papel
9	Tampa aberta
24	Impressora ONLINE
32	Impressora sem papel

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

RF0010 – int Le_Status_Gaveta()

Esta função efetua a leitura do status da gaveta

Entradas:

Nenhuma

Saídas:

Nenhuma

Retornos: CHECAR CAPTURA DE RETORNO

CIS_ERRO	
1	Gaveta aberta
2	Gaveta fechada

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

RF0011 – int ConfiguraCodigoBarras (int iAltura, int iHRI, int iFonte, int iMargem)

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

Entradas:

iAltura	Altura do código de barras (1 e 255)
iLargura	Largura da barra utilizada 0 – Barras finas 1 – Barras médias 2 – Barras grossas 3 – Barras padrão
iHRI	Posição dos caracteres do código de barras 0 – Não imprime os caracteres 1 – Imprime os caracteres acima 2 – Imprime os caracteres abaixo 3 – Imprime os caracteres acima e abaixo
iFonte	0 – Normal 1 – Condensada
iMargem	Valor da margem (0 e 575)

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0012 – int ImprimeCodigoBarrasCODABAR (char * szCodigo)*

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

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0013 – int ImprimeCodigoBarrasCODE128 (char * szCodigo)*

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

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0014 – int ImprimeCodigoBarrasCODE39 (char * szCodigo)*

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

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

RF0015 – int ImprimeCodigoBarrasCODE93 (char * szCodigo)

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

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

RF0016 – int ImprimeCodigoBarrasEAN13 (char * szCodigo)

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

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

RF0017 – int ImprimeCodigoBarrasEAN8 (char * szCodigo)

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

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

RF0018 – int ImprimeCodigoBarrasITF (char * szCodigo)

Esta função efetua a impressão de um código de barras do tipo Interleave 2&5.

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0019 – int ImprimeCodigoBarrasUPCA (char * szCodigo)*

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

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0020 – int ImprimeCodigoBarrasUPCE (char * szCodigo)*

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

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0021 – int ImprimeCodigoBarrasPDF417 (char * szCodigo)*

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

Entradas:

szCodigo: Código que será impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0022 – int ImprimeCodigoQRCode (int iRestauracao, int iModulo, int iTipo, int iVersao, int iModo, char * szCodigo)*

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

Entradas:

iRestauracao	Nível de restauração do código 0 – 7% 1 – 15% 2 – 25% 3 – 30%
iModulo	Tamanho do módulo do código Entre 1 e 127
iTipo	0 – Normal 1 – Reduzido
iVersao	Versão do QR-Code (Entre 1 e 40)
iModo	Tipo dos dados que serão impressos 0 – Numérico 1 – Alfanumérico 2 – Binário 3 – Kanji
szCodigo	Código a ser impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0023 – int GerarQRCodeArquivo (char * szNomeArquivoDestino, char * szDados)*

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

Entradas:

szNomeArquivoDestino: nome do arquivo que será salvo o bitmap

szDados: dados que serão informados no QRCODE

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0024 – int ImprimeBitmap (char * szPathArquivo)*

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

Entradas:

szPathArquivo: Path completo do arquivo a ser impresso

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

RF0025 – int LeModelo (char * szModelo)

Retorna o modelo da impressora que está em uso

Entradas:

Nenhuma

Saídas:

szModelo: Modelo da impressora

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

RF0025 – int AcionaGaveta ()

Aciona a gaveta de dinheiro conectada na impressora

Entradas:

Nenhuma

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0026 – int ImprimirSAT (char *szPathArquivoXML, char szTipo)*

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

Entradas:

szPathArquivoXML: Path completo do arquivo xml

szTipo: C – Completo, S - Simplificado

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0026 – int ImprimirSAT_Cancelamento (char *szPathArquivoXML, char *szQrCodeVenda, char *szModoProducao)*

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

Entradas:

szPathArquivoXML: Path completo do arquivo xml

szQrCodeVenda: assinatura do QRCode do xml de venda

szModoProducao: S – Produção, N - testes

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Media	Alta	Alta

*RF0027 – int LinhasGuilhotina (char *szQtdeLinhas)*

Esta função permite configurar a quantidade de linhas que a impressora irá saltar antes de realizar o corte da guilhotina, caso esteja habilitada. Valor Padrão 3.

Entradas:

szQtdeLinhas: quantide de linhas (0...20..30)

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

RF0028 – int ConfiguraGuilhotina (int iHabilita, int qdteLinhas)

Esta função permite configurar a quantidade de linhas que a impressora irá saltar antes de realizar o corte da guilhotina, e habilita ou desabilita a guilhotina.

Entradas:

iHabilita: 0=Desabilita, 1=Habilita

qtdeLinhas: Quantidade linhas de avanço

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

RF0029 – int StatusGuilhotina ()

Esta função permite verificar se a guilhotina está habilitada.

Entradas:

Nenhuma

Saídas:

0 = Desabilitada

1 = Habilitada

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

RF0030 – int ModoGaveta (int iModo)

Esta função permite verificar trabalhar com o cabo da gaveta invertido ou não

Entradas:

iModo: 0 = Padrão, 1 = Invertido

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

RF0030 – int ModoGaveta (int iModo)

Esta função permite verificar trabalhar com o cabo da gaveta invertido ou não

Entradas:

iModo: 0 = Padrão, 1 = Invertido

Saídas:

Nenhuma

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

*RF0031 – int LeVersaoFirmware (char * szModelo)*

Esta função permite retornar a versão do firmware da impressora

Entradas:

Nenhum

Saídas:

Nenhum

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

*RF0032 – int LeDataVersaoFirmware (char * szModelo)*

Esta função permite retornar a data da versão do firmware da impressora

Entradas:

Nenhum

Saídas:

Nenhum

Retornos:

CIS_SUCESSO

CIS_ERRO

Versão	Dificuldade	Prioridade	Estabilidade
1	Baixa	Alta	Alta

Requisitos não funcionais

Software

RNF0001 – Definição de Retornos

CIS_SUCESSO – 0

CIS_ERRO – 1

RNF0002 – Captura de informações da impressora fiscal

- Modelo da impressora
- Tampa aberta
- Fim de papel
- Pouco Papel
- Sensor de gaveta aberta

RNF0003 – Desenvolver lib/algorithmo para geração do QRCODE

Para atender o requisito de gerar o QRCode para arquivo, deve-se desenvolver uma rotina para gerar o QRCode em modo Bitmap.