

# Manual da DLL ELGIN

Revisão 1.0.0.5

## CONTATOS

Rua Barão de Campinas, 305 - Centro São Paulo - SP - Brasil

CEP 01201-901

Tel.: (55) (11) 3383-5555 (SP)

Tel.: (55) 0800 703 54 46 (Outras localidades)

Fax.: (55) (11) 3383-5851

Fórum Elgin: <http://blog.elgin.com.br>

Apoio ao Desenvolvedor: 0800 707 98 90

[www.elgin.com.br](http://www.elgin.com.br)

## Tabela de Conteúdo

<b>Parte I</b>	<b>Introdução</b>	<b>7</b>
1	Impressoras .....	7
2	Firmware da impressora fiscal MFD .....	7
3	Histórico de Revisões .....	7
<b>Parte II</b>	<b>Informações Gerais</b>	<b>10</b>
1	Criptografia do GT .....	10
2	Inicializações Necessárias para o uso do ECF .....	10
3	Limites do ECF .....	11
4	Situações tributárias .....	12
5	Tabelas do ECF .....	13
6	DLL para wind paralela .....	13
<b>Parte III</b>	<b>DLL Elgin</b>	<b>14</b>
1	Conhecendo a DLL Elgin.dll .....	14
2	Funções da DLL .....	14
	<b>Funções de impressão de cheque</b> .....	<b>14</b>
	Elgin_ImprimeCheque.....	14
	Elgin_ImprimeChequeMFD.....	15
	Elgin_LeituraCheque.....	16
	Elgin_ProgramaMoedaPlural.....	16
	Elgin_ProgramaMoedaSingular.....	16
	Elgin_VerificaStatusCheque.....	17
	<b>Funções de Inicialização</b> .....	<b>17</b>
	Elgin_AlteraSimboloMoeda .....	17
	Elgin_AtivaDesativaVendaUmaLinhaMFD.....	18
	Elgin_AumentaDescricaoItem .....	18
	Elgin_EspacoEntreLinhas.....	19
	Elgin_LinhasEntreCupons.....	19
	Elgin_NomeiaDepartamento.....	20
	Elgin_NomeiaTotalizadorNaoSujeitoCMS.....	20
	Elgin_ProgramaAliquota.....	21
	Elgin_ProgramaArredondamento .....	21
	Elgin_ProgramaBaudRate.....	22
	Elgin_ProgramaHorarioVerao.....	22
	Elgin_ProgramaTruncamento.....	23
	<b>Funções de Cupom Fiscal</b> .....	<b>23</b>
	Elgin_AbreCupom.....	23
	Elgin_CancelaCupom.....	24
	Elgin_CancelaItemAnterior.....	24
	Elgin_CancelaItemGenerico.....	24
	Elgin_EfetuaFormaPagamento.....	25
	Elgin_EfetuaFormaPagamentoDescricaoForma.....	25
	Elgin_EstornoFormasPagamento.....	26
	Elgin_FechaCupom.....	27
	Elgin_FechaCupomResumido.....	27
	Elgin_IniciaFechamentoCupom.....	28
	Elgin_ResetaImpressora.....	29
	Elgin_TerminaFechamentoCupom.....	29
	Elgin_Vendeltem.....	30

Elgin_VendeItemDepartamento.....	31
<b>Funções de Relatórios .....</b>	<b>32</b>
Elgin_AbreRelatorioGerencial.....	32
Elgin_FechaRelatorioGerencial.....	32
Elgin_LeituraMemoriaFiscalData.....	32
Elgin_LeituraMemoriaFiscalReducao.....	33
Elgin_LeituraMemoriaFiscalSerialData.....	34
Elgin_LeituraMemoriaFiscalSerialReducao.....	34
Elgin_LeituraX.....	35
Elgin_LeNomeRelatorioGerencial.....	35
Elgin_ReducacaoZ.....	36
Elgin_RelatorioGerencial.....	36
<b>Funções de Operações não Fiscais .....</b>	<b>37</b>
Elgin_AbreComprovanteNaoFiscalVinculado.....	37
Elgin_CancelaAcrescimoNaoFiscalMFD.....	38
Elgin_CancelaItemNaoFiscalMFD.....	38
Elgin_FechaComprovanteNaoFiscalVinculado.....	38
Elgin_RecebimentoNaoFiscal.....	39
Elgin_Sangria.....	39
Elgin_Suprimimento.....	40
Elgin_UsaComprovanteNaoFiscalVinculado.....	40
<b>Funções de Informações da impressora .....</b>	<b>41</b>
Elgin_Acrescimos.....	41
Elgin_Cancelamentos.....	41
Elgin_CGC_IE.....	42
Elgin_ClicheProprietario.....	42
Elgin_CNPJ_IE.....	43
Elgin_ContadoresTotalizadoresNaoFiscais.....	43
Elgin_DadosUltimaReducao.....	44
Elgin_DataHoraImpressora.....	44
Elgin_DataHoraReducao.....	45
Elgin_DataHoraSoftwareBasico.....	45
Elgin_DataMovimento.....	46
Elgin_Descontos.....	46
Elgin_FlagsFiscais.....	47
Elgin_FlagsFiscaisStr.....	47
Elgin_GrandeTotal.....	48
Elgin_LeIndicadores.....	49
Elgin_LeituraXSerial.....	49
Elgin_MinutosImprimindo.....	50
Elgin_MinutosLigada.....	50
Elgin_NumeroCaixa.....	51
Elgin_NumeroCupom.....	51
Elgin_NumeroCuponsCancelados.....	52
Elgin_NumeroIntervencoes.....	52
Elgin_NumeroLoja.....	52
Elgin_NumeroOperacoesNaoFiscais.....	53
Elgin_NumeroReducoes.....	53
Elgin_NumeroSerie.....	54
Elgin_NumeroSubstituicoesProprietario.....	54
Elgin_RetornoAliquotas.....	54
Elgin_RetornoImpressora.....	55
Elgin_SimboloMoeda.....	55
Elgin_SubTotal.....	56
Elgin_TotalDiaTroco.....	56
Elgin_TotalDocTroco.....	57
Elgin_UltimoItemVendido.....	57
Elgin_ValorFormaPagamento.....	58

Elgin_ValorPagoUltimoCupom.....	58
Elgin_ValorTotalizadorNaoFiscal.....	59
Elgin_VendaBruta.....	59
Elgin_VendaLiquida.....	59
Elgin_VerificaAliquotasICMS.....	60
Elgin_VerificaAliquotasIss.....	60
Elgin_VerificaDepartamentos.....	61
Elgin_VerificaEstadoImpressora.....	62
Elgin_VerificaEstadoImpressoraStr.....	62
Elgin_VerificaFormasPagamento.....	63
Elgin_VerificaIndiceAliquotasICMS.....	64
Elgin_VerificaIndiceAliquotasIss.....	64
Elgin_VerificaModoOperacao.....	65
Elgin_VerificaRecebimentoNaoFiscal.....	65
Elgin_VerificaTipoImpressora.....	66
Elgin_VerificaTipoImpressoraStr.....	67
Elgin_VerificaTotalizadoresNaoFiscais.....	67
Elgin_VerificaTotalizadoresParciais.....	68
Elgin_VerificaTruncamento.....	69
Elgin_VerificaZPendente.....	69
Elgin_VersaoFirmware.....	69
<b>Funções de Autenticação e Gaveta de dinheiro .....</b>	<b>70</b>
Elgin_AcionaGaveta.....	70
Elgin_VerificaEstadoGaveta.....	70
Elgin_VerificaEstadoGavetaStr.....	71
<b>Funções para impressoras MFD .....</b>	<b>71</b>
Elgin_AbreComprovanteNaoFiscalVinculadoMFD.....	71
Elgin_AbreCupomMFD.....	72
Elgin_AbreRecebimentoNaoFiscalMFD.....	73
Elgin_AbreRelatorioGerencialMFD.....	73
Elgin_AcrescimoDescontoItemMFD.....	74
Elgin_AcrescimoDescontoSubtotalMFD.....	74
Elgin_AcrescimoDescontoSubtotalRecebimentoMFD.....	75
Elgin_AcrescimoItemNaoFiscalMFD.....	75
Elgin_CancelaAcrescimoDescontoItemMFD.....	76
Elgin_CancelaAcrescimoDescontoSubtotalMFD.....	76
Elgin_CancelaAcrescimoDescontoSubtotalRecebimentoMFD.....	77
Elgin_CancelaCupomMFD.....	77
Elgin_CancelaRecebimentoNaoFiscalMFD.....	78
Elgin_CNPJMFD.....	78
Elgin_ComprovantesNaoFiscaisNaoEmitidosMFD.....	79
Elgin_ContadorComprovantesCreditoMFD.....	79
Elgin_ContadorCupomFiscalMFD.....	80
Elgin_ContadoresTotalizadoresNaoFiscaisMFD.....	80
Elgin_ContadorFitaDetalheMFD.....	81
Elgin_ContadorOperacoesNaoFiscaisCanceladasMFD.....	81
Elgin_ContadorRelatoriosGerenciaisMFD.....	81
Elgin_CupomAdicionalMFD.....	82
Elgin_DadosUltimaReducaoMFD.....	82
Elgin_DataHoraUltimoDocumentoMFD.....	84
Elgin_DataMovimentoUltimaReducaoMFD.....	84
Elgin_DownloadMFD.....	84
Elgin_EfetuaFormaPagamentoMFD.....	85
Elgin_EfetuaRecebimentoNaoFiscalMFD.....	86
Elgin_EstornoNaoFiscalVinculadoMFD.....	86
Elgin_FechaRecebimentoNaoFiscalMFD.....	87
Elgin_FormatoDadosMFD.....	87
Elgin_GrandeTotalUltimaReducaoMFD.....	88

Elgin_HabilitaDesabilitaRetornoEstendidoMFD.....	89
Elgin_IniciaFechamentoCupomMFD.....	92
Elgin_IniciaFechamentoRecebimentoNaoFiscalMFD.....	93
Elgin_InicioFimCOOsMFD.....	94
Elgin_InicioFimGTsMFD.....	94
Elgin_InscricaoEstadualMFD.....	95
Elgin_InscricaoMunicipalMFD.....	95
Elgin_MapaResumoMFD.....	96
Elgin_MarcaModeloTipoImpressoraMFD.....	97
Elgin_MinutosEmitindoDocumentosFiscaisMFD.....	97
Elgin_NomeiaRelatorioGerencialMFD.....	98
Elgin_NumeroSerieMemoriaMFD.....	98
Elgin_PercentualLivreMFD.....	98
Elgin_ProgramaFormaPagamentoMFD.....	99
Elgin_ReducoesRestantesMFD.....	99
Elgin_RegistrosTipo60.....	100
Elgin_ReimpressaoNaoFiscalVinculadoMFD.....	101
Elgin_RelatorioTipo60AnaliticoMFD.....	102
Elgin_SegundaViaNaoFiscalVinculadoMFD.....	103
Elgin_StatusEstendidoMFD.....	103
Elgin_SubTotalComprovanteNaoFiscalMFD.....	104
Elgin_TamanhoTotalMFD.....	104
Elgin_TempoOperacionalMFD.....	105
Elgin_TotalLivreMFD.....	105
Elgin_UsaRelatorioGerencialMFD.....	106
Elgin_ValorFormaPagamentoMFD.....	106
Elgin_ValorTotalizadorNaoFiscalMFD.....	107
Elgin_VerificaEstadoImpressoraMFD.....	107
Elgin_VerificaFormasPagamentoMFD.....	108
Elgin_VerificaRecebimentoNaoFiscalMFD.....	108
Elgin_VerificaRelatorioGerencialMFD.....	109
Elgin_VerificaSensorPoucoPapelMFD.....	110
Elgin_VerificaTotalizadoresNaoFiscaisMFD.....	110
Elgin_VerificaTotalizadoresParciaisMFD.....	111
<b>Autenticação .....</b>	<b>112</b>
Elgin_Autenticacao.....	112
Elgin_AutenticacaoMFD.....	113
<b>Funções de Código de Barras .....</b>	<b>113</b>
Configura Código de Barras.....	113
Elgin_CodigoBarrasCODABARMFD.....	114
Elgin_CodigoBarrasCODE39MFD.....	115
Elgin_CodigoBarrasEAN13MFD.....	116
Elgin_CodigoBarrasEAN8MFD.....	116
Elgin_CodigoBarrasITFMFD.....	117
Elgin_CodigoBarrasUPCAMFD.....	117
Elgin_CodigoBarrasUPCEMFD.....	118
<b>Outras Funções .....</b>	<b>118</b>
Elgin_AberturaDoDia.....	118
Elgin_AbrePortaSerial.....	119
Elgin_AcionaGuilhotinaMFD.....	119
Elgin_FechamentoDoDia.....	120
Elgin_FechaPortaSerial.....	120
Elgin_IdentificaConsumidor.....	120
Elgin_ImprimeConfiguracoesImpressora.....	121
Elgin_ImprimeDepartamentos.....	122
Elgin_IncluiCidadeFavorecido.....	122
Elgin_LeArquivoRetorno.....	123
Elgin_LeBalanca.....	123

Elgin_MapaResumo.....	124
Elgin_ModeloImpressora.....	125
Elgin_NumeroSerieCriptografado.....	126
Elgin_NumeroSerieDescriptografado.....	126
Elgin_ProgramaCaracterAutenticacao.....	126
Elgin_ProgramaIdAplicativoMFD.....	127
Elgin_ProgramaOperador.....	127
Elgin_RelatorioTipo60Analitico.....	128
Elgin_RelatorioTipo60Mestre.....	129
Elgin_VerificaImpressoraLigada.....	130
Elgin_VerificaReducaoZAutomatica.....	130
RFD_ConvertedaMFD.....	130
RFD_ConvertedaMFDData.....	131
Elgin_RelatorioSintegraMFD.....	131
Elgin_LeParametrosPAF.....	132
Elgin_LeCodigoNacionalIdentificacaoECF.....	133
Elgin_ExecutaComando.....	134
Elgin_ExecutaLeitura.....	135
Elgin_TotalcmsCupom.....	135
<b>Funções para impressora Wind .....</b>	<b>136</b>
Wind_AcionaGaveta.....	136
Wind_AcionaGuilhotina.....	136
Wind_AcionaGuilhotinaParcial.....	137
Wind_AjustaLarguraPapel.....	137
Wind_ConfiguraCodigoBarras.....	138
Wind_EnviaBuffer.....	138
Wind_EnviaBufferFormatado.....	139
Wind_EnviaComando.....	140
Wind_ImprimeCodigoBarrasCODABAR.....	140
Wind_ImprimeCodigoBarrasCODE128.....	141
Wind_ImprimeCodigoBarrasCODE39.....	141
Wind_ImprimeCodigoBarrasCODE93.....	142
Wind_ImprimeCodigoBarrasEAN13.....	143
Wind_ImprimeCodigoBarrasEAN8.....	143
Wind_ImprimeCodigoBarrasISBN.....	144
Wind_ImprimeCodigoBarrasITF.....	145
Wind_ImprimeCodigoBarrasMSI.....	145
Wind_ImprimeCodigoBarrasPDF417.....	146
Wind_ImprimeCodigoBarrasPLESSEY.....	147
Wind_ImprimeCodigoBarrasUPCA.....	148
Wind_ImprimeCodigoBarrasUPCE.....	148
Wind_VerificaEstadoGaveta.....	149
Wind_VerificaFimPapel.....	149
Wind_VerificaOnLine.....	150
Wind_VerificaPoucoPapel.....	150
<b>Parte IV Configuração do arquivo elgin.ini .....</b>	<b>150</b>
<b>Parte V Configuração do arquivo cheques.ini .....</b>	<b>152</b>
<b>Parte VI Sequência de comandos .....</b>	<b>153</b>
<b>1 Venda Completa .....</b>	<b>153</b>
<b>Parte VII Procedimentos para usar o Emulador Log2 com a DLL Elgin .....</b>	<b>154</b>

---

<b>Parte VIII Retornos da Impressora</b>	<b>156</b>
<b>Parte IX Retorno Impressora (ACK, ST1 e ST2)</b>	<b>160</b>
<b>Parte X Comprovante Não Fiscal Vinculado</b>	<b>162</b>
<b>Parte XI Declaração das Funções da Elgin.dll em Delphi</b>	<b>162</b>
<b>Parte XII Declaração das Funções da Elgin.dll em VB</b>	<b>168</b>

## 1 Introdução

### 1.1 Impressoras

A Elgin oferece ao mercado a maior gama de impressoras para automação comercial, seja para ambientes menores, com pouco fluxo de clientes ou para supermercados com extenso volume de transações.

As aplicações de automação comercial na parte do PDV (ponto de venda) farão a utilização da dll Elgin, que será apresentada neste help para a comunicação com as seguintes impressoras Elgin:

- IF MFD 6000 TH 2E
- IF MFD FIT 1E
- X5
- Wind serial e paralela

### 1.2 Firmware da impressora fiscal MFD

#### Modos de Operação

O ECF possui 2 (dois) modos de operação: Modo de Intervenção Técnica (MIT) e Modo Normal de Operação. Por motivo de legislação, não existe mais o Modo Treinamento nestes equipamentos.

#### Modo de intervenção Técnica

Do ponto de vista do *software* básico do ECF, diz-se Modo de Intervenção Técnica (**MIT**) o modo onde operações especiais, como configurações, redefinições, acertos de relógio e etc, são permitidas. Em contrapartida, algumas outras operações fiscais são bloqueadas, tais como: emissão de cupons fiscais e não fiscais, além de outros.

Para entrar em MIT, o *jumper* específico para esta função, posicionado na placa fiscal do ECF, deve ser retirado com o equipamento desligado. Ao ser ligado sem o *jumper*, o *software* básico reconhecerá o estado de MIT. Caso haja algo que impossibilite a entrada em MIT, uma mensagem será impressa na bobina indicando o motivo. Automaticamente, ao ligar o ECF sem o *jumper* para entrar em MIT, o *software* básico faz as seguintes verificações:

- Se o dia estiver aberto, é impresso um menu na bobina, solicitando ao usuário proceder o acerto do relógio em mais ou menos 5 minutos com relação ao relógio atual do ECF.
- Se o dia estiver aberto, também é impresso um documento de fechamento diário automaticamente ([Redução Z](#)), sendo que neste caso, será possível continuar emitindo cupons fiscais após o MIT, pois este tipo de Redução Z não fecha o dia.
- Independente da situação de dia aberto, um relatório fiscal será impresso automaticamente ([Leitura X](#)), indicando "ENTRADA EM INTERVENCAO".

O ECF somente sairá de MIT para o modo normal de operação após ser colocado o *jumper* de intervenção com o mesmo desligado e depois ligá-lo. Ao ser ligado, o ECF emitirá uma Leitura X indicando "SAÍDA DE INTERVENÇÃO". Caso haja algum problema que impeça a saída de MIT, um erro indicando o motivo será impresso na bobina.

#### Modo Normal de Operação

O Modo Normal de Operação é também chamado de Modo Fiscal. Neste modo, as operações normais de emissão de cupons são permitidas, tais como: venda de itens, cupons não fiscais, relatórios gerenciais, emissão de vias de crédito/débito, cancelamentos de documentos fiscais e não-fiscais, etc.

O Modo Normal de Operação e Modo de Intervenção Técnica são exclusivos, ou seja, o ECF estará no Modo Normal de Operação sempre que não estiver em MIT.

### 1.3 Histórico de Revisões

21/03/2007 - Foram acrescentados alguns tópicos (informações sobre o arquivo .ini e retornos da impressora).

25/01/2007 - Criação do manual/help.

02/04/2007 - Inserida a função Elgin\_EstornoNaoFiscalVinculadoMFD.

09/04/2007 - Correção da função Elgin\_RetornoAliquotas, acréscimo da função Elgin\_AtivaDesativaVendaUmaLinhaMFD e Elgin\_AumentaDescricaoItem.

- 24/04/2007 - Inserido o retorno -50 das funções, alguns comentários nas funções da Wind, foram removidos tópicos repetidos e foram inseridas as funções pendentes nos tópicos de Declaração das Funções da Elgin.dll em Delphi e Declaração das Funções da Elgin.dll em VB.
- 07/05/2007 - Foram feitas correções de conteúdo e exemplos. Foi inserida a função Wind\_AcionaGuilhotinaParcial.
- 06/06/2007 - O manual foi atualizado de acordo com a versão da dll 0.0.0.28. Dentre as alterações da dll foi retirado o parâmetro da função **Elgin\_AbreRelatorioGerencial** e acrescentada a nova declaração. A função **Elgin\_AberturaDoDia** foi atualizada para só imprimir cupom de suprimento se o parâmetro valor for diferente de 0.
- 15/06/2007 - Incluída função Elgin\_DownloadMFD e Elgin\_FormatoDadosMFD.
- 06/07/2007 - Incluída a função Wind\_VerificaOnLine, Elgin\_ImprimeChequeMFD e Elgin\_LeituraCheque. Inserido o retorno "-10 - Mensagem: Impressora offline ou desligada" nas funções da impressora Wind. Alterada informações erradas da função Elgin\_DadosUltimaReducaoMFD e Elgin\_DadosUltimaReducao. Inserido no Elgin.ini o comentário sobre o corte automatico da IF 6000 TH 2E.
- 13/07/2007 - Inclusão de retornos da função Wind\_VerificaEstadoGaveta (1-gaveta aberta e 2-gaveta fechada), alteração na descrição da função Elgin\_RecebimentoNaoFiscal.
- 17/07/2007 - Inclusão do tópico que detalha o arquivo Cheques.ini.
- 26/07/2007 - Inserido o parâmetro HabilitaCorte no arquivo ELGIN.INI.
- 06/08/2007 - Inserida as funções Elgin\_AutenticacaoMFD, Elgin\_Autenticacao, Elgin\_CodigoBarrasCODABARMFD, Elgin\_CodigoBarrasCODE39MFD, Elgin\_CodigoBarrasEAN13MFD, Elgin\_CodigoBarrasEAN8MFD, Elgin\_CodigoBarrasITFMFD, Elgin\_CodigoBarrasUPCAMFD, Elgin\_CodigoBarrasUPCEMFD, Elgin\_ConfiguraCodigoBarrasMFD. Corrigidas as funções de abertura do cupom.
- 22/08/2007 - Inserida a função **RFD\_ConvertedaMFD**(CRZ:String) aplicável a CAT52 ou Ato COTEPE 17/04. Inserido o parâmetro GERA\_RFD\_REDZ do Elgin.ini.
- 12/09/2007 - Inserido no arquivo Elgin.ini o parâmetro TimeOutRedZ. Inserida observação na Elgin\_ProgramaFormaPagamentoMFD. Alterada a localização do arquivo Elgin.ini. Inseridas as funções Elgin\_IdentificaConsumidor, Elgin\_ProgramaIdAplicativoMFD, Elgin\_ModeloImpressora, Elgin\_VerificaReducaoZAutomatica, Elgin\_AcionaGuilhotinaMFD, Elgin\_NumeroSerieCriptografado e Elgin\_NumeroSerieDescriptografado.
- 19/09/2007 - Inserção do parâmetro Endereco no Elgin.INI.
- 27/09/2007 - Inclusão das funções Elgin\_ProgramaCaracterAutenticacao e Elgin\_IncluiCidadeFavorecido. Inclusão no arquivo Elgin.ini do parâmetro ModoGaveta.
- 15/10/2007 - Inclusão no Elgin.ini das informações referentes ao registro E00 da CAT60.
- 18/10/2007 - Inclusão da função Elgin\_ProgramaOperador.
- 19/10/2007 - Alterada a Elgin\_ImprimeChequeMFD que não imprimirá no verso e Elgin\_EfetuaFormaPagamentoMFD que não mostra descrição.
- 13/11/2007 - Foi adicionado o parâmetro NumPkt=133 no arquivo Elgin.INI.
- 10/12/2007 - Foi adicionado o parâmetro PathRFD no arquivo Elgin.INI.
- 30/01/2008 - Foram adicionadas as funções Elgin\_DataHoraSoftwareBasico e Elgin\_ProgramaBaudRate.
- 03/03/2008 - Revisão nos arquivos de declarações em Delphi e VB, Adicionando os códigos das alíquotas de Isento, não tributável e substituição tributária nas funções Elgin\_VendeItem e Elgin\_VendeItemDepartamento, Adicionado tópicos com a seqüência completa dos comandos para cupom fiscal e cupom não fiscal vinculado.
- 28/03/2008 - Correção de informações nas funções Elgin\_NomeiaRelatorioGerencialMFD, Elgin\_IniciaFechamentoCupomMFD e Elgin\_IniciaFechamentoRecebimentoNaoFiscalMFD.
- 15/05/2008 - Implementada a função para a geração do Sintegra de saída Elgin\_RelatorioSintegraMFD e a função RFD\_ConvertedaMFDData.
- 27/05/2008 - Corrigidos problemas de formatação dos dados da MFD quando a Descrição do PLU possuía ASPAS e retirado do E21 o registrador TROCO.
- 09/06/2008 - Corrigidos problemas de formatação dos dados da MFD quando a Descrição do PLU possuía o caractere especial do protocolo de comunicação - Ponto e Vírgula (;) e Retirada duplicidade de declaração da função "Elgin\_RelatorioSintegraMFD".
- 11/06/2008 - Ao colocar em ELGIN.INI, "ModeloImp=ELGIN\_X5", o modelo assumido é a ELGIN\_X5, sendo assim compatível com a ELGIN FIT. Colocada proteção para tentar evitar "Exception Econverterror" quando a serial recebe muita sujeira.
- 16/06/2008 - Corrigido erro ao gerar arquivo do ATO COTEPE referente a unidade do produto.
- 10/07/2008 - Corrigido erro ao gerar arquivo do ATO COTEPE referente as formas de pagamento em um cupom fiscal.
- 11/07/2008 - Corrigido erro ao gerar arquivo do ATO COTEPE referente ao problema de formatação do cabeçalho quando o mesmo tem linhas em branco.
- 17/07/2008 - Adicionanda uma observação, quanto as casas decimais, na função

Elgin\_EfetuaRecebimentoNaoFiscalMFD.

22/07/2008 - Corrigido problema na situação em que o E14 trazia a informação do cliente identificado no cupom de TEF imediatamente anterior.

22/07/2008 - Corrigido problema de erro de estouro quando QTDE X Preço atinge valores elevados.

24/07/2008 - Corrigido problema quando a descrição do produto se assemelha a uma linha de QTDE X PRECO VALOR Ex: 001 0000000000001 CAPA P/ COLCHAO PARA SOLTEIRO 1,88 x 0,90 alt 0,20 1,000Un x 10,00 I1 10,00°

24/07/2008 - Corrigido problema que ao dar acrescimo ou desconto em item, os proximos itens traziam a mesma informação.

30/07/2008 - Corrigido problema quando o aplicativo envia virgula (",") no campo Unidade Ex:048

2023500003713 MUSSARELA FT KG 0,312, x 11,90 T01 A 3,71°

30/07/2008 - Alterada a rotina Wind\_EnviaComando para impressora Paralela.

06/08/2008 - Corrigido os problemas na Consolidação dos Arquivos do Ato COTEPE em algumas situações.

21/08/2008 - Corrigida situação que apresentava Mensagem de Erro ESC e criados os flags na Seção [Sistema] de TIMEOUTCOMANDOS e CHECACTS.

21/08/2008 - Corrigida situação de identificação das Leituras de Memória Fiscal na montagem do Ato Cotepe.

17/09/2008 - Correção na identificação do CNPJ/CPF quando utilizado TC-1000, pois não utiliza a função "IdentificaConsumidor" e ChecaCTS passa a ter default =1.

13/11/2008 - Correção ao identificar algum erro na Descrição do Produto na venda de Item, a descrição no registro E15 é colocada como "ERRO DESCRICAO".

13/11/2008 - Corrigido a abertura de portas seriais acima da COM9 (COM10,COM11,etc).

13/11/2008 - Acrescentadas as rotinas "Elgin\_ExecutaComando" e "Elgin\_ExecutaLeitura"

13/11/2008 - Corrigido a passagem de parâmetros para a rotina "Elgin\_CancelaAcrescimoDescontoItemMFD".

20/11/2008 - Adicionado na seção [Sistema] o flag CorrigeRelogioZ; Este flag tem como default o valor 1 e deve ser utilizado para acerto automático do relógio durante a ReduçãoZ, usando como referência a hora do PC, lembrando que o ECF só permite o acerto de 5 em 5 minutos.

20/11/2008 - Corrigido problema de não incluir a letra de MF Adicional nos registros do Ato Cotepe.

01/12/2008 - Adicionado o reconhecimento do termo "CPF do adquirente:" juntamente com "CNPJ/CPF CONSUMIDOR:" na identificação do consumidor no Cupom Fiscal.

18/03/2009 - Corrigido retorno para Elgin\_DataHoraSoftwareBasico.

18/03/2009 - Corrigido retorno da Elgin\_TamanhoTotalMFD quando tamanho da Memoria MFD é maior ou igual a 128MB.

18/03/2009 - Corrigida no sintegra consolidação do 60R e ordenação do arquivo.

18/03/2009 - Criada a função function Elgin\_LeBalanca(Porta,Baud,Valor:PChar):integer;stdcall; para comunicação com toda a linha de balancas Elgin.

18/03/2009 - Adicionado o reconhecimento do termo "Cliente:" na identificação do consumidor no Cupom Fiscal.

22/05/2009 - Corrigido a situação que após fechar gerencial e reiniciar a DLL e executar Abertura de Cupom, a DLL tentava fechar o Relatorio Gerencial e apresentava erro.

28/07/2009 - Corrigido comando "Elgin\_DadosUltimaReducaoMFD" que agora traz os valores de Cancelamentos, Descontos e Acréscimos ICMS e ISSSQN separados e não consolidados como nas versões anteriores.

26/08/2009 - Otimizadas rotinas de Leitura do Ato Cotepe e Sintegra via texto.

26/08/2009 - Corrigido problema de estouro de inteiros na Leitura do Sintegra.

26/08/2009 - Implementada a Conversao de arquivos da Leitura Binario em Sintegra (somente para Gera1704).

02/09/2009 - Corrigida falha ao Ler Diversos parâmetros caso o ECF fosse desligado após a execução do comando "Elgin\_DadosUltimaReducao".

25/09/2009 - Corrigida falhas de retorno dos comandos "Elgin\_Sangria" e "Elgin\_Suprimento" que retornavam código 1 mesmo se a execução não fosse realizada com sucesso.

21/10/2009 - Realizadas alterações para poder criar os arquivos TDM E MFD utilizando o Storage (Alteração conjugada com a versão 1.53 do Gera1704P.exe).

21/10/2009 - Realizada correção na geração do registro E01 ao criar os arquivos MFD e TDM através do Storage.

21/10/2009 - Corrigido retorno da função "Elgin\_RelatorioSintegraMFD" que em algumas situações retornava sucesso (1) mesmo havendo Falha de Comunicação.

30/10/2009 - Criada a função "Elgin\_LeCodigoNacionalIdentificacaoECF" que traz a leitura doCodigo CNI conforme tabela abaixo:

Modelo	Versao	CNI
=====	=====	=====
Elgin Fit	01.00.08	140702

IF 6000TH	01.00.08	141102
FX7	01.00.07	141201
X5	01.00.07	141301
ZPM/1 Fit Logger	03.03.04	491207
ZPM/2EFC Logger	03.03.04	491307

30/10/2009 - Criada a função "Elgin\_LeParametrosPAF" que faz a Leitura automatica de alguns parametros de inicialização do PAF.

23/11/2009 - Adicionadas funções para o CUPOM MANIA (RJ).

a) Criado o flag 'HabilitaICMSCupom' na area [Sistema] que permite o controle do valor do ICMS do ultimo cupom através da função, A variavel deve ser preparada com 14 bytes. Caso em [ECF] o estado seja RJ, a impressão do cupom mania é feita automaticamente.

16/03/2010 - Correções no arquivo do Sintegra, que nos registros 60I, quando o item era cancelado nao trazia no campo 12 a expressao "CANC";

12/04/2010 - Criada a função para emissão automatica da LMF no inicio do mes, referente ao mes anterior.

10/05/2010 - Correções na geração do arquivo Sintegra Binário, campos Valor dos registros 60D era gravado mesmo estando zerado.

21/05/2010 - Correções na geração do arquivo Sintegra Binário:

a) Em situações em que o valor liquido do item estivesse zerado (Ex: Venda e depois cancelamento do item) o campos "Valor" dos registros 60D eram gravados;

b) O valor do imposto nos registros 60I estava sendo calculado de forma errada;

c) Nao estava sendo gravado no 60A o valor de ISS corretamente (IS1,FS1,NS1) .

06/07/2010 - Incluído o modelo de ECF ZPM-200 nas rotinas de criação dos arquivos para o Gera1704 nas funções correspondente na DLL.

## 2 Informações Gerais

### 2.1 Criptografia do GT

A obtenção da criptografia do GT deve ser obtida no final do documento de Leitura da Memória Fiscal (impresso ou via serial).

### 2.2 Inicializações Necessárias para o uso do ECF

O ECF necessita que sejam inicializados alguns parâmetros para o seu funcionamento em modo fiscal. Estes parâmetros são:

**1) Dados de Identificação do Proprietário:** Para a emissão de cupons fiscais e não fiscais é necessário a definição do CNPJ e da Inscrição Estadual do proprietário. A Inscrição Municipal(\*) é opcional. O número de definições de proprietários é limitado pelo ECF. A identificação do proprietário exige uma senha para que a operação seja efetivada.

**2) Clichê:** Armazena os dados do clichê na impressora fiscal: nome fantasia, razão social, endereço, telefone, número do ECF, número da loja, etc. Pode ser alterado independentemente dos dados de identificação do proprietário.

**3) Alíquotas (ou Situações Tributárias):** A carga da tabela de alíquotas é necessária para os comandos de venda de item. Como existem alíquotas pré-definidas, este comando não é fundamental para emitir cupons fiscais de venda. As alíquotas programáveis podem ser configuradas tanto para ICMS quanto para ISSQN.

**4) Meios de Pagamento:** Define os meios de pagamento necessários para finalização de um cupom de venda ou cupom não fiscal. Como existe um meio de pagamento pré-definido, este comando não é fundamental para emissão de cupons. No momento da definição de cada meio de pagamento deve ser informado se permite ou não a emissão de documento vinculado (diz-se documento de crédito/débito).

**5) Não fiscais e Gerenciais:** Caso seja necessário, deve-se carregar os códigos dos totalizadores não fiscais e dos gerenciais para a emissão de documentos não fiscais.

**6) Departamentos:** Opcionalmente pode-se carregar os departamentos para acumular valores na venda de itens em cupom fiscal. Os valores acumulados por departamento podem ser impressos apenas em relatórios gerenciais.

**7) Definição da Moeda:** O dados da moeda, tais como símbolo, nome singular e plural, são valores pré-definidos pelo software básico, porém não são de programação obrigatória: "R\$", "Real" e "Reais", respectivamente.

Os parâmetros 1 e 2 só podem ser definidos via intervenção técnica e após o fechamento diário.

Os parâmetros 3, 4, 5 e 6 podem ser definidos sem intervenção técnica, porém será necessário intervenção técnica quando houver a necessidade de alteração de pelo menos um atributo destes itens. A redefinição destes parâmetros somente pode ocorrer após o fechamento diário.

O parâmetro 7 pode ser definido e redefinido sempre que o ECF não estiver durante a emissão de um cupom.

Após realizadas estas inicializações, a impressora fiscal estará apta a funcionar para as atividades fiscais.

#### NOTA

A impressora fiscal não necessita a carga da tabela de criptografia para ser inicializada. A tabela é gerada automaticamente a partir do software básico quando ocorre a carga do cabeçalho.

#### OBSERVAÇÕES

(\*) na versão 03.03.04 a Inscrição Municipal pode ser informada em um momento posterior a Lacração Inicial desde que tenha sido deixada sem preenchimento algum na lacração Inicial.

## 2.3 Limites do ECF

Os limites dos valores armazenados pelo ECF são definidos através de seus tipos de dados.

Além do tipo de dado, muitos Registradores (ou variáveis) ainda são restritos a limites mais específicos. Abaixo são relacionados os limites de valores gerenciados pelo ECF:

Identificação da Variável	Limite Máximo
Contador de Reduções Z (CRZ)	3.196
Contador de Reinício de Operação (CRO)	200
Definição de Proprietários	20*
Alterações de Software Básico	10
Emissões de Fita Detalhe Impressa por Intervenção Técnica	2

\* Nas versões de Software Básico 01.00.00 e 03.03.00 são permitidos cadastros de até 20 proprietários. A partir das versões 01.00.07 e 03.03.02 é permitido apenas um proprietário.

Identificação da Variável	Apartir da versão 03.03.00/ 01.00.00	
	Dígitos	Formato
Preço por Item	8	999.999,99
Capacidade Máxima por Registro de Item	11	999.999.999,99
Quantidade por item	7	9.999,999
Totalizadores parciais	13	99.999.999.999,99
GT	18	9.999.999.999.999.999,99
Venda Bruta	14	999.999.999.999,99
Venda Líquida	14	999.999.999.999,99
Número de Cupons Cancelados (CFC)	4	9999
Contador de Ordem de Operação (COO)	6	999999
Geral de Operação Não-Fiscal (GNF)	6	999999
Contador de Cupom Fiscal (CCF)	6	999999
Comprovante de Crédito/Débito (CDC)	4	9999
Geral de Relatórios Gerenciais (GRG)	6	999999
Contador de Reduções Z (CRZ)	4	9999
Geral de Oper. Não-Fiscal Canc. (NFC)	4	9999
Contador de Fita-Detalhe (CFD)	6	999999
Contador de Reinício de Operação (CRO)	3	999
Itens por Cupom Fiscal	3	999

## 2.4 Situações tributárias

Uma situação tributária (ou alíquota) é composta pelas seguintes informações:

Atributo	Descrição
CodAliquotaProgramavel	Código único que identifica uma alíquota
PercentualAliquota	Percentual associado ao código de alíquota programado
AliquotaICMS	Indica se a alíquota é de ICMS ( <i>Yes</i> ) ou ISSQN ( <i>No</i> )
DescricaoAliquota	Descrição opcional para a alíquota

Pode ser identificada pelo seu código ou pelo seu percentual e tipo, conjuntamente. A descrição de uma alíquota é um texto informativo opcional que a descreve, não possui nenhuma função específica, mas pode ser consultado com o comando LeAliquota.

A impressora fiscal gerencia uma tabela de alíquotas com as seguintes características:

- **16** (dezesseis) **alíquotas programáveis** (índices 00 a 15), através de comando, podendo ser configuradas como ICMS e/ou ISSQN.
- **Substituição tributária ICMS e ISSQN**, pré-configuradas com os índices -2 e -11, respectivamente.
- **Isenção ICMS e ISSQN**, pré-configuradas com os índices -3 e -12, respectivamente.
- **Não tributadas ICMS e ISSQN**, pré-configuradas com os índices -4 e -13, respectivamente.

É obrigatória a informação da situação tributária ao comandar a venda de um item. O registro da venda de item informa a situação tributária da seguinte forma:

Alíquotas ICMS programáveis: **TXX**, onde **XX** identifica o índice da alíquota

Alíquotas ISSQN programáveis: **SXX**, onde **XX** identifica o índice da alíquota

Alíquota de Substituição Tributária: **FX**, onde X é '1' para ICMS e 'S1' para ISSQN

Alíquota de Isenção: **IX**, onde X é '1' para ICMS e 'S1' para ISSQN

Alíquota de Isenção: **NX**, onde X é '1' para ICMS e 'S1' para ISSQN

Os totalizadores parciais de cada situação tributária programada são listados nos Relatórios Fiscais X e Z, nas seções *ICMS* e *ISSQN*. Também são informados os percentuais programados para cada situação tributária, bem como os valores de venda acumulados nestas alíquotas e o valor do imposto referente cada uma delas. Após a emissão de um fechamento - Redução Z, os valores acumulados nas alíquotas e os valores dos impostos são zerados para início do novo dia fiscal.

### IMPORTANTE

Uma alíquota somente pode ser redefinida ou excluída após o fechamento diário ([Redução Z](#)) e no estado de [intervenção técnica](#).

## 2.5 Tabelas do ECF

O software básico gerencia um conjunto de tabelas para controlar as operações fiscais e não fiscais. As tabelas e seus limites são os seguintes:

Nome da Tabela	Índice Mínimo	Índice Máximo	Programáveis	Pré-Definidos
Situações Tributárias	0	15	16	F1, I1, N1, FS1, IS1, NS1
Meios de Pagamentos	0	14	15	Dinheiro
Não Fiscais	0	14	15	-
Gerenciais*	0	19	20	-
Departamentos	1	19	19	Combustível

Os itens das tabelas que possuem valores pré-definidos são acessíveis através dos seguintes índices de tabela:

Nome da Tabela	Totalizador Pré-definido	Sigla	Índice
Meios de Pagamentos	Dinheiro	-	-2
Situações Tributárias	Substituição Tributária ICMS	F1	-2
Situações Tributárias	Isenção ICMS	I1	-3
Situações Tributárias	Não Tributada ICMS	N1	-4
Situações Tributárias	Substituição Tributária ISSQN	FS1	-11
Situações Tributárias	Isenção ISSQN	IS1	-12
Situações Tributárias	Não Tributada ISSQN	IN1	-13
Departamentos	Combustível	-	0

### NOTA

Nas versões 01.00.07 e 03.03.04 o gerencial de posição 99 é pré-definido e imprime alguns parâmetros da impressora.

## 2.6 DLL para wind paralela

Para programar para a wind paralela é necessário utilizar uma DLL auxiliar, a **inpout32.dll**. Além da DLL Elgin.

## 3 DLL Elgin

### 3.1 Conhecendo a DLL Elgin.dll

A Elgin.dll é uma DLL de comunicação que facilita o desenvolvimento de aplicações PDV (ponto de venda). É utilizada em ambiente Windows 32 bits, realizando a comunicação com as impressoras fiscais da Elgin.

Todas as funções desta DLL são funções de alto-nível.

Ao utilizar qualquer função desta DLL será apresentado na tela para o desenvolvedor todos os parâmetro que a função possui (caso possua) facilitando a sua utilização.

A DLL pode estar no diretório da aplicação ou no diretório do Windows (Ex: C:\WINDOWS) ou no diretório de sistema (Ex:C:\WINDOWS\SYSTEM32).

Exemplo:

C:\WINDOWS

C:\WINDOWS\SYSTEM

C:\WINNT\SYSTEM32

A dll Elgin é uma interface ( forma de comunicação ) entre o dispositivo e aplicação que está em desenvolvimento, faz uso da plataforma Windows 32 bits Os modelos de impressoras termicas compatíveis com DLL Elgin são:

- MFD FIT 1E ( 01.00.08 / 03.03.04 )
- X-5 ( 01.00.07 )
- IF 6000 TH ( 03.03.04 )
- WIND ( Serial / Paralela )

### 3.2 Funções da DLL

#### 3.2.1 Funções de impressão de cheque

##### 3.2.1.1 Elgin\_ImprimeCheque

Envia o comando de impressão de cheque para a impressora.

**Parâmetros:**

**Banco:** String com o Número do banco com 3 dígitos.

**Valor:** String com o Valor do cheque com até 14 dígitos.

**Favorecido:** String com o Favorecido com até 45 caracteres.

**Cidade:** String com a Cidade com até 27 caracteres.

**Data:** String com a Data no formato ddmmaa, dd/mm/aa, ddmmaaaa ou dd/mm/aaaa.

**Mensagem:** String com o Comentário de até 120 caracteres.

**Obs.:** Para utilizar esta função é necessário que exista o arquivo [CHEQUES.INI](#).

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-7: Banco não encontrado no arquivo Elgin.ini.

-50: Número de série inválido.

**Exemplos**

' Exemplo em Visual Basic

```
iResultado = Elgin_ImprimeCheque("353", "50,00", "Elgin", "Manaus", "10/09/06", "Bom para dia 01/10")
```

**// Exemplo em Delphi**

```
cCodBanco := '273';
cValor := '100,00';
cNominal := 'Elgin';
cCidade := 'Manaus';
cData := '10/09/06';
cMsg := 'Bom para dia 01/10';
iResultado := Elgin_ImprimeCheque( pchar( cCodBanco ), pchar( cValor ), pchar( cNominal ), pchar( cCidade ), pchar( cData ), pchar( cMsg ) );
```

### 3.2.1.2 Elgin\_ImprimeChequeMFD

Envia o comando de impressão de cheque para a impressora. Possibilita a impressão apenas na frente do Cheque.

**Parâmetros:**

**Banco:** String com o Número do banco com 3 dígitos.

**Valor:** String com o Valor do cheque com até 14 dígitos.

**Favorecido:** String com o Favorecido com até 45 caracteres.

**Cidade:** String com a Cidade com até 27 caracteres.

**Data:** String com a Data no formato ddmmaa, dd/mm/aa, ddmmaaaa ou dd/mm/aaaa.

**Mensagem:** String com o Comentário até 120 caracteres. A mensagem será impressa 1 (uma) linha após a cidade caso não indicada para impressão no verso.

**ImpressaoVerso:** String com o valor zero "0" para impressão na frente do cheque e o valor um "1" para impressão no verso.

**Linhas:** String com um valor entre 0 e 35 com o número de linhas a serem saltadas antes da impressão da mensagem(só utilizada na impressão da mensagem no verso)

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-7: Banco não encontrado no arquivo Elgin.ini.

-50: Número de série inválido.

**Observações:**

- Para utilizar esta função é necessário que exista o arquivo [CHEQUES.INI](#).

- O parâmetro *ImpressaoVerso* só poderá ser "0" em virtude de não ser possível imprimir no verso devido a restrição do equipamento, bem como o parâmetro *Linhas* devido a sua aplicabilidade estar relacionada a impressão no verso.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_ImprimeChequeMFD("009", "50,00", "Elgin", "Manaus", "10/09/06", "", "0", "0")
```

**// Exemplo em Delphi**

```
cCodBanco := '273';
cValor := '100,00';
cFavorecido := 'Elgin';
cCidade := 'Manaus';
cData := '10/09/06';
cMsg := 'Bom para dia 01/10';
```

```
cVerso := '0';  
cLinhas := '0';  
iResultado := Elgin_ImprimeChequeMFD( pchar( cCodBanco ), pchar( cValor ), pchar( cNominal ), pchar( cCidade ), pchar( cData ), pchar( cMsg ), pchar( cVerso ), pchar( cLinhas ) );
```

### 3.2.1.3 Elgin\_LeituraCheque

Permite a Leitura do CMC7.

#### **Parâmetro:**

**CodigoCMC7:** Variável String para receber o código do CMC7.

#### **Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu  
-2: Parâmetro inválido na função.  
-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.  
-5: Erro ao abrir a porta de comunicação.  
-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
-50: Número de série inválido.

#### **Exemplos**

##### **' Exemplo em Visual Basic**

```
iResultado = Elgin_LeituraCheque(sCodigoCMC7)
```

##### **// Exemplo em Delphi**

```
iResultado := Elgin_LeituraCheque(sCodigoCMC7);
```

### 3.2.1.4 Elgin\_ProgramaMoedaPlural

Programa o nome da moeda no plural para a impressão de cheques. Ex: Reais.

#### **Parâmetro:**

**MoedaPlural:** String com o Nome da moeda no plural com até 22 caracteres.

#### **Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu  
-2: Parâmetro inválido na função.  
-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.  
-5: Erro ao abrir a porta de comunicação.  
-50: Número de série inválido.

#### **Exemplos**

##### **' Exemplo em Visual Basic**

```
iResultado = Elgin_ProgramaMoedaPlural("Reais")
```

##### **// Exemplo em Delphi**

```
cMoedaPlural := 'Reais';  
iResultado := Elgin_ProgramaMoedaPlural( pchar( cMoedaPlural ) );
```

### 3.2.1.5 Elgin\_ProgramaMoedaSingular

Programa o nome da moeda no singular para a impressão de cheques. Ex. (Real)

#### **Parâmetro:**

**MoedaSingular:** string com o Nome da Moeda no singular com até 19 caracteres.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_ProgramaMoedaSingula("Real")
```

**// Exemplo em Delphi**

```
cMoedaSingular := 'Real';
```

```
iResultado := Elgin_ProgramaMoedaSingula( pchar ( cMoedaSingular ) );
```

### 3.2.1.6 Elgin\_VerificaStatusCheque

Verifica o status do cheque.

**Parâmetro:**

**StatusCheque:** Variável inteira para receber o status do cheque (veja abaixo em "**Observações**" os valores de status).

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observações:**

Os valores de status são:

1 – Impressora ok.

2 – Cheque em impressão.

3 – Cheque posicionado.

4 – Aguardando o posicionamento do cheque.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_VerificaStatusCheque(iStatusCheque)
```

**// Exemplo em Delphi**

```
iResultado := Elgin_VerificaStatusCheque( iStatusCheque );
```

### 3.2.2 Funções de Inicialização

#### 3.2.2.1 Elgin\_AlteraSimboloMoeda

Altera o símbolo da moeda programada na Impressora Fiscal.

**Parâmetro:**

**Moeda:** string contendo o símbolo da moeda. O \$ (cifrão) é inserido automaticamente.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- O cupom fiscal deve estar fechado.

- Será executada somente após uma Redução Z.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_AlteraSimboloMoeda("R")
```

**// Exemplo em Delphi**

```
cSimboloMoeda := ' R';
```

```
iResultado := Elgin_AlteraSimboloMoeda( pchar ( cSimboloMoeda ) );
```

**3.2.2.2 Elgin\_AtivaDesativaVendaUmaLinhaMFD**

Ativa ou desativa a venda de item em apenas uma linha.

**Parâmetro:**

**iFlag:** INTEGER com 1 (um) dígito, onde:

0 (zero): desativa

1 (um): ativa

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- Esta função só será executada, caso a impressora não tenha nenhum movimento no dia ou após a Redução Z.

- Não será necessário sua execução a cada início de dia.

- Para que o item possa ser impresso em uma única linha, o seu código e sua descrição (juntos) não podem ultrapassar 16 caracteres; a quantidade deve possuir até 2 dígitos (entre 1 e 99) e o valor unitário deve possuir até 3 dígitos inteiros (entre 0,01 e 999,99).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_AtivaDesativaVendaUmaLinhaMFD(1)
```

**// Exemplo em Delphi**

```
iFlag := 1;
```

```
iResultado := Elgin_AtivaDesativaVendaUmaLinhaMFD( iFlag );
```

**3.2.2.3 Elgin\_AumentaDescricaoItem**

Esta função permite aumentar a descrição do item até 200 caracteres.

**Parâmetro:**

**Descrição:** STRING com a descrição do item com até 200 caracteres.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- Esta função deve ser usada antes da função [Elgin\\_VendeItem](#), pois assim, o item será impresso com a nova descrição.

- Ela tem validade somente para a impressão de um item, voltando ao default que é de 29 caracteres na descrição do item.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_AumentaDescricaoItem("Exemplo de descrição maior")
```

**// Exemplo em Delphi**

```
cDescMaior := 'Exemplo de descrição maior';
```

```
iResultado := Elgin_AumentaDescricaoItem( pchar ( cDescMaior ) );
```

**3.2.2.4 Elgin\_EspacoEntreLinhas**

Programa o espaçamento entre as linhas impressas no cupom.

**Parâmetro:**

**Dots:** INTEIRO entre 0 e 255 indicando o espaço (dots) entre as linhas. O valor default da impressora é 0.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_EspacoEntreLinhas(002)
```

**// Exemplo em Delphi**

```
iResultado := Elgin_EspacoEntreLinhas( 002 );
```

**3.2.2.5 Elgin\_LinhasEntreCupons**

Programa o espaçamento de linhas entre os cupons.

**Parâmetro:**

**Linhas:** inteiro entre 0 e 255 indicando o número de linhas. O valor default da impressora é 8 linhas.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_LinhasEntreCupons( 5 )
```

#### // Exemplo em Delphi

```
iResultado := Elgin_LinhasEntreCupons( 5 );
```

### 3.2.2.6 Elgin\_NomeiaDepartamento

Programa departamento na impressora.

#### Parâmetros:

**Índice:** Inteiro com a posição em que o Departamento será cadastrado.

**Departamento:** String com até 15 caracteres com o nome do departamento.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

#### Observações:

- O cupom fiscal deve estar fechado
- Será executada somente após uma Redução Z.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_NomeiaDepartamento( 05, "Gasolina" )
```

#### // Exemplo em Delphi

```
cDepartamento := 'Gasolina';  
iResultado := Elgin_NomeiaDepartamento( 05, pchar ( cDepartamento ) );
```

### 3.2.2.7 Elgin\_NomeiaTotalizadorNaoSujeitoICMS

Programa Totalizador Não Sujeito ao ICMS.

#### Parâmetros:

**Índice:** Inteiro com a posição em que o totalizador será programado.

**Totalizador:** String até 19 caracteres com o nome do totalizador.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

#### Observações:

- O cupom fiscal deve estar fechado.

- Não pode haver outro totalizador cadastrado com o mesmo índice.
- Será executada somente após uma Redução Z ou antes do movimento de valor na impressora.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_NomeiaTotalizadorNaoSujeitoIcms(05, "Conta de Luz")
```

**// Exemplo em Delphi**

```
cTot := 'Conta de Luz';
```

```
iResultado := Elgin_NomeiaTotalizadorNaoSujeitoIcms( 05, pchar ( cTot ) );
```

**3.2.2.8 Elgin\_ProgramaAliquota**

Programa alíquota tributária na Impressora Fiscal.

**Parâmetros:**

**Aliquota:** string com o valor da alíquota a ser programada.

**Vinculo:** inteiro com o valor 0 (zero) para vincular a alíquota ao ICMS e 1 (um) para vincular ao ISS.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- O cupom fiscal deve estar fechado.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ProgramaAliquota("0810", 1)
```

**// Exemplo em Delphi**

```
cValor := '0810';
```

```
iResultado := Elgin_ProgramaAliquota( pchar ( cValor ), 1 );
```

**3.2.2.9 Elgin\_ProgramaArredondamento**

Programa o modo arredondamento na impressora. Este arredondamento se refere à venda de item com quantidade fracionária.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- O cupom fiscal deve estar fechado.

- Será executada somente após uma Redução Z.

- Esta operação só poderá ser executada em [MIT](#).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ProgramaArredondamento()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_ProgramaArredondamento();
```

**3.2.2.10 Elgin\_ProgramaBaudRate**

Programa o Baud Rate na impressora Fiscal.

**Parâmetro:**

**BaudRate:** String contendo o Baud Rate desejado: 9600,19200,38400,57600,115200.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- Esta função grava no arquivo [ELGIN.INI](#) o valor do novo baud rate. Para que este valor seja efetivado, o software precisa ser reiniciado ou então deve-se chamar Elgin\_AbrePortaSerial e o ECF deve ser desligado e religado.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ProgramaBaudRate("9600")
```

**// Exemplo em Delphi**

```
BaudRate := '9600';
```

```
iResultado := Elgin_ProgramaBaudRate(BaudRate);
```

**3.2.2.11 Elgin\_ProgramaHorarioVerao**

Programa e desprograma o horário de verão. Se a impressora já estiver no horário de verão o mesmo será desprogramado atrasando o relógio em 1 (uma) hora, caso contrário será adiantado 1 (uma) hora.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observação:**

- A programação do horário de verão será realizada somente após uma Redução Z. Para desprogramar, somente 1 (uma) hora e 1(um) minuto após a Redução Z e não pode ter havido movimento na impressora nesse período.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ProgramaHorarioVerao()
```

```
// Exemplo em Delphi  
iResultado := Elgin_ProgramaHorarioVerao();
```

### 3.2.2.12 Elgin\_ProgramaTruncamento

Programa o modo truncamento na impressora.

#### Parâmetro:

Não há.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu  
-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.  
-5: Erro ao abrir a porta de comunicação.  
-50: Número de série inválido.

#### Observações:

- O cupom fiscal deve estar fechado  
- Será executada somente após uma [Redução Z](#).

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_ProgramaTruncamento()
```

##### // Exemplo em Delphi

```
iResultado := Elgin_ProgramaTruncamento();
```

### 3.2.3 Funções de Cupom Fiscal

#### 3.2.3.1 Elgin\_AbreCupom

Abre o cupom fiscal na impressora.

#### Parâmetro:

**CNPJ\_CPF:** String até 26 caracteres com o CNPJ ou CPF do cliente (opcional).

#### Retornos da Função (INTEIRO):

A função retorna um inteiro indicando sucesso ou falha na operação.

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu  
-2: Parâmetro inválido na função.  
-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.  
-5: Erro ao abrir a porta de comunicação.  
-50: Número de série inválido.

#### Observações:

- O cupom fiscal deve estar fechado.  
- Caso não seja informado o CNPJ ou CPF não será impresso no cupom as informações do consumidor (CNPJ/CPF consumidor, Nome e Endereço).

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_AbreCupom("000.000.000-000")
```

##### // Exemplo em Delphi

```
iResultado := Elgin_AbreCupom('12844858000102');
```

### 3.2.3.2 Elgin\_CancelaCupom

Cancela o último cupom emitido ou cupom atual aberto.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- Não poderá ter ocorrido Redução Z e nem Leitura X após a emissão do cupom.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaCupom()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_CancelaCupom();
```

### 3.2.3.3 Elgin\_CancelaItemAnterior

Cancela o último item vendido.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- O cupom fiscal deve estar aberto.

- Ao menos um item deve ter sido vendido.

- Não deve ter sido cancelado nenhum item imediatamente anterior.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaItemAnterior()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_CancelaItemAnterior();
```

### 3.2.3.4 Elgin\_CancelaItemGenerico

Cancela qualquer item dentre os itens vendidos.

**Parâmetro:**

**NumeroItem:** String com o número do item a ser cancelado com no máximo 3 dígitos.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observações:**

- O cupom fiscal deve estar aberto.
- Ao menos um item deve ter sido vendido.
- O item não pode ter sido cancelado anteriormente.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaItemGenerico("005")
```

**// Exemplo em Delphi**

```
cNumeroItem := '005';  
iResultado := Elgin_CancelaItemGenerico( pchar ( cNumeroItem ) );
```

**3.2.3.5 Elgin\_EfetuaFormaPagamento**

Efetua o pagamento.

**Parâmetros:**

**FormaPagamento:** String com o índice ou com a descrição da forma de pagamento contendo no máximo 16 caracteres.

**ValorFormaPagamento:** String com o valor da forma de pagamento com até 13 dígitos.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 24: Forma de pagamento não programada.
- 50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_EfetuaFormaPagamento("Cartão VISA", "50,00")
```

**// Exemplo em Delphi**

```
cFormaPgto := 'Cartão VISA';  
cValorPago := '50,00';  
iResultado := Elgin_EfetuaFormaPagamento( pchar ( cFormaPgto ), pchar( cValorPago ) );
```

**3.2.3.6 Elgin\_EfetuaFormaPagamentoDescricaoForma**

Efetua o pagamento e permite que uma descrição seja impressa junto ao pagamento.

**Parâmetros:**

**FormaPagamento:** String com o índice ou com a descrição da forma de pagamento contendo no máximo 16 caracteres.

**ValorFormaPagamento:** String com o valor da forma de pagamento com até 13 dígitos.

**DescricaoFormaPagto:** String com a descrição da forma de pagamento com no máximo 80 caracteres.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 24: Forma de pagamento não programada.
- 50: Número de série inválido.

**Observações:**

- A descrição será impressa na linha após a forma de pagamento.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_EfetuaFormaPagamentoDescricaoForma("Cheque PRE", "75,00", "Vencimento em 15/02/02")
```

**// Exemplo em Delphi**

```
cFormaPgto := 'Cheque PRE';  
cValorPago := '75,00';  
cDescricao := 'Vencimento em 15/02/02';
```

```
iResultado := Elgin_EfetuaFormaPagamentoDescricaoForma( pchar ( cFormaPgto ), pchar( cValorPago ), pchar( cDescricao ) );
```

**3.2.3.7 Elgin\_EstornoFormasPagamento**

Permite estornar valores de uma forma de pagamento e inserir em outra.

**Parâmetros:**

**FormaOrigem:** String com o índice ou com a descrição da forma de pagamento de onde o valor será estornado, com até 16 caracteres.

**FormaDestino:** String com o índice ou com a descrição da forma de pagamento onde o valor será inserido, com até 16 caracteres.

**Valor:** String com o valor a ser estornado com até 13 dígitos. Não pode ser maior que o total da forma de pagamento de origem.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_EstornoFormasPagamento("Ticket", "Dinheiro", "50,00")
```

**// Exemplo em Delphi**

```
cFormaPgtoOrigem := 'Ticket';  
cFormaPgtoDestino := 'Dinheiro';  
cValorForma := '50,00';  
iResultado := Elgin_EstornoFormasPagamento( pchar ( cFormaPgtoOrigem ), pchar( cFormaPgtoDestino ), pchar( cValorForma ) );
```

### 3.2.3.8 Elgin\_FechaCupom

Fecha o cupom fiscal com a impressão da mensagem promocional.

#### Parâmetros:

**FormaPagamento:** STRING com o índice ou com a descrição da forma de pagamento contendo no máximo 16 caracteres.

**A acrescimoDesconto:** STRING que informa se haverá acréscimo ou desconto no cupom. 'A' para acréscimo e 'D' para desconto.

**TipoAcrescimoDesconto:** STRING que informa se o acréscimo ou desconto é por valor ou por percentual. '\$' para desconto por valor e '%' para percentual.

**ValorAcrescimoDesconto:** STRING com no máximo 14 dígitos para acréscimo ou desconto por valor e 4 dígitos para acréscimo ou desconto por percentual.

**ValorPago:** STRING com o valor pago com no máximo 13 dígitos.

**Mensagem:** STRING com a mensagem promocional com até 384 caracteres (8 linhas X 48 colunas).

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Observações:

- O cupom deve estar aberto.

- Pelo menos 1 (um) item deve ter sido vendido e não pode ter sido cancelado.

- A utilização dessa função elimina a obrigatoriedade de uso das funções

[Elgin\\_IniciaFechamentoCupom](#), [Elgin\\_EfetuaFormaPagamento](#) e [Elgin\\_TerminaFechamentoCupom](#) que estão implementadas internamente na função.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_FechaCupom("Dinheiro", "A", "$", "0000", "35,00", "Obrigado, volte sempre !!!")
```

##### // Exemplo em Delphi

```
cFormaPgto := 'Dinheiro';
```

```
cAcresDesc := 'A';
```

```
cTipoAcresDesc := '$';
```

```
cValorAcresDesc := '0000';
```

```
cValorPago := '35,00';
```

```
cMsgPromocional := 'Obrigado, volte sempre !!!';
```

```
iResultado := Elgin_FechaCupom( pchar( cFormaPgto ), pchar( cAcresDesc ), pchar( cTipoAcresDesc ), pchar( cValorAcresDesc ), pchar( cValorPago ), pchar( cMsgPromocional ) );
```

### 3.2.3.9 Elgin\_FechaCupomResumido

Permite fechar o cupom de forma resumida, ou seja, sem acréscimo ou desconto no cupom e com apenas uma forma de pagamento. Essa função lê o subtotal do cupom para fecha-lo.

#### Parâmetros:

**FormaPagamento:** STRING com o índice ou com a descrição da forma de pagamento contendo no máximo 16 caracteres.

**Mensagem:** STRING com a mensagem promocional com até 384 caracteres (8 linhas X 48 colunas).

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observações:**

- O cupom deve estar aberto.
- Pelo menos 1 (um) item deve ter sido vendido e não pode ter sido cancelado.
- A utilização dessa função elimina a obrigatoriedade de uso das funções [Elgin\\_IniciaFechamentoCupom](#), [Elgin\\_EfetuaFormaPagamento](#) e [Elgin\\_TerminaFechamentoCupom](#) que estão implementadas internamente na função.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_FechaCupomResumido("Ticket", "Obrigado, volte sempre !!!")
```

**// Exemplo em Delphi**

```
cFormaPgto := 'Ticket';  
cMsgPromocional := 'Obrigado, volte sempre !!!';  
iResultado := Elgin_FechaCupomResumido( pchar ( cFormaPgto ), pchar( cMsgPromocional ) );
```

**3.2.3.10 Elgin\_IniciaFechamentoCupom**

Inicia o fechamento do cupom com o uso das formas de pagamento.

**Parâmetros:**

**AcrescimoDesconto:** Indica se haverá acréscimo ou desconto no cupom. 'A' para acréscimo e 'D' para desconto.

**TipoAcrescimoDesconto:** Indica se o acréscimo ou desconto é por valor ou por percentual. '\$' para desconto por valor e '%' para percentual.

**ValorAcrescimoDesconto:** String com no máximo 14 dígitos para acréscimo ou desconto por valor e 4 dígitos para acréscimo ou desconto por percentual.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observações:**

- O cupom deve estar aberto.
- Pelo menos um item deve ter sido vendido.
- O valor do acréscimo ou descontado deve ser menor que o subtotal do cupom.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_IniciaFechamentoCupom("A", "%", "1000")
```

**// Exemplo em Delphi**

```
cAcreDesc := 'A';  
cTipoAcreDesc := '%';  
cValorAcreDesc := '1000';  
iResultado := Elgin_IniciaFechamentoCupom( pchar ( cAcreDesc ), pchar( cTipoAcreDesc ), pchar(
```

```
cValorAcreDesc ) );
```

### 3.2.3.11 Elgin\_ResetaImpressora

Reseta a impressora em caso de erro.

**Parâmetro:**

Não há

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observação:**

- Será executada somente se a impressora estiver em erro.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ResetaImpressora()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_ResetaImpressora();
```

### 3.2.3.12 Elgin\_TerminaFechamentoCupom

Termina o fechamento do cupom com mensagem promocional.

**Parâmetro:**

**Mensagem:** String com a mensagem promocional com até 384 caracteres (8 linhas X 48 colunas).

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-36: Forma de pagamento não finalizada.

-50: Número de série inválido.

**Observações:**

- A forma de pagamento deve ter sido efetuada.

- Na impressora fiscal **Elgin**, serão impressas **8 linhas de texto**.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_TerminaFechamentoCupom("Obrigado, volte sempre !!!")
```

**// Exemplo em Delphi**

```
cMsgPromocional := 'Obrigado, volte sempre !!!';
```

```
iResultado := Elgin_TerminaFechamentoCupom( pchar ( cMsgPromocional ) );
```

### 3.2.3.13 Elgin\_VendeItem

Vende item após a abertura do cupom fiscal. Essa função permite também a venda de itens com 3 casas decimais no valor unitário.

#### Parâmetros:

**Codigo:** String até 48 caracteres com o código do produto para a versão 01.00.00. Para versões superiores só é permitido 14 caracteres.

**Descricao:** String até 200 caracteres com a descrição do produto.

**Aliquota:** string com o valor ou o índice da alíquota tributária. Se for o valor deve ser informado com o tamanho de 4 caracteres ou 5 com a vírgula. Se for o índice da alíquota deve ser 2 caracteres. Ex. (18,00 para o valor ou 05 para o índice).

**TipoQuantidade:** 1 (um) caracter indicando o tipo de quantidade. I - Inteira e F - Fracionária.

**Quantidade:** String com até 4 dígitos para quantidade inteira e 7 dígitos para quantidade fracionária. Na quantidade fracionária são 3 casas decimais.

**CasasDecimais:** Inteiro indicando o número de casas decimais para o valor unitário (2 ou 3).

**ValorUnitario:** String até 8 dígitos para valor unitário.

**TipoDesconto:** 1 (um) caracter indicando a forma do desconto. '\$' desconto por valor e '%' desconto percentual.

**ValorDesconto:** String com até 8 dígitos para desconto por valor (2 casas decimais) e 4 dígitos para desconto percentual.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Observações:

- Imagine que há duas alíquotas com o valor 12% cadastradas na impressora. A primeira cadastrada na posição 01 como ICMS e a outra na posição 05 como ISS. Se você informar o valor 1200 ou 12,00 no parâmetro "alíquota" a função irá imprimir o item usando a alíquota 01 de ICMS. A função lê as alíquotas da impressora e usa o índice da primeira ocorrência. Para usar a alíquota de ISS você deverá passar o índice 05 e não o valor 1200 ou 12,00. Se você não tiver duas alíquotas com o mesmo valor cadastradas na impressora use sempre o valor no parâmetro alíquota.

- Para vender itens com alíquotas de:

- **Substituição tributária ICMS e ISSQN**, utilize os índices -2 e -11, respectivamente.

- **Isenção ICMS e ISSQN**, utilize os índices -3 e -12, respectivamente.

- **Não tributadas ICMS e ISSQN**, utilize os índices -4 e -13, respectivamente.

- O cupom fiscal deve estar aberto.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_VendeItem("121", "Teclado", "1200", "I", "10", 2, "0,25", "%", "0000")
```

##### // Exemplo em Delphi

```
cCodigo := '121';  
cDescricao := 'Teclado';  
cAliquota := '1200';  
cTipoQtde := 'I';  
cQtde := '10';  
iCasasDecimais := 2;  
cValor := '0,25';  
cTipoDesconto := '%';  
cValorDesc := '0000';
```

iResultado := Elgin\_VendeItem( pchar( cCodigo ), pchar( cDescricao ), pchar( cAliquota ), pchar( cTipoQtde ), pchar( cQtde ), iCasasDecimais, pchar( cValor ), pchar( cTipoDesconto ), pchar( cValorDesc ) );

### Tópico relacionado:

[Situções Tributárias](#)

#### 3.2.3.14 Elgin\_VendeItemDepartamento

Essa função permite a venda de itens com entrada de departamento, desconto e unidade de medida.

#### Parâmetros:

**Codigo:** String até 48 caracteres com o código do produto para a versão 01.00.00. Para versões superiores só é permitido 14 caracteres.

**Descricao:** String até 200 caracteres com a descrição do produto.

**Aliquota:** String com o valor ou o índice da alíquota tributária. Se for o valor deve ser informado com o tamanho de 4 caracteres ou 5 com a vírgula. Se for o índice da alíquota deve ser 2 caracteres. Ex. (18,00 para o valor ou 05 para o índice)

**ValorUnitario:** String com até 8 dígitos para o valor (três casas decimais).

**Quantidade:** String com até 7 dígitos para a quantidade. Na venda com departamento a quantidade é fracionária e são 3 casas decimais.

**Acrescimo:** String com o acréscimo por valor com até 5 dígitos (3 casas decimais).

**Desconto:** String com o desconto por valor com até 5 dígitos (3 casas decimais).

**IndiceDepartamento:** String com o índice do departamento com 2 dígitos.

**UnidadeMedida:** String com no máximo 2 caracteres para a unidade de medida. Caso não seja passado nenhum caracter a unidade não é impressa.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu

-2: Parâmetro inválido na função.

-3: Alíquota não programada.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Observações:

- Obedece as mesmas situações descrita na função [Elgin\\_VendeItem](#).

- Caso não exista *NENHUM* departamento cadastrado, pode ser passado o código "00" que será cadastrado.

- Para vender itens com alíquotas de:

- **Substituição tributária ICMS e ISSQN**, utilize os índices -2 e -11, respectivamente.

- **Isenção ICMS e ISSQN**, utilize os índices -3 e -12, respectivamente.

- **Não tributadas ICMS e ISSQN**, utilize os índices -4 e -13, respectivamente.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_VendeItemDepartamento("123", "Caneta", "1200", "0,25", "10", "0", "0", "03", "UN")
```

##### // Exemplo em Delphi

```
cCodigo := '123';
cDescricao := 'Caneta';
cAliquota := '1200';
cValor := '0,250';
cQtde := '10,000';
cAcresc := '0';
cDesc := '0';
cIndice := '03';
```

cUnidade := 'UN';

iResultado := Elgin\_VendeItemDepartamento( pchar( cCodigo ), pchar( cDescricao ), pchar( cAliquota ), pchar( cValor ), pchar( cQtde ), pchar( cAcresc ), pchar( cDesc ), pchar( cIndice ), pchar( cUnidade ) );

### 3.2.4 Funções de Relatórios

#### 3.2.4.1 Elgin\_AbreRelatorioGerencial

Abre o primeiro Relatório Gerencial localizado, imprimindo seu cabeçalho e nome.

##### **Retornos da Função (INTEIRO):**

A função retorna um inteiro indicando sucesso ou falha na operação.

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

##### **Observações:**

- O cupom fiscal deve estar fechado.

##### **Exemplos**

###### **' Exemplo em Visual Basic**

```
iResultado = Elgin_AbreRelatorioGerencial
```

###### **// Exemplo em Delphi**

```
iResultado := Elgin_AbreRelatorioGerencial;
```

#### 3.2.4.2 Elgin\_FechaRelatorioGerencial

Encerra o relatório gerencial.

##### **Parâmetro:**

Não há.

##### **Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

##### **Observação:**

- O relatório gerencial deve estar aberto.

##### **Exemplos**

###### **' Exemplo em Visual Basic**

```
iResultado = Elgin_FechaRelatorioGerencial()
```

###### **// Exemplo em Delphi**

```
iResultado := Elgin_FechaRelatorioGerencial();
```

#### 3.2.4.3 Elgin\_LeituraMemoriaFiscalData

Imprime a leitura da memória fiscal da impressora por intervalo de datas.

##### **Parâmetros:**

**DataInicial:** STRING para receber a Data inicial no formato dd/mm/aa ou dd/mm/aaaa.

**DataFinal:** STRING para receber a Data final no formato dd/mm/aa ou dd/mm/aaaa.

**FlagLeitura:** STRING com o valor "s" para leitura simplificada e "c" para leitura completa.

#### **Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### **Observação:**

- O cupom fiscal deve estar fechado.

#### **Exemplos**

##### **' Exemplo em Visual Basic**

```
iResultado = Elgin_LeituraMemoriaFiscalData("03/03/07", "15/03/07", "c")
```

##### **// Exemplo em Delphi**

```
cDataInicial := '3/03/07';
```

```
cDataFinal := '15/03/07';
```

```
cTipo := 'c';
```

```
iResultado := Elgin_LeituraMemoriaFiscalData( pchar( cDataInicial ), pchar( cDataFinal ), pchar( cTipo ) );
```

#### **3.2.4.4 Elgin\_LeituraMemoriaFiscalReducao**

Emite a leitura da memória fiscal da impressora por intervalo de reduções.

#### **Parâmetros:**

**ReducaoInicial:** STRING com o Número da reducao inicial com até 4 dígitos.

**ReducaoFinal:** STRING com o Número da reducao final com até 4 dígitos.

**FlagLeitura:** STRING com o valor "s" para leitura simplificada e "c" para leitura completa.

#### **Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### **Observação:**

- O cupom fiscal deve estar fechado.

#### **Exemplos**

##### **' Exemplo em Visual Basic**

```
iResultado = Elgin_LeituraMemoriaFiscalReducao("0011", "0013", "c")
```

##### **// Exemplo em Delphi**

```
cReducaoInicial := '0011';
```

```
cReducaoFinal := '0013';
```

```
cTipo := 'c';
```

```
iResultado := Elgin_LeituraMemoriaFiscalReducao( pchar( cReducaoInicial ), pchar( cReducaoFinal ), pchar( cTipo ) );
```

### 3.2.4.5 Elgin\_LeituraMemoriaFiscalSerialData

Recebe os dados da memória fiscal por intervalo de datas pela serial e grava em arquivo texto. Será gerado um arquivo chamado Impressão.txt no caminho especificado no arquivo [Elgin.ini](#) (Parâmetro Path).

**Parâmetros:**

**DataInicial:** STRING para receber a Data inicial no formato dd/mm/aa ou dd/mm/aaaa.

**DataFinal:** STRING para receber a Data final no formato dd/mm/aa ou dd/mm/aaaa.

**FlagLeitura:** STRING com o valor "s" para leitura simplificada e "c" para leitura completa.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observação:**

- O cupom fiscal deve estar fechado.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_LeituraMemoriaFiscalSerialData("01/03/07", "15/03/07", "c")
```

**// Exemplo em Delphi**

```
cDataInicial := '01/03/07';
```

```
cDataFinal := '15/03/07';
```

```
cTipo := 'c';
```

```
iResultado := Elgin_LeituraMemoriaFiscalSerialData( pchar( cDataInicial ), pchar( cDataFinal ), pchar( cTipo ) );
```

### 3.2.4.6 Elgin\_LeituraMemoriaFiscalSerialReducao

Recebe os dados da leitura da memória fiscal por intervalo de reduções pela serial e grava em arquivo texto. Será gerado um arquivo chamado Impressão.txt no caminho especificado no arquivo [Elgin.ini](#) (Parâmetro Path).

**Parâmetros:**

**ReducaoInicial:** STRING com o Número da reducao inicial com até 4 dígitos.

**ReducaoFinal:** STRING com o Número da reducao final com até 4 dígitos.

**FlagLeitura:** STRING com o valor "s" para leitura simplificada e "c" para leitura completa.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observação:**

- O cupom fiscal deve estar fechado.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_LeituraMemoriaFiscalSerialReducao("0010", "0015", "c")
```

**// Exemplo em Delphi**

```
cReducaoInicial := '0010';
```

```
cReducaoFinal := '0015';
```

```
cTipo := 'c';
```

```
iResultado := Elgin_LeituraMemoriaFiscalSerialReducao( pchar ( cReducaoInicial ), pchar( cReducaoFinal ),  
pchar( cTipo ) );
```

**3.2.4.7 Elgin\_LeituraX**

Fornece a Leitura "X" impressa. Este comando não zera os totalizadores e contadores diários e não atualiza a memória fiscal.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observação:**

- Um relatório de leitura X pode ser impresso a qualquer momento, exceto quando a impressora encontra-se com um cupom fiscal ou comprovante não fiscal aberto;

- Se a impressora encontra-se no estado livre, a impressão da leitura X coloca a impressora em período de venda, mesmo que nenhuma venda tenha ainda ocorrido. É a chamada "abertura do dia".

- O cupom fiscal deve estar fechado.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_LeituraX()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_LeituraX();
```

**3.2.4.8 Elgin\_LeNomeRelatorioGerencial**

Retorna os relatórios gerenciais cadastrados na impressora.

**Parâmetro:**

**codigo:** STRING com valor de '00' a '19' indicando o índice do gerencial que se quer ler.

**NomeRelatorio:** STRING preparada com 30 bytes para receber o nome do relatório.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observação:**

- Informações sobre o cadastro de Relatórios Gerenciais podem ser encontradas em: [Tabelas do ECF](#).

### Exemplos

#### Exemplo em Visual Basic

```
NomeRelatorio = Space(30)
```

```
iResultado = Elgin_LeNomeRelatorioGerencial("2", NomeRelatorio)
```

#### // Exemplo em Delphi

```
NomeRelatorio := StringOfChar(' ', 30);
```

```
iResultado := Elgin_LeNomeRelatorioGerencial('2', NomeRelatorio);
```

### 3.2.4.9 Elgin\_ReducacaoZ

Este comando imprime o relatório da Redução "Z". Com isso, zera os totalizadores e contadores diários e grava na memória fiscal os dados relativos à venda bruta do dia.

Com este comando, finaliza-se o período de venda no dia correspondente. Após a Redução "Z" e antes de abrir um novo período de venda é permitida a programação e cadastros de parametrização da impressora.

#### Parâmetros:

**Data:** STRING com a Data atual da impressora no formato ddmmaa ou dd/mm/aa, dd/mm/aaaa ou dd/mm/aa.

**Hora:** STRING com a Hora a ser alterada no formato hhmmss ou hh:mm:ss.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Observações:

- **A impressora Elgin não permite que seja alterada a Data, somente a hora pode ser mudada.**

- Somente será aceito um ajuste de +/- 5 minutos. Se os valores estiverem fora dessa faixa serão limitados a 5 minutos.

- O horário não pode ser ajustado para um valor inferior ao último cupom emitido.

- O cupom fiscal deve estar fechado.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_ReducacaoZ("13/11/2006", "15:00:00")
```

```
ou
```

```
iResultado = Elgin_ReducacaoZ("", "")
```

#### // Exemplo em Delphi

```
cData := '13/11/2006';
```

```
cHora := '15:00:00';
```

```
iResultado := Elgin_ReducacaoZ( pchar ( cData ), pchar( cHora ) );
```

```
ou
```

```
iResultado := Elgin_ReducacaoZ( pchar ( ' ' ), pchar( ' ' ) );
```

### 3.2.4.10 Elgin\_RelatorioGerencial

Imprime o relatório gerencial.

#### Parâmetro:

**Texto:** STRING com o texto a ser impresso no relatório com até 492 caracteres.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observações:**

- O cupom fiscal deve estar fechado.
- Se qualquer função diferente da Elgin\_RelatorioGerencial for chamada com o relatório aberto, efetuará seu fechamento automaticamente.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_RelatorioGerencial("Texto a ser impresso")
```

**// Exemplo em Delphi**

```
ElginTexto := 'Relatório Gerencial Elgin.dll';  
iResultado := Elgin_RelatorioGerencial( pchar ( ElginTexto ) );
```

**3.2.5 Funções de Operações não Fiscais****3.2.5.1 Elgin\_AbreComprovanteNaoFiscalVinculado**

Abre o comprovante não fiscal vinculado.

**Parâmetros:**

**FormaPagamento:** STRING com a forma de pagamento com até 16 caracteres.

**Valor:** STRING com o valor pago na forma de pagamento com até 13 dígitos (2 casas decimais).

**NumeroCupom:** STRING com o número do cupom a que se refere o comprovante com até 6 dígitos.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observações:**

- A forma de pagamento utilizada não pode ser dinheiro.
- A forma de pagamento deve ter sido utilizada no cupom fiscal.
- Só pode ser emitido um comprovante não fiscal por forma de pagamento.
- Caso o comprovante emitido seja referente ao último cupom fiscal, é opcional informar o número do cupom.
- O parâmetro "Numero do Cupom" torna-se obrigatório se o comprovante emitido não for referente ao último cupom fiscal emitido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_AbreComprovanteNaoFiscalVinculado("Cartao VISA", "10,00", "")
```

**// Exemplo em Delphi**

```
cFormaPgto := 'Cartao VISA';  
cValor := '100,00';  
iResultado := Elgin_AbreComprovanteNaoFiscalVinculado( pchar ( cFormaPgto ), pchar(cValor), pchar( " ) );
```

### 3.2.5.2 Elgin\_CancelaAcredicoNaoFiscalMFD

Cancela o acréscimo ou desconto em um item não fiscal na impressora fiscal MFD.

**Parâmetros:**

**NumeroItem:** STRING contendo o número do item não fiscal, com até 3 caracteres.

**AcredicoDesconto:** STRING com "A" para o cancelamento do Acréscimo ou "D" para o cancelamento do desconto.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observação:**

- O cupom não fiscal não pode estar subtotalizado.

**Exemplos**

**Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaAcredicoNaoFiscalMFD("005", "A")
```

**Exemplo em Delphi**

```
cNumeroItem := '005';
```

```
cAcresDesc := 'A';
```

```
iResultado := Elgin_CancelaAcredicoNaoFiscalMFD( pchar ( cNumeroItem ), pchar( cAcresDesc ) );
```

### 3.2.5.3 Elgin\_CancelaItemNaoFiscalMFD

Cancela um item não fiscal na impressora fiscal MFD.

**Parâmetro:**

**NumeroItem:** STRING contendo o número do item não fiscal a ser cancelado, com até 3 caracteres.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos**

**Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaItemNaoFiscalMFD("1")
```

**Exemplo em Delphi**

```
iResultado := Elgin_CancelaItemNaoFiscalMFD('1');
```

### 3.2.5.4 Elgin\_FechaComprovanteNaoFiscalVinculado

Encerrar o comprovante não fiscal vinculado.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observação:**

- O comprovante não fiscal vinculado deve ter sido aberto.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_FechaComprovanteNaoFiscalVinculado()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_FechaComprovanteNaoFiscalVinculado();
```

**3.2.5.5 Elgin\_RecebimentoNaoFiscal**

Imprime o comprovante não fiscal não vinculado.

**Parâmetros:**

**ÍndiceTotalizador:** STRING com o índice do totalizador para recebimento parcial com até 2 dígitos.

**ValorRecebimento:** STRING com o Valor do recebimento (suprimento ou sangria) com até 13 dígitos (duas casas decimais).

**FormaPagamento:** STRING com a Forma de pagamento com até 16 caracteres.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- O cupom fiscal deve estar fechado.

- O totalizador deve estar cadastrado.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_RecebimentoNaoFiscal("05", "30,00", "Dinheiro")
```

**// Exemplo em Delphi**

```
cIndiceTot := '05';
```

```
cValor := '30,00';
```

```
cFormaPgto := 'Dinheiro';
```

```
iResultado := Elgin_RecebimentoNaoFiscal( pchar( cIndiceTot ), pchar( cValor ), pchar( cFormaPgto ) );
```

**3.2.5.6 Elgin\_Sangria**

Faz uma sangria na impressora (retirada de dinheiro).

**Parâmetro:**

**Valor:** STRING com o Valor da sangria com até 13 dígitos (2 casas decimais).

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observação:**

- O cupom fiscal deve estar fechado.
- Deve existir um comprovante não fiscal de sangria cadastrado na impressora. Esse comprovante deve ser de retirada, ou seja, o TipoNaoFiscal do comprovante de retirada é sempre igual a N.
- O índice do comprovante de sangria cadastrado na impressora deve ser adicionado no parâmetro SANGRIA do arquivo [Elgin.ini](#).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_Sangria("25,00")
```

**// Exemplo em Delphi**

```
cValor := '25,00';  
iResultado := Elgin_Sangria( pchar ( cValor ) );
```

**3.2.5.7 Elgin\_Suprimento**

Faz um suprimento na impressora (entrada de dinheiro).

**Parâmetros:**

**Valor:** STRING com o Valor do suprimento com até 13 dígitos (2 casas decimais).

**FormaPagamento:** STRING com a Forma de pagamento com até 16 caracteres. Se não for informada, o suprimento será feito em Dinheiro.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observação:**

- Deve existir um comprovante não fiscal de suprimento cadastrado na impressora esse comprovante deve ser de entrada, ou seja, o TipoNaoFiscal do comprovante de entrada é sempre igual a Y.
- O índice do comprovante de suprimento cadastrado na impressora deve ser adicionado no parâmetro SUPRIMENTO do arquivo [Elgin.ini](#).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_Suprimento("50,00", "Dinheiro")
```

**// Exemplo em Delphi**

```
cValor := '50,00';  
cFormaPgto := 'Dinheiro';  
iResultado := Elgin_Suprimento( pchar ( cValor ), pchar( cFormaPgto ) );
```

**3.2.5.8 Elgin\_UsaComprovanteNaoFiscalVinculado**

Imprime o comprovante não fiscal vinculado.

**Parâmetro:**

**Texto:** STRING com o Texto a ser impresso no comprovante não fiscal vinculado com até 492

caracteres.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_UsaComprovanteNaoFiscalVinculado("Digite o texto a ser impresso aqui !!!")
```

**// Exemplo em Delphi**

```
cTexto := 'Obrigado!!!';
```

```
iResultado := Elgin_UsaComprovanteNaoFiscalVinculado( pchar ( cTexto ) );
```

**3.2.6 Funções de Informações da impressora****3.2.6.1 Elgin\_Acrescimos**

Retorna a valor acumulado dos acréscimos efetuados nos cupons.

**Parâmetro:**

*ValorAcrescimo*: STRING com 14 posições para receber o valor dos acréscimos.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
ValorAcrescimo = Space(14)
```

```
iResultado = Elgin_Acrescimos(ValorAcrescimo)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 14 do ValorAcrescimo := ValorAcrescimo + ' ';
```

```
iResultado := Elgin_Acrescimos( ValorAcrescimo );
```

**3.2.6.2 Elgin\_Cancelamentos**

Retorna o valor acumulado dos itens e dos cupons cancelados

**Parâmetro:**

*Cancelamentos*: STRING com 14 posições para receber o valor dos cancelamentos com 2 casas decimais.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
Cancelamentos = Space(14)  
iResultado = Elgin_Cancelamentos(Cancelamentos)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 14 do Cancelamentos := Cancelamentos + '  
iResultado := Elgin_Cancelamentos( Cancelamentos );
```

### 3.2.6.3 Elgin\_CGC\_IE

Retorna o CGC e a Inscrição Estadual do cliente/proprietário que está cadastrado na impressora.

### Parâmetros:

*CGC*: STRING com 18 posições para receber o CGC.

*IE*: STRING com 15 posições para receber a Inscrição Estadual.

### Possíveis retornos da Função (INTEIRO):

0: Erro de comunicação.

1: OK.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
strRetorno01 := Space (18);  
strRetorno02 := Space (15);  
iResultado := Elgin_CGC_IE (strRetorno01, strRetorno02);
```

#### // Exemplo em Delphi

```
strRetorno01 := StringOfChar (' ', 18);  
strRetorno02 := StringOfChar (' ', 15);  
iResultado := Elgin_CGC_IE (strRetorno01, strRetorno02);
```

### 3.2.6.4 Elgin\_ClicheProprietario

Retorna o clichê do proprietário cadastrado na impressora.

### Parâmetro:

*ClicheProprietario*: STRING com 186 posições para receber clichê cadastrado.

### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
ClicheProprietario = Space(186)  
iResultado = Elgin_ClicheProprietario(ClicheProprietario)
```

```
// Exemplo em Delphi
for iConta := 1 to 186 do ClicheProprietario := ClicheProprietario + ' ';
iResultado := Elgin_ClicheProprietario( ClicheProprietario );
```

### 3.2.6.5 Elgin\_CNPJ\_IE

Retorna o CGC e a Inscrição Estadual do cliente/proprietário que está cadastrado na impressora.

#### Parâmetros:

*CGC*: STRING com 18 posições para receber o CGC.

*IE*: STRING com 15 posições para receber a Inscrição Estadual.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

CGC = Space(18)

IE = Space(15)

iResultado = Elgin\_CNPJ\_IE(CGC, IE)

##### // Exemplo em Delphi

```
for iConta := 1 to 18 do CGC := CGC + ' ';
```

```
for iConta := 1 to 15 do IE := IE + ' ';
```

```
iResultado := Elgin_CNPJ_IE( CGC, IE );
```

### 3.2.6.6 Elgin\_ContadoresTotalizadoresNaoFiscais

Retorna o número de vezes em que os totalizadores não sujeitos ao ICMS foram usados.

#### Parâmetro:

*Contadores*: STRING com 44 posições para receber os contadores dos totalizadores.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

Contadores = Space(44)

iResultado = Elgin\_ContadoresTotalizadoresNaoFiscais(Contadores)

##### // Exemplo em Delphi

```
for iConta := 1 to 44 do Contadores := Contadores + ' ';
```

```
iResultado := Elgin_ContadoresTotalizadoresNaoFiscais( Contadores );
```

### 3.2.6.7 Elgin\_DadosUltimaReducao

Retorna os dados da impressora no momento da última Redução Z.

**Parâmetro:**

*DadosReducao*: STRING com o tamanho de 631 posições para receber os dados da última redução.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observações:**

- São retornados os valores das seguintes informações separados por vírgulas:

Modo de Redução Z	2 bytes (se 00 redução por comando, se 01 redução automática)
Grande Total	18 bytes
Cancelamentos	14 bytes
Descontos	14 bytes
Tributos	64 bytes
Totalizadores Parciais Tributados	266 bytes
Sangria	14 bytes
Suprimentos	14 bytes
Totalizadores não sujeitos ao ICMS	126 bytes
Contadores dos TP's não sujeitos ao ICMS	36 bytes
Contador de ordem de operação	6 bytes
Contador de operações não sujeitas ao ICMS	6 bytes
Número de Alíquotas cadastradas	2 bytes
Data do movimento	6 bytes
Acréscimos	14 bytes
Acréscimo financeiro	14 bytes

**Exemplos**

**' Exemplo em Visual Basic**

```
DadosReducao = Space(631)
```

```
iResultado = Elgin_DadosUltimaReducao(DadosReducao)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 631 do DadosReducao := DadosReducao + ' ';
```

```
iResultado := Elgin_DadosUltimaReducao( DadosReducao );
```

### 3.2.6.8 Elgin\_DataHoraImpressora

Retorna a data e a hora atual da impressora.

**Parâmetros:**

*Data*: STRING com 6 posições para receber a data atual da impressora no formato ddmmaa.

*Hora*: STRING com 6 posições para receber a hora atual da impressora no formato hhmss.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
Data = Space(6)
```

```
Hora = Space(6)
```

```
iResultado = Elgin_DataHoraImpressora(Data, Hora)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 6 do Data := Data + ' ';
```

```
for iConta := 1 to 6 do Hora := Hora + ' ';
```

```
iResultado := Elgin_DataHoraImpressora( Data, Hora );
```

**3.2.6.9 Elgin\_DataHoraReducao**

Retorna a data da última Redução Z.

**Parâmetros:**

*DataReducao*: STRING com 6 posições para receber a data da última redução no formato ddmmaa.

*HoraReducao*: STRING com 6 posições para receber a hora da última redução no formato hhmmss.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
DataReducao = Space(6)
```

```
HoraReducao = Space(6)
```

```
iResultado = Elgin_DataHoraReducao(DataReducao, HoraReducao)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 6 do DataReducao := DataReducao + ' ';
```

```
for iConta := 1 to 6 do HoraReducao := HoraReducao + ' ';
```

```
iResultado := Elgin_DataHoraReducao( DataReducao, HoraReducao );
```

**3.2.6.10 Elgin\_DataHoraSoftwareBasico**

Retorna a data e a hora de gravação do Software Básico.

**Parâmetros:**

*DataSW*: STRING com 6 posições para receber a data de gravação do software básico no formato ddmmaa.

*HoraSW*: STRING com 6 posições para receber a hora de gravação do software básico no formato hhmmss.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
Data = Space(6)
```

```
Hora = Space(6)
```

```
iResultado = Elgin_DataHoraSoftwareBasico(DataSW, HoraSW)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 6 do Data := Data + ' ';
```

```
for iConta := 1 to 6 do Hora := Hora + ' ';
```

```
iResultado := Elgin_DataHoraSoftwareBasico( DataSW, HoraSW );
```

**3.2.6.11 Elgin\_DataMovimento**

Retorna a data do último movimento.

**Parâmetro:**

*DataMovimento*: STRING com 6 posições para receber a data do movimento no formato ddmmaa.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
DataMovimento = Space(6)
```

```
iResultado = Elgin_DataMovimento(DataMovimento)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 6 do DataMovimento := DataMovimento + ' ';
```

```
iResultado := Elgin_DataMovimento( DataMovimento );
```

**3.2.6.12 Elgin\_Descontos**

Retorna a valor acumulado dos descontos.

**Parâmetro:**

*Descontos*: STRING com 14 posições para receber o valor dos descontos com 2 casas decimais.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
Descontos = Space(14)
iResultado = Elgin_Descontos(Descontos)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 14 do Descontos := Descontos + ' ';
iResultado := Elgin_Descontos( Descontos );
```

**3.2.6.13 Elgin\_FlagsFiscais**

Retorna um número referente ao flag fiscal da impressora. Veja discriminação abaixo.

**Parâmetro:**

*FlagFiscal*: INTEIRO para receber um número representando o flag fiscal da impressora. Veja discriminação abaixo.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observações:**

- Discriminação dos valores para o flag fiscal:

Descrição	Valor
Cupom fiscal aberto	1
Fechamento de formas de pagamento iniciado	2
Horário de verão selecionado	4
Já houve redução Z no dia	8
Não utilizado	16
Permite cancelar cupom fiscal	32
Não utilizado	64
Memória fiscal sem espaço	128

- Os valores podem vir somados o que indica mais de um estado.

**Exemplos**

FlagFiscal = 37 (Cupom fiscal aberto, Horário de verão selecionado, Permite cancelar o cupom fiscal).

**' Exemplo em Visual Basic**

```
iResultado = Elgin_FlagsFiscais(FlagFiscal)
```

**// Exemplo em Delphi**

```
iResultado := Elgin_FlagsFiscais( FlagFiscal );
```

**3.2.6.14 Elgin\_FlagsFiscaisStr**

Retorna um número referente ao flag fiscal da impressora.

**Parâmetro:**

*FlagFiscal*: STRING para receber um número representando o flag fiscal da impressora. Veja discriminação abaixo.

**Possíveis retornos da Função (INTEIRO):**

0: Erro de comunicação.

1: OK.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observações:**

- Discriminação dos valores para o flag fiscal:

Descrição	Valor
Cupom fiscal aberto	1
Fechamento de formas de pagamento iniciado	2
Horário de verão selecionado	4
Já houve redução Z no dia	8
Permite cancelar cupom fiscal	32
Memória fiscal sem espaço	128

**Exemplos****'Exemplo em Visual Basic**

```
strRetorno01 = Space(3)
```

```
iResultado = Elgin_FlagsFiscaisStr(strRetorno01)
```

**// Exemplo em Delphi**

```
strRetorno01 := StringOfChar(' ',3);
```

```
iResultado := Elgin_FlagsFiscaisStr(strRetorno01);
```

**3.2.6.15 Elgin\_GrandeTotal**

Retorna o valor do Grande Total da impressora.

**Parâmetro:**

*GrandeTotal*: STRING com 18 posições para receber o valor do grande total com 2 casas decimais.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
GrandeTotal = Space(18)
```

```
iResultado = Elgin_GrandeTotal(GrandeTotal)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 18 do GrandeTotal := GrandeTotal + ' ';
```

```
iResultado := Elgin_GrandeTotal( GrandeTotal );
```

### 3.2.6.16 Elgin\_LeIndicadores

Retorna em Indicadores a composição dos seguintes flags:

Valor	Situação	Descrição
1	FLAG_INTERVENCAO_TECNICA	Equipamento em Intervenção Técnica.
2	FLAG_SEM_MFD	MFD não encontrada.
4	FLAG_RAM_NOK	RAM não está consistente.
8	FLAG_RELOGIO_NOK	Relógio inconsistente.
16	FLAG_SEM_MF	Memória fiscal não encontrada.
32	FLAG_DIA_FECHADO	Dia fiscal já encerrado.
64	FLAG_DIA_ABERTO	Dia aberto.
128	FLAG_Z_PENDENTE	Redução Z pendente.
256	FLAG_SEM_PAPEL	Sem papel na estação de cupom fiscal.
512	FLAG_MECANISMO_NOK	Mecanismo impressor não configurado.
1024	FLAG_DOCUMENTO_ABERTO	Documento em emissão não foi encerrado.
2048	FLAG_INSCRICOES_OK	Inscrições carregadas.
4096	FLAG_CLICHE_OK	Clichê carregado.
8192	FLAG_EM_LINHA	Equipamento está em linha.
16384	FLAG_MFD_ESGOTADA	MFD esgotada.

Vários indicadores podem estar ativos ou não simultaneamente.

#### Parâmetro:

*Indicador:* Variável INTEIRA para receber algum indicador.

#### Possíveis retornos da Função (INTEIRO):

0: Erro de comunicação.

1: OK.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### 'Exemplo em Visual Basic

```
iResultado = Elgin_LeIndicadores(iRetorno01)
```

##### //Exemplo em Delphi

```
iResultado := Elgin_LeIndicadores(iRetorno01);
```

### 3.2.6.17 Elgin\_LeituraXSerial

Recebe os dados da Leitura X pela serial e grava em arquivo texto no caminho especificado no arquivo [Elgin.ini](#) no parâmetro "PATH".

#### Parâmetro:

Não há.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

**Observação:**

- O cupom fiscal deve estar fechado.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_LeituraXSerial()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_LeituraXSerial();
```

**3.2.6.18 Elgin\_MinutosImprimindo**

Verifica se a impressora está em modo normal ou em intervenção técnica.

**Parâmetro:**

*ModoOperacao*: STRING com 1 posição para receber o modo de operação da impressora. Onde:

1 - Modo normal

0 - Intervenção técnica.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
ModoOperacao = " "
```

```
iResultado = Elgin_VerificaModoOperacao(ModoOperacao)
```

**// Exemplo em Delphi**

```
ModoOperacao := ' ';
```

```
iResultado := Elgin_VerificaModoOperacao( ModoOperacao );
```

**3.2.6.19 Elgin\_MinutosLigada**

Retorna o tempo em minutos que a impressora está ligada.

**Parâmetro:**

*MinutosLigada*: STRING com 4 posições para receber os minutos em que a impressora está ligada.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
MinutosLigada = Space(4)  
iResultado = Elgin_MinutosLigada(MinutosLigada)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 4 do MinutosLigada := MinutosLigada + ' ';  
iResultado := Elgin_MinutosLigada( MinutosLigada );
```

**3.2.6.20 Elgin\_NumeroCaixa**

Retorna o número do caixa cadastrado na impressora.

**Parâmetro:**

*NumeroCaixa*: STRING com 4 posições para receber o número do caixa cadastrado na impressora.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
NumeroCaixa = Space(4)  
iResultado = Elgin_NumeroCaixa(NumeroCaixa)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 4 do NumeroCaixa := NumeroCaixa + ' ';  
iResultado := Elgin_NumeroCaixa( NumeroCaixa );
```

**3.2.6.21 Elgin\_NumeroCupom**

Retorna o número do cupom.

**Parâmetro:**

*NumeroCupom*: STRING com 6 posições para receber o número do último cupom.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
NumeroCupom = Space(6)  
iResultado = Elgin_NumeroCupom(NumeroCupom)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 6 do NumeroCupom := NumeroCupom + ' ';  
iResultado := Elgin_NumeroCupom( NumeroCupom );
```

### 3.2.6.22 Elgin\_NumeroCuponsCancelados

Retorna o número de cupons cancelados.

**Parâmetro:**

*CuponsCancelados*: STRING com o tamanho de 4 bytes para receber o número de cupons cancelados.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo Elgin.ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
Cupons = Space(4)
```

```
iResultado = Elgin_NumeroCuponsCancelados(Cupons)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 4 do Cupons := Cupons + ' ';
```

```
iResultado := Elgin_NumeroCuponsCancelados( Cupons );
```

### 3.2.6.23 Elgin\_NumeroIntervencoes

Retorna o número de intervenções técnicas realizadas na impressora.

**Parâmetro:**

*Intervencoes*: STRING com 4 posições para receber o número de intervenções.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
Intervencoes = Space(4)
```

```
iResultado = Elgin_NumeroIntervencoes(Intervencoes)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 4 do Intervencoes := Intervencoes + ' ';
```

```
iResultado := Elgin_NumeroIntervencoes( Intervencoes );
```

### 3.2.6.24 Elgin\_NumeroLoja

Retorna o número da loja cadastrado na impressora.

**Parâmetro:**

*NumeroLoja*: STRING com 4 posições para receber o número da loja cadastrado na impressora.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
NumeroLoja = Space(4)  
iResultado = Elgin_NumeroLoja(NumeroLoja)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 4 do NumeroLoja := NumeroLoja + ' ';  
iResultado := Elgin_NumeroLoja( NumeroLoja );
```

### 3.2.6.25 Elgin\_NumeroOperacoesNaoFiscais

Retorna o número de operações não fiscais executadas na impressora.

#### Parâmetro:

*Operacoes*: STRING com 6 posições para receber o número de operações.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
Operacoes = Space(6)  
iResultado = Elgin_NumeroOperacoesNaoFiscais(Operacoes)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 6 do Operacoes := Operacoes + ' ';  
iResultado := Elgin_NumeroOperacoesNaoFiscais( Operacoes );
```

### 3.2.6.26 Elgin\_NumeroReducoes

Retorna o número de reduções Z realizadas na impressora.

#### Parâmetro:

*Reducoes*: STRING com 4 posições para receber o número de Reduções Z.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
Reducoes = Space(4)  
iResultado = Elgin_NumeroReducoes(Reducoes)
```

```
// Exemplo em Delphi
for iConta := 1 to 4 do Reducoes := Reducoes + ' ';
iResultado := Elgin_NumeroReducoes( Reducoes );
```

### 3.2.6.27 Elgin\_NumeroSerie

Retorna o número de série da impressora MFD.

#### Parâmetro:

*NumeroSerie*: STRING com o tamanho de 20 posições para receber a informação.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
sNumeroSerie = space(20)
```

```
iResultado = Elgin_NumeroSerie(sNumeroSerie)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 20 do sNumeroSerie := sNumeroSerie + ' ';
```

```
iResultado := Elgin_NumeroSerie(sNumeroSerie);
```

### 3.2.6.28 Elgin\_NumeroSubstituicoesProprietario

Retorna o número de substituições de proprietário.

#### Parâmetro:

*Substituicoes*: STRING com 4 posições para receber o número de substituições.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
Substituicoes = Space(4)
```

```
iResultado = Elgin_NumeroSubstituicoesProprietario(Substituicoes)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 4 do Substituicoes := Substituicoes + ' ';
```

```
iResultado := Elgin_NumeroSubstituicoesProprietario( Substituicoes );
```

### 3.2.6.29 Elgin\_RetornoAliquotas

Retorna as alíquotas cadastradas na impressora.

#### Parâmetro:

*Aliquotas*: STRING com o tamanho de 79 posições para receber as alíquotas.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observação:**

- As alíquotas retornadas estarão separadas por vírgula. Ex.: 1200, 1700, 0500, 1800

**Exemplos****' Exemplo em Visual Basic**

```
Aliquotas = Space(79)
```

```
iResultado = Elgin_RetornoAliquotas(Aliquotas)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 79 do Aliquotas := Aliquotas + ' ';
```

```
iResultado := Elgin_RetornoAliquotas( Aliquotas );
```

**3.2.6.30 Elgin\_RetornoImpressora**

Lê o retorno da impressora referente ao último comando enviado.

**Parâmetros:**

*i*: Variável INTEIRA para receber o primeiro byte.

*strErroMsg*: STRING com o erro retornado pela impressora.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observação:**

- Essa função deve ser usada após a execução de qualquer outra função da DLL para ler o

[Retornos da Impressora](#) referente a função executada. Essa função devolve o status da impressora.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_RetornoImpressora(iCodErro, strErroMsg)
```

**// Exemplo em Delphi**

```
strErroMsg := StringOfChar(' ',1024);
```

```
iResultado := Elgin_RetornoImpressora( iCodErro, strErroMsg );
```

**3.2.6.31 Elgin\_SimboloMoeda**

Retorna o símbolo da moeda cadastrado na impressora.

**Parâmetro:**

*SimboloMoeda*: STRING com 2 posições para receber o símbolo da moeda.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
SimboloMoeda = Space(2)
iResultado = Elgin_SimboloMoeda(SimboloMoeda)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 2 do SimboloMoeda := SimboloMoeda + ' ';
iResultado := Elgin_SimboloMoeda( SimboloMoeda );
```

### 3.2.6.32 Elgin\_SubTotal

Retorna o valor do subtotal do cupom.

#### Parâmetro:

*SubTotal*: STRING com o tamanho de 14 posições para receber o subtotal do cupom.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
SubTotal = Space(14)
iResultado = Elgin_SubTotal(SubTotal)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 14 do SubTotal := SubTotal + ' ';
iResultado := Elgin_SubTotal( SubTotal );
```

### 3.2.6.33 Elgin\_TotalDiaTroco

Totalizador diário de troco.

#### Parâmetros:

*troco*: Variável STRING com 18 posições para receber a informação.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

### Exemplos

#### 'Exemplo em Visual Basic

```
strRetorno01 = Space(18)
```

```
iResultado = Elgin_TotalDiaTroco(strRetorno01)
```

**// Exemplo em Delphi**

```
strRetorno01 := StringOfChar('0',18);  
Elgin_TotalDiaTroco(strRetorno01);
```

### 3.2.6.34 Elgin\_TotalDocTroco

Retorna o valor total de troco do cupom atual.

**Parâmetros:**

*troco*: Variável STRING com 18 posições para receber a informação.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- Esta função só poderá ser executada após o fechamento do cupom.

**Exemplos**

**' Exemplo em Visual Basic**

```
strRetorno01 = Space(18)  
iResultado = Elgin_TotalDocTroco(strRetorno01)
```

**// Exemplo em Delphi**

```
strRetorno01 := StringOfChar('0',18);  
Elgin_TotalDocTroco(strRetorno01);
```

### 3.2.6.35 Elgin\_UltimoItemVendido

Retorna o número do último item vendido.

**Parâmetro:**

*UltimoItem*: Variável string com 4 posições para receber o número do último item vendido.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
UltimoItem = Space(4)  
iResultado = Elgin_UltimoItemVendido(UltimoItem)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 4 do UltimoItem := UltimoItem + '  
iResultado := Elgin_UltimoItemVendido( UltimoItem );
```

### 3.2.6.36 Elgin\_ValorFormaPagamento

Retorna o valor acumulado em uma determinada forma de pagamento.

**Parâmetro:**

*FormaPagamento*: STRING com até 16 posições com a descrição ou posição da Forma de Pagamento que deseja retornar o seu valor.

*Valor*: STRING com 14 posições para receber a informação.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-2: Parâmetro inválido na função.

-24: Forma de pagamento não programada.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
Valor = Space(14)
```

```
iResultado = Elgin_ValorFormaPagamento("-2", Valor)
```

**// Exemplo em Delphi**

```
cFormaPgto := 'Cheque';
```

```
for iConta := 1 to 14 do Valor := Valor + ' ';
```

```
iResultado := Elgin_ValorFormaPagamento(pchar ( cFormaPgto ), Valor );
```

### 3.2.6.37 Elgin\_ValorPagoUltimoCupom

Retorna o valor pago no último cupom.

**Parâmetro:**

*Valor*: STRING com 14 posições para receber o valor pago no último cupom.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observações:**

- Valor pago no último cupom é diferente do valor total do cupom.

**Exemplos**

Imaginemos que o valor do cupom seja R\$15,00 e foi pago pelo cliente R\$ 20,00. Teve um troco de R\$ 5,00. O valor retornado por essa função será R\$ 20,00 e não R\$ 15,00.

**' Exemplo em Visual Basic**

```
Valor = Space(14)
```

```
iResultado = Elgin_ValorPagoUltimoCupom(Valor)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 14 do Valor := Valor + ' ';
```

```
iResultado := Elgin_ValorPagoUltimoCupom( Valor );
```

### 3.2.6.38 Elgin\_ValorTotalizadorNaoFiscal

Retorna o valor acumulado em um determinado totalizador não fiscal.

**Parâmetros:**

*Totalizador*: STRING com até 19 posições com a descrição ou o índice do Totalizador.

*Valor*: STRING com 13 posições para receber a informação.

**Possíveis retornos da Função (INTEIRO):**

0: Erro de comunicação.

1: OK.

-1: Erro de execução da função.

-2: Parâmetro inválido na função.

-25: Totalizador não fiscal não programado.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
Valor = Space(13)
```

```
iResultado = Elgin_ValorTotalizadorNaoFiscal("Rec. Prestacao", Valor)
```

**// Exemplo em Delphi**

```
cDescTot := 'Rec. Prestacao';
```

```
for iConta := 1 to 13 do Valor := Valor + ' ';
```

```
iResultado := Elgin_ValorTotalizadorNaoFiscal( pchar( cDescTot ), Valor );
```

### 3.2.6.39 Elgin\_VendaBruta

Retorna o valor da venda bruta do dia.

**Parâmetros:**

*Valor*: Variável STRING com 14 posições para receber a informação.

**OBS.:** O formato do retorno da função será de acordo com o exemplo a seguir. Ex.: R\$50,00 = 0000000005000.

**Possíveis retornos da Função (INTEIRO):**

0: Erro de comunicação.

1: OK.

-1: Erro de execução da função.

-2: Parâmetro inválido na função.

-24: Forma de pagamento não programada.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
strRetorno01 := space(14)
```

```
iResultado = Elgin_VendaBruta(strRetorno01)
```

**// Exemplo em Delphi**

```
strRetorno01 := StringOfChar('0',14);
```

```
iResultado := Elgin_VendaBruta(strRetorno01);
```

### 3.2.6.40 Elgin\_VendaLiquida

Retorna o valor da venda líquida do dia.

**Parâmetros:**

*Valor*: STRING com 14 posições para receber a informação.

**OBS.:** O formato do retorno da função será de acordo com o exemplo a seguir. Ex.: R\$50,00 = 0000000005000

**Possíveis retornos da Função (INTEIRO):**

0: Erro de comunicação.

1: OK.

-1: Erro de execução da função.

-2: Parâmetro inválido na função.

-24: Forma de pagamento não programada.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
strRetorno01 = Space(14)
```

```
iResultado = Elgin_VendaLiquida(strRetorno01)
```

**// Exemplo em Delphi**

```
strRetorno01 := StringOfChar('0',14);
```

```
iResultado := Elgin_VendaLiquida(strRetorno01);
```

**3.2.6.41 Elgin\_VerificaAliquotasICMS**

Retorna as alíquotas de vinculação ao ICMS.

**Parâmetro:**

*AliquotasICMS*: STRING com 79 posições para receber as alíquotas vinculadas ao ICMS.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observação:**

- As alíquotas retornadas estarão separadas por vírgula. Ex.: 0300, 0500

**Exemplos**

**' Exemplo em Visual Basic**

```
AliquotasICMS = Space(79)
```

```
iResultado = Elgin_VerificaAliquotasICMS(AliquotasICMS)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 79 do AliquotasIss := AliquotasICMS + ' ';
```

```
iResultado := Elgin_VerificaAliquotasICMS( AliquotasICMS);
```

**3.2.6.42 Elgin\_VerificaAliquotasIss**

Retorna as alíquotas de vinculação ao ISS.

**Parâmetro:**

*AliquotasIss*: STRING com 79 posições para receber as alíquotas vinculadas ao ISS.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.



**' Exemplo em Visual Basic**

```
Departamentos = Space(1019)
iResultado = Elgin_VerificaDepartamentos(Departamentos)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 1019 do Departamentos := Departamentos + ' ';
iResultado := Elgin_VerificaDepartamentos( Departamentos );
```

**3.2.6.44 Elgin\_VerificaEstadoImpressora**

Retorna o estado da impressora.

**Parâmetros:**

*ACK*: INTEIRO para receber o primeiro byte.

*ST1*: INTEIRO para receber o segundo byte.

*ST2*: INTEIRO para receber o terceiro byte.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observação:**

- Essa função devolve o status da impressora ([Retorno da Impressora](#)).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_VerificaEstadoImpressora( iAck, iSt1, iSt2 )
```

**// Exemplo em Delphi**

```
iResultado := Elgin_VerificaEstadoImpressora( iAck, iSt1, iSt2 );
```

**3.2.6.45 Elgin\_VerificaEstadoImpressoraStr**

Retorna o estado da impressora.

**Parâmetros:**

*ACK*: STRING com o tamanho 3, para receber o primeiro byte.

*ST1*: STRING com o tamanho 3, para receber o segundo byte.

*ST2*: STRING com o tamanho 3, para receber o terceiro byte.

**Possíveis retornos da Função (INTEIRO):**

0: Erro de comunicação.

1: OK.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observação:**

- Essa função devolve o status da impressora ([Retorno da Impressora \(ACK, ST1, ST2\)](#)).

**Exemplos**



- A impressora permite programar até 16 formas de pagamento, as formas que não estiverem programadas estarão com os valores zerados e a descrição em branco (veja exemplo acima).

### Exemplos

#### ' Exemplo em Visual Basic

```
FormasPagamento = Space(3016)
iResultado = Elgin_VerificaFormasPagamento(FormasPagamento)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 3016 do FormasPagamento := FormasPagamento + ' ';
iResultado := Elgin_VerificaFormasPagamento( FormasPagamento );
```

### 3.2.6.47 Elgin\_VerificaIndiceAliquotasICMS

Retorna os índices das alíquotas de ICMS.

#### Parâmetro:

*IndiceAliquotas*: STRING com o tamanho de 48 posições para receber os índices das alíquotas de ICMS.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu.  
-1: Erro de execução da função.  
-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.  
-5: Erro ao abrir a porta de comunicação.  
-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
-50: Número de série inválido.

#### Observação:

- Os índices retornados estarão separados por vírgula. Ex.: 03, 05.

### Exemplos

#### ' Exemplo em Visual Basic

```
IndiceAliquotas = Space(48)
iResultado = Elgin_VerificaIndiceAliquotasICMS(IndiceAliquotas)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 48 do IndiceAliquotas := IndiceAliquotas + ' ';
iResultado := Elgin_VerificaIndiceAliquotasICMS( IndiceAliquotas );
```

### 3.2.6.48 Elgin\_VerificaIndiceAliquotasIss

Retorna os índices das alíquotas de ISS.

#### Parâmetro:

*IndiceAliquotas*: Variável string com o tamanho de 48 posições para receber os índices das alíquotas de ISS.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu.  
-1: Erro de execução da função.  
-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.  
-5: Erro ao abrir a porta de comunicação.  
-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
-50: Número de série inválido.

**Observação:**

- Os índices retornados estarão separados por vírgula. Ex.: 03, 05.

**Exemplos****' Exemplo em Visual Basic**

```
IndiceAliquotas = Space(48)
```

```
iResultado = Elgin_VerificaIndiceAliquotasIss(IndiceAliquotas)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 48 do IndiceAliquotas := IndiceAliquotas + ' ';
```

```
iResultado := Elgin_VerificaIndiceAliquotasIss(IndiceAliquotas);
```

**3.2.6.49 Elgin\_VerificaModoOperacao**

Verifica se a impressora está em modo normal ou em intervenção técnica.

**Parâmetro:**

*ModoOperacao*: Variável string com 1 posição para receber o modo de operação da impressora.

Onde:

1 - Modo normal

0 - Intervenção técnica.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
ModoOperacao = " "
```

```
iResultado = Elgin_VerificaModoOperacao(ModoOperacao)
```

**// Exemplo em Delphi**

```
ModoOperacao := ' ';
```

```
iResultado := Elgin_VerificaModoOperacao( ModoOperacao );
```

**3.2.6.50 Elgin\_VerificaRecebimentoNaoFiscal**

Retorna os recebimentos não fiscais não vinculados programados na impressora.

**Parâmetro:**

*Recebimentos*: STRING com 2200 posições para receber as informações.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observações:**

- Essa função retorna as informações para os 50 totalizadores não fiscais não vinculados na seguinte ordem:

Valor indicando quantas vezes cada recebimento foi utilizado	4 bytes
Valor acumulado para cada recebimento (2 casas decimais)	20 bytes
Descrição do recebimento	19 bytes

**Exemplos**

```
000200000000000000005460Conta de água ,
00150000000000000000145850Conta de Luz ,
000000000000000000000000Carnê ,
000000000000000000000000 ,
000000000000000000000000 ,
000000000000000000000000Iptu ,
0005000000000000000078437Conta de Telefone ,
.
.
.
000000000000000000000000Prestacao de contas
```

- A impressora permite programar até 50 totalizadores não fiscais não vinculados, os totalizadores que não estiverem programados estarão com os valores zerados e a descrição em branco (veja exemplo acima).

**' Exemplo em Visual Basic**

```
Recebimento = Space(2200)
iResultado = Elgin_VerificaRecebimentoNaoFiscal(Recebimento)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 2200 do Recebimento := Recebimento + ' ';
iResultado := Elgin_VerificaRecebimentoNaoFiscal(Recebimento);
```

**3.2.6.51 Elgin\_VerificaTipoImpressora**

Retorna o tipo de impressora.

**Parâmetro:**

*TipoImpressora*: INTEIRO para receber o tipo da impressora (veja abaixo em "Observações" os valores retornados).

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 1: Erro de execução da função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

**Observações:**

- O valor do tipo de impressora pode ser:

- 1 - Impressora fiscal, gaveta, autenticação.
- 2 - Impressora fiscal, gaveta, cutter.
- 3 - Impressora fiscal, presenter, autenticação.
- 4 - Impressora fiscal, presenter, cutter.
- 5 - Impressora bilhete de passagem, gaveta, autenticação.
- 6 - Impressora bilhete de passagem, gaveta, cutter.
- 7 - Impressora bilhete de passagem, presenter, autenticação.

8 - Impressora bilhete de passagem, presenter, cutter.

### Exemplos

#### 'Exemplo em Visual Basic

```
iResultado = Elgin_VerificaTipoImpressora(TipoImpressora)
```

#### // Exemplo em Delphi

```
iResultado := Elgin_VerificaTipoImpressora( TipoImpressora );
```

### 3.2.6.52 Elgin\_VerificaTipoImpressoraStr

Retorna o tipo de impressora.

#### Parâmetro:

*TipoImpressora*: STRING com o tamanho 40 para receber o tipo da impressora (veja abaixo em "Observações" os valores retornados).

#### Possíveis retornos da Função (INTEIRO):

0: Erro de comunicação.

1: OK.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Observações:

- O valor do tipo de impressora pode ser:

1 - Impressora fiscal, gaveta, autenticação.

2 - Impressora fiscal, gaveta, cutter.

3 - Impressora fiscal, presenter, autenticação.

4 - Impressora fiscal, presenter, cutter.

5 - Impressora bilhete de passagem, gaveta, autenticação.

6 - Impressora bilhete de passagem, gaveta, cutter.

7 - Impressora bilhete de passagem, presenter, autenticação.

8 - Impressora bilhete de passagem, presenter, cutter.

### Exemplos

#### 'Exemplo em Visual Basic

```
strRetorno01 = Space(' ',40)
```

```
iResultado = Elgin_VerificaTipoImpressoraStr(strRetorno01)
```

#### // Exemplo em Delphi

```
strRetorno01 := StringOfChar(' ',40);
```

```
iResultado := Elgin_VerificaTipoImpressoraStr(strRetorno01);
```

### 3.2.6.53 Elgin\_VerificaTotalizadoresNaoFiscais

Retorna a descrição dos totalizadores não fiscais programados na impressora.

#### Parâmetro:

*Totalizadores*: STRING com 179 posições para receber a descrição dos totalizadores não fiscais programados.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

- 1: Erro de execução da função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

#### Observações:

- Os totalizadores virão separados por vírgula.
- A impressora permite a programação de até 15 totalizadores não fiscais, porém essa função retorna somente os 9 primeiros cadastrados.

#### Exemplos

##### ' Exemplo em Visual Basic

```
Totalizadores = Space(179)
iResultado = Elgin_VerificaTotalizadoresNaoFiscais(Totalizadores)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 179 do Totalizadores := Totalizadores + ' ';
iResultado := Elgin_VerificaTotalizadoresNaoFiscais( Totalizadores );
```

### 3.2.6.54 Elgin\_VerificaTotalizadoresParciais

Retorna os totalizadores parciais cadastrados na impressora com as seguintes informações:

#### Parâmetro:

*Totalizadores*: STRING com o tamanho de 445 posições para receber os totalizadores parciais cadastrados.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 1: Erro de execução da função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

#### Observações:

- São retornadas as seguintes informações separadas por vírgulas:

Totalizadores parciais tributados	224 bytes
Isenção	14 bytes
Não incidência	14 bytes
Substituição	14 bytes
Totalizadores parciais não sujeitos ao ICMS	126 bytes
Sangria	14 bytes
Suprimento	14 bytes
Grande Total	18 bytes

#### Exemplos

##### ' Exemplo em Visual Basic

```
Totalizadores = Space(445)
iResultado = Elgin_VerificaTotalizadoresParciais(Totalizadores)
```

```
// Exemplo em Delphi
for iConta := 1 to 445 do Totalizadores := Totalizadores + ' ';
iResultado := Elgin_VerificaTotalizadoresParciais( Totalizadores );
```

### 3.2.6.55 Elgin\_VerificaTruncamento

Retorna 1 se a impressora estiver no modo truncamento e 0 se estiver no modo arredondamento.

#### Parâmetro:

*FlagTruncamento*: STRING com 1 posição para receber o flag de truncamento.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### 'Exemplo em Visual Basic

```
FlagTruncamento = " "
```

```
iResultado = Elgin_VerificaTruncamento(FlagTruncamento)
```

##### // Exemplo em Delphi

```
FlagTruncamento := ' ';
```

```
iResultado := Elgin_VerificaTruncamento( FlagTruncamento );
```

### 3.2.6.56 Elgin\_VerificaZPendente

Verifica se existe alguma redução Z pendente.

#### Parâmetro:

*Flag*: INTEIRO para receber um número representando o flag de redução Z pendente ou não, onde:

Flag = 1 é quando existe Z Pendente

Flag = 0 é quando não Existe Z Pendente

#### Possíveis retornos da Função (INTEIRO):

0: Erro de comunicação.

1: OK.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### 'Exemplo em Visual Basic

```
iResultado = Elgin_VerificaZPendente(iRetorno01)
```

##### // Exemplo em Delphi

```
iResultado := Elgin_VerificaZPendente(iRetorno01);
```

### 3.2.6.57 Elgin\_VersaoFirmware

Retorna a versão do firmware da impressora.

#### Parâmetro:

*VersaoFirmware*: STRING com o tamanho de 8 posições para receber a versão do firmware.

**Possíveis retornos da Função (INTEIRO):**

0: Erro de comunicação.

1: OK.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
VersaoFirmware = Space(8)
```

```
iResultado = Elgin_VersaoFirmware(VersaoFirmware)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 8 do VersaoFirmware := VersaoFirmware + ' ';
```

```
iResultado := Elgin_VersaoFirmware( VersaoFirmware );
```

**3.2.7 Funções de Autenticação e Gaveta de dinheiro****3.2.7.1 Elgin\_AcionaGaveta**

Abre a gaveta de dinheiro.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_AcionaGaveta()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_AcionaGaveta();
```

**3.2.7.2 Elgin\_VerificaEstadoGaveta**

Retorna se a gaveta está fechada ou aberta.

**Parâmetro:**

*Estado*: INTEIRO com a Variável para receber o estado da gaveta, onde:

Estado = 1 sensor em nível 1 (fechada)

Estado = 0 sensor em nível 0 (aberta)

**IMPORTANTE**

Em alguns modelos de gaveta, este retorno poderá ser invertido, ou seja, o retorno que corresponde a "gaveta aberta", poderá ser o retorno para "gaveta fechada". Caso isso ocorra, mude a chave "**ModoGaveta**" (encontrado nas configurações do arquivo [Elgin.ini](#)), para 1 (um) e efetue o teste de retorno, desta função, abrindo e fechando a gaveta.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

- 1: Erro de execução da função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_VerificaEstadoGaveta( iEstado )
```

#### // Exemplo em Delphi

```
iResultado := Elgin_VerificaEstadoGaveta( iEstado );
```

### 3.2.7.3 Elgin\_VerificaEstadoGavetaStr

Retorna se a gaveta está fechada ou aberta.

#### Parâmetros:

*Estado*: STRING com a variável para receber o estado da gaveta, onde:

Estado = 1 sensor em nível 1 (aberta)

Estado = 0 sensor em nível 0 (fechada)

#### Possíveis retornos da Função (INTEIRO):

0: Erro de comunicação.

1: OK.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
strRetorno01 := Space(1);
```

```
iResultado = Elgin_VerificaEstadoGavetaStr(strRetorno01)
```

#### // Exemplo em Delphi

```
strRetorno01 := StringOfChar(' ',1);
```

```
iResultado := Elgin_VerificaEstadoGavetaStr(strRetorno01);
```

### 3.2.8 Funções para impressoras MFD

#### 3.2.8.1 Elgin\_AbreComprovanteNaoFiscalVinculadoMFD

Abre o Comprovante Não Fiscal Vinculado.

#### Parâmetros:

**FormaPagamento**: STRING com a Forma de Pagamento com até 16 caracteres.

**Valor**: STRING com o Valor Pago na forma de pagamento do cupom a que se refere o comprovante, com até 13 dígitos (2 casas decimais).

**NumeroCupom**: STRING com o Número do cupom a que se refere o comprovante com até 6 dígitos.

**CGC**: STRING com até 29 caracteres com o CGC ou CPF do cliente.

**Nome**: STRING com até 30 caracteres com o nome do cliente.

**Endereco**: STRING com até 80 caracteres com o endereço do cliente.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

- 2: Parâmetro inválido.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observações:**

- A forma de pagamento utilizada não pode ser Dinheiro.
- A forma de pagamento deve ter sido utilizada no cupom ao qual se refere o comprovante.
- Poderá ser emitido um comprovante não fiscal para cada parcela informada no cupom fiscal. Os parâmetros "Valor" e "Numero do Cupom" tornam-se obrigatórios se o comprovante emitido não for referente ao último cupom fiscal emitido.
- Caso o comprovante emitido seja referente ao último cupom fiscal, é opcional informar o número do cupom.
- O parâmetro "Numero do Cupom" torna-se obrigatório se o comprovante emitido não for referente ao último cupom fiscal emitido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_AbreComprovanteNaoFiscalVinculadoMFD("Cartao", "50,00", "000165", "1.111.111-1", "Fulano de Tal", "R. Sem Fim, 1000")
```

**// Exemplo em Delphi**

```
cFormaPgto := 'Cartao';  
cValorPgto := '50,00';  
cCOO := '000165';  
cCPF := '1.111.111-1';  
cNomeCliente := 'Fulano de Tal';  
cEnderecoCliente := 'R. Sem Fim, 1000';  
iResultado := Elgin_AbreComprovanteNaoFiscalVinculadoMFD(pchar( cFormaPgto ), pchar( cValorPgto ),  
pchar( cCOO ), pchar( cCPF ), pchar( cNomeCliente ), pchar( cEnderecoCliente ) );
```

**3.2.8.2 Elgin\_AbreCupomMFD**

Abre o cupom fiscal na impressora MFD.

**Parâmetros:**

**CGC\_CPF:** String até 26 caracteres com o CGC ou CPF do cliente.

**Nome:** String até 30 caracteres com o nome do cliente.

**Endereco:** String até 80 caracteres com o endereço do cliente.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observação:**

- O cupom fiscal deve estar fechado.

- Caso não seja informado **nenhum** dos parâmetros **não** será impresso no cupom as informações do consumidor (CNPJ/CPF consumidor, Nome e Endereço).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_AbreCupomMFD("10.123.154-98", "Fulano de Tal", "R. Sem Fim, 1000")
```

```
// Exemplo em Delphi
cCPF := '10.123.154-98';
cNome := 'Fulano de Tal';
cEndereco := 'R. Sem Fim, 1000';
iResultado := Elgin_AbreCupomMFD(pchar( cCPF ), pchar( cNome ), pchar( cEndereco ) );
```

### 3.2.8.3 Elgin\_AbreRecebimentoNaoFiscalMFD

Abre o comprovante não fiscal não vinculado para que sejam lançados os recebimentos não fiscais.

#### Parâmetros:

**CGC:** String até 29 caracteres com o CGC ou CPF do cliente.

**Nome:** String até 30 caracteres com o nome do cliente.

**Endereco:** String até 80 caracteres com o endereço do cliente.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_AbreRecebimentoNaoFiscalMFD("1.111.111-1", "Fulano de Tal", "Rua Sem Fim, 1000")
```

##### // Exemplo em Delphi

```
cCPF := '1.111.111-1';
cNomeCliente := 'Fulano de Tal';
cEnderecoCliente := 'Rua Sem Fim, 1000';
iResultado := Elgin_AbreRecebimentoNaoFiscalMFD(pchar( cCPF ), pchar( cNomeCliente ), pchar(
cEnderecoCliente ) );
```

### 3.2.8.4 Elgin\_AbreRelatorioGerencialMFD

Abre Relatório Gerencial, na impressora fiscal MFD.

#### Parâmetro:

**Índice:** STRING numérica com o valor entre 0 e 19, com o índice do relatório.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Observações:

- O relatório deve estar programado.

- O relatório permanecerá aberto por 2 minutos, se não for enviado o comando de fechamento durante esse período, o relatório será fechado automaticamente.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_AbreRelatorioGerencialMFD("05")
```

```
// Exemplo em Delphi
cIndice := '5';
iResultado := Elgin_AbreRelatorioGerencialMFD(pchar ( cIndice ) );
```

### 3.2.8.5 Elgin\_AcrescimoDescontoItemMFD

Efetua acréscimo ou desconto em qualquer item enquanto o cupom fiscal não estiver totalizado.

#### Parâmetros:

**Item:** String numérica até 3 dígitos com o número do item.

**AcrescimoDesconto:** Indica se é acréscimo ou desconto. 'A' para acréscimo ou 'D' para desconto.

**TipoAcrescimoDesconto:** Indica se o acréscimo ou desconto é por valor ou por percentual. '\$' para desconto por valor e '%' para percentual.

**ValorAcrescimoDesconto:** String com no máximo 14 dígitos para acréscimo ou desconto por valor e 4 dígitos para acréscimo ou desconto percentual.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_AcrescimoDescontoItemMFD("005", "D", "$", "1000")
```

##### // Exemplo em Delphi

```
cNumeroItem := '005';
```

```
cAcresDesc := 'D';
```

```
cTipoAcresDesc := '%';
```

```
cValorAcresDesc := '1000';
```

```
iResultado := Elgin_AcrescimoDescontoItemMFD(pchar ( cNumeroItem ), pchar( cAcresDesc ), pchar( cTipoAcresDesc ), pchar( cValorAcresDesc ) );
```

### 3.2.8.6 Elgin\_AcrescimoDescontoSubtotalMFD

Efetua acréscimo ou desconto em subtotal do cupom.

#### Parâmetros:

**Flag:** String com "A" para Acréscimo ou "D" para Desconto.

**Tipo:** String com "\$" para Acréscimo ou Desconto por valor, ou "%" para Acréscimo ou Desconto percentual.

**Valor:** String com o valor no máximo de 14 dígitos para Acréscimo ou Desconto, ou valor com 4 dígitos para Acréscimo ou Desconto por percentual.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Observações:

- O cupom deve estar subtotalizado.

- O cupom não pode estar totalizado.

- É permitido o registro de apenas uma operação válida para acréscimo e uma para desconto no subtotal.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_AcrescimoDescontoSubtotalMFD("A", "%", "10,00")
```

#### // Exemplo em Delphi

```
cAcresDesc := 'A';
cTipoAcresDesc := '%';
cValorAcresDesc := '10,00';
iResultado := Elgin_AcrescimoDescontoSubtotalMFD(pchar( cAcresDesc ), pchar( cTipoAcresDesc ), pchar(
cValorAcresDesc ) );
```

### 3.2.8.7 Elgin\_AcrescimoDescontoSubtotalRecebimentoMFD

Efetua acréscimo ou desconto em subtotal do recebimento não fiscal.

#### Parâmetros:

**Flag:** String com "A" para Acréscimo ou "D" para Desconto.

**Tipo:** String com "\$" para acréscimo ou desconto por valor, ou "%" para acréscimo ou desconto por percentual.

**Valor:** String com no máximo 14 dígitos para o valor ou 4 dígitos para o percentual.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Observações:

- O recebimento não fiscal deve estar subtotalizado.

- O recebimento não fiscal não pode estar totalizado.

- É permitido o registro de apenas uma operação válida para acréscimo e uma para desconto no subtotal.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_AcrescimoDescontoSubtotalRecebimentoMFD("D", "$", "15,00")
```

#### // Exemplo em Delphi

```
cAcresDesc := 'D';
cTipoAcresDesc := '$';
cValorAcresDesc := '15,00';
iResultado := Elgin_AcrescimoDescontoSubtotalRecebimentoMFD(pchar( cAcresDesc ), pchar(
cTipoAcresDesc ), pchar( cValorAcresDesc ) );
```

### 3.2.8.8 Elgin\_AcrescimoItemNaoFiscalMFD

Efetua um acréscimo ou desconto em um item não fiscal na impressora fiscal MFD.

#### Parâmetros:

**NumeroItem:** STRING contendo o número do item não fiscal, com até 3 caracteres.

**AcrescimoDesconto:** STRING com "A" para Acréscimo ou "D" para desconto.

**TipoAcrescimoDesconto:** STRING com "\$" para Acréscimo ou Desconto por valor, ou "%" para

para percentual.

**ValorAcrecimoDesconto:** STRING com o valor do desconto ou Acréscimo.

#### **Retornos da Função (INTEIRO):**

0: erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### **Observação:**

- O cupom não fiscal não pode estar subtotalizado.

#### **Exemplos:**

##### **' Exemplo em Visual Basic**

```
iResultado = Elgin_AcrecimoItemNaoFiscalMFD("005", "A", "$", "10,00")
```

##### **// Exemplo em Delphi**

```
cNumeroItem := '005';
```

```
cAcresDesc := 'A';
```

```
cTipoAcresDesc := '$';
```

```
cValorAcresDesc := '10,00';
```

```
iResultado := Elgin_AcrecimoItemNaoFiscalMFD( pchar( cNumeroItem ), pchar( cAcresDesc ), pchar( cTipoAcresDesc ), pchar( cValorAcresDesc ) );
```

#### **3.2.8.9 Elgin\_CancelaAcrecimoDescontoItemMFD**

Cancela a acréscimo ou a desconto dado no item.

#### **Parâmetros:**

**Flag:** String com "A" para cancelar o Acréscimo ou "D" para cancelar o Desconto.

**Item:** String de até 3 dígitos com o número do item a ser cancelado restrito aos 300 últimos registros efetuados.

#### **Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### **Exemplos**

##### **' Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaAcrecimoDescontoItemMFD("A", "005")
```

##### **// Exemplo em Delphi**

```
cAcresDesc := 'A';
```

```
cNumeroItem := '005';
```

```
iResultado := Elgin_CancelaAcrecimoDescontoItemMFD(pchar( cAcresDesc ), pchar( cNumeroItem ) );
```

#### **3.2.8.10 Elgin\_CancelaAcrecimoDescontoSubtotalMFD**

Cancela acréscimo e desconto efetuados em subtotal do cupom.

#### **Parâmetro:**

**Flag:** String com "A" para cancelar o Acréscimo ou "D" para cancelar o Desconto, dado no subtotal.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- O cupom deve estar subtotalizado.

- O cupom deve ter operações anteriores de acréscimo ou desconto em subtotal.

- O cupom não pode estar totalizado.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaAcrecimoDescontoSubtotalMFD("D")
```

**// Exemplo em Delphi**

```
cAcrecDesc := 'D';
```

```
iResultado := Elgin_CancelaAcrecimoDescontoSubtotalMFD(pchar ( cAcrecDesc ));
```

**3.2.8.11 Elgin\_CancelaAcrecimoDescontoSubtotalRecebimentoMFD**

Cancela acréscimo e desconto efetuados em subtotal do recebimento não fiscal.

**Parâmetro:**

**Flag:** String com "A" para cancelar o Acréscimo ou "D" para cancelar o Desconto, dado no subtotal do recebimento.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- O recebimento não fiscal deve estar subtotalizado.

- O recebimento não fiscal deve ter operações anteriores de acréscimo ou desconto em subtotal.

- O recebimento não fiscal não pode estar totalizado.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaAcrecimoDescontoSubtotalRecebimentoMFD("D")
```

**// Exemplo em Delphi**

```
cAcrecDesc := 'D';
```

```
iResultado := Elgin_CancelaAcrecimoDescontoSubtotalRecebimentoMFD(pchar ( cAcrecDesc ));
```

**3.2.8.12 Elgin\_CancelaCupomMFD**

Cancela o último cupom emitido.

**Parâmetros:**

**CGC\_CPF:** String até 29 caracteres com o CGC ou CPF do cliente.

**Nome:** String até 30 caracteres com o nome do cliente.

**Endereco:** String até 80 caracteres com o endereço do cliente.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro Inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaCupomMFD("11.111.111-11", "Fulano de Tal", "R. Sem Fim, 1000")
```

**// Exemplo em Delphi**

```
cCPF := '11.111.111-11';
```

```
cNomeCliente := 'Fulano de Tal';
```

```
cEnderecoCliente := 'R. Sem Fim, 1000';
```

```
iResultado := Elgin_CancelaCupomMFD(pchar( cCPF ), pchar( cNomeCliente ), pchar( cEnderecoCliente ) );
```

**3.2.8.13 Elgin\_CancelaRecebimentoNaoFiscalMFD**

Cancela o recebimento não fiscal.

**Parâmetros:**

**CGC:** String até 29 caracteres com o CGC ou CPF do cliente.

**Nome:** String até 30 caracteres com o nome do cliente.

**Endereco:** String até 80 caracteres com o endereço do cliente.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos:**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_CancelaRecebimentoNaoFiscalMFD("11.111.111-11", "Fulano de Tal", "R. Sem Fim, 1000")
```

**// Exemplo em Delphi**

```
cCPF := '11.111.111-11';
```

```
cNomeCliente := 'Fulano de Tal';
```

```
cEnderecoCliente := 'R. Sem Fim, 1000';
```

```
iResultado := Elgin_CancelaRecebimentoNaoFiscalMFD(pchar( cCPF ), pchar( cNomeCliente ), pchar( cEnderecoCliente ) );
```

**3.2.8.14 Elgin\_CNPJMFD**

Retorna o CNPJ do cliente cadastrado na impressora.

**Parâmetro:**

**CNPJ:** Variável String com o tamanho de 20 posições para receber o CNPJ.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.  
 1: indica que nenhum erro ocorreu.  
 -4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.  
 -5: Erro ao abrir a porta de comunicação.  
 -8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
 -50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
sCNPJ = space(20)
iResultado = Elgin_CNPJMFD(sCNPJ)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 20 do sCNPJ := sCNPJ + ' ';
iResultado := Elgin_CNPJMFD( sCNPJ );
```

### 3.2.8.15 Elgin\_ComprovantesNaoFiscaisNaoEmitidosMFD

Retorna o número de comprovantes não fiscais não emitidos.

#### Parâmetro:

**Comprovantes:** Variável String com o tamanho de 4 posições para receber a informação.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.  
 1: indica que nenhum erro ocorreu.  
 -4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.  
 -5: Erro ao abrir a porta de comunicação.  
 -8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
 -50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
sComp = space(4)
iResultado = Elgin_ComprovantesNaoFiscaisNaoEmitidosMFD(sComp)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 4 do sComp := sComp + ' ';
iResultado := Elgin_ComprovantesNaoFiscaisNaoEmitidosMFD( sComp );
```

### 3.2.8.16 Elgin\_ContadorComprovantesCreditoMFD

Retorna o número de comprovantes de crédito emitidos.

#### Parâmetro:

**Comprovantes:** Variável String com o tamanho de 4 posições para receber a informação.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.  
 1: indica que nenhum erro ocorreu.  
 -4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.  
 -5: Erro ao abrir a porta de comunicação.  
 -8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
 -50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
sComp = space(4)
```

```
iResultado = Elgin_ContadorComprovantesCreditoMFD(sComp)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 4 do sComp := sComp + ' ';  
iResultado := Elgin_ContadorComprovantesCreditoMFD( sComp );
```

### 3.2.8.17 Elgin\_ContadorCupomFiscalMFD

Retorna o número de cupons fiscais emitidos.

**Parâmetro:**

**CuponsEmitidos:** Variável String com o tamanho de 6 posições para receber a informação.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
sCupom = space(6)  
iResultado = Elgin_ContadorCupomFiscalMFD(sCupom)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 6 do sCupom := sCupom + ' ';  
iResultado := Elgin_ContadorCupomFiscalMFD( sCupom );
```

### 3.2.8.18 Elgin\_ContadoresTotalizadoresNaoFiscaisMFD

Retorna o número de vezes em que os totalizadores não sujeitos ao ICMS foram usados.

**Parâmetro:**

**Contadores:** Variável String com 149 posições para receber as informações.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1 - Erro de execução.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observações:**

- O conteúdo da variável retornada será 120 dígitos separados de 4 em 4 por vírgula que representam os 30 totalizadores.

Ex.: "0001,0003,0001,0005,0004,0002,0003,0004,0007 ..."

- O primeiro valor corresponde ao número de vezes que o totalizador 01 foi usado, o segundo corresponde ao totalizador 02 e assim sucessivamente.

**Exemplos**

**' Exemplo em Visual Basic**

```
sCont = space(149)  
iResultado = Elgin_ContadoresTotalizadoresNaoFiscaisMFD(sCont)
```

```
// Exemplo em Delphi
for iConta := 1 to 149 do sCont := sCont + ' ';
iResultado := Elgin_ContadoresTotalizadoresNaoFiscaisMFD( sCont );
```

### 3.2.8.19 Elgin\_ContadorFitaDetalheMFD

Retorna o número de vezes em que foi impressa a fita detalhe.

#### Parâmetro:

**ContadorFita:** Variável String com o tamanho de 6 posições para receber a informação.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
sCont = space(6)
```

```
iResultado = Elgin_ContadorFitaDetalheMFD(sCont)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 6 do sCont := sCont + ' ';
```

```
iResultado := Elgin_ContadorFitaDetalheMFD( sCont );
```

### 3.2.8.20 Elgin\_ContadorOperacoesNaoFiscaisCanceladasMFD

Retorna o número de operações não fiscais canceladas.

#### Parâmetro:

**OperacoesCanceladas:** Variável String com o tamanho de 4 posições para receber a informação.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
sOp = space(4)
```

```
iResultado = Elgin_ContadorOperacoesNaoFiscaisCanceladasMFD(sOp)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 4 do sOp := sOp + ' ';
```

```
iResultado := Elgin_ContadorOperacoesNaoFiscaisCanceladasMFD( sOp );
```

### 3.2.8.21 Elgin\_ContadorRelatoriosGerenciaisMFD

Retorna o número de relatórios gerenciais emitidos.

#### Parâmetro:

**Relatorios:** Variável String com o tamanho de 6 posições para receber a informação.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
sRel = space(6)
```

```
iResultado = Elgin_ContadorRelatoriosGerenciaisMFD(sRel)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 6 do sRel := sRel + ' ';
```

```
iResultado := Elgin_ContadorRelatoriosGerenciaisMFD( sRel );
```

**3.2.8.22 Elgin\_CupomAdicionalMFD**

Emite um cupom adicional com as informações do COO e valor do cupom fiscal.

**Parâmetro:**

Não há

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_CupomAdicionalMFD()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_CupomAdicionalMFD();
```

**3.2.8.23 Elgin\_DadosUltimaReducaoMFD**

Retorna os dados da impressora no momento da última redução Z.

**Parâmetro:**

**DadosReducao:** Variável String com o tamanho de 1278 posições para receber os dados da última redução.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observações:**

- São retornados os valores das seguintes informações separados por vírgulas:

Modo de redução Z:	2 bytes (se 00 redução por comando se 01 redução automática)
Contador de reinício de operação:	4 bytes
Contador de redução z:	4 bytes
Contador de ordem de operação:	6 bytes
Contador Geral de operações não fiscais:	6 bytes
Contador de cupom fiscal:	6 bytes
Contador Geral de relatório gerencial:	6 bytes
Contador de fita detalhe emitida:	6 bytes
Contador de operação não fiscal cancelada:	4 bytes
Contador de cupom fiscal cancelado:	4 bytes
Contadores específicos de operações não fiscais:	120 bytes
Contadores específicos de relatórios gerenciais:	120 bytes
Contador de comprovantes de débito ou crédito:	4 bytes
Contador de comprovantes de débito ou crédito não emitidos:	4 bytes
Contador de comprovantes de débito ou crédito cancelados:	4 bytes
Totalizador geral:	18 bytes
Totalizadores Parciais Tributados:	224 bytes
Totalizador de isenção de ICMS:	14 bytes
Totalizador de não incidência de ICMS:	14 bytes
Totalizador de substituição tributária de ICMS:	14 bytes
Totalizador de isenção de ISSQN:	14 bytes
Totalizador de não incidência de ISSQN:	14 bytes
Totalizador de substituição tributária de ISSQN:	14 bytes
Totalizador de descontos em ICMS:	14 bytes
Totalizador de descontos em ISSQN:	14 bytes
Totalizador de acréscimos em ICMS:	14 bytes
Totalizador de acréscimos em ISSQN:	14 bytes
Totalizador de cancelamentos em ICMS:	14 bytes
Totalizador de cancelamentos em ISSQN:	14 bytes
Totalizadores parciais não sujeitos ao ICMS:	392 bytes
Totalizador de sangria:	14 bytes
Totalizador de suprimento:	14 bytes
Totalizador de cancelamentos de não fiscais:	14 bytes
Totalizador de descontos de não fiscais:	14 bytes
Totalizador de acréscimos de não fiscais:	14 bytes
Alíquotas tributárias:	64 bytes
Data do movimento:	6 bytes

### Exemplos

#### ' Exemplo em Visual Basic

```
sDados = space(1242)
iResultado = Elgin_DadosUltimaReducaoMFD(sDados)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 1242 do sDados := sDados + ' ';
```

```
iResultado := Elgin_DadosUltimaReducaoMFD( sDados );
```

### 3.2.8.24 Elgin\_DataHoraUltimoDocumentoMFD

Retorna a data e hora do último documento armazenado na MFD no formato dd/mm/aa hh/mm/ss (sem barras e espaço).

**Parâmetro:**

**DataHora:** Variável String com o tamanho de 12 posições para receber os dados.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
sDataHora = space(12)
```

```
iResultado = Elgin_DataHoraUltimoDocumentoMFD(sDataHora)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 12 do sDataHora := sDataHora + ' ';
```

```
iResultado := Elgin_DataHoraUltimoDocumentoMFD( sDataHora );
```

### 3.2.8.25 Elgin\_DataMovimentoUltimaReducaoMFD

Retorna a data do movimento da última redução Z.

**Parâmetro:**

**cDataMovimento:** Variável STRING para receber a informação no padrão DDMMAA.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função. .

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos:**

**' Exemplo em Visual Basic**

```
cDataMovimento = space(6)
```

```
iResultado = Elgin_DataMovimentoUltimaReducaoMFD(cDataMovimento)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 6 do cDataMovimento := cDataMovimento + ' ';
```

```
iResultado := Elgin_DataMovimentoUltimaReducaoMFD( cDataMovimento);
```

### 3.2.8.26 Elgin\_DownloadMFD

Realiza o download da MFD (Memória de Fita Detalhe) das impressoras fiscais térmicas Elgin.

**Parâmetros:**

**cArquivo:** String com o nome do arquivo que será gerado, exemplo: "c:\DOWNLOAD.txt".

**cTipoDownload:** String com o tipo de download, onde:

"0" (zero): download total.

"1" (um): download por data.

"2" (dois): download por COO.

**cDadoInicial:** String com a data ou o COO inicial (data no formato DDMMAA, COO com no máximo 6 dígitos). Este parâmetro é obrigatório se o tipo de download for por data ou por COO..

**cDadoFinal:** String com a data ou o COO final (data no formato DDMMAA, COO com no máximo 6 dígitos). Este parâmetro é obrigatório se o tipo de download for por data ou por COO..

**cUsuario:** String com o número de ordem do proprietário do ECF, exemplo: primeiro proprietário "cUsuario = 1". Pois para cada proprietário o COO dos cupons serão diferentes. Este parâmetro é obrigatório se o download for por COO. Caso o download seja por data seja por data o parâmetro cUsuario deve ser passado vazio, porque o numero de proprietario e datas nao devem ser informados simultaneamente na emissao de fita detalhe-

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar o arquivo.

-50: Número de série inválido.

#### Observações:

- Esta função é utilizada somente nas impressoras fiscais térmicas.

- Recomendamos que seja feito o tratamento de exceção (ex: try... except) para esta função.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_DownloadMFD("c:\DOWNLOAD.txt", "2", "000001", "000010", "1")
```

##### // Exemplo em Delphi

```
iResultado := Elgin_DownloadMFD('c:\Download.txt','1','120607','140607','');
```

### 3.2.8.27 Elgin\_EfetuaFormaPagamentoMFD

Imprime a(s) forma(s) de pagamento e o(s) valor(es) pago(s) nessa forma.

#### Parâmetros:

**FormaPagamento:** String com a forma de pagamento com no máximo 16 caracteres, contendo a descrição ou código da forma de pagamento.

**ValorFormaPagamento:** String com o valor da forma de pagamento com até 14 dígitos.

**Parcelas:** String numérica entre 1 e 24 com o número de parcelas em que o pagamento será realizado.

**DescricaoFormaPagto:** String com a descrição da forma de pagamento com no máximo 80 caracteres.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-24 - Forma de Pagamento não programada.

-50: Número de série inválido.

#### Observações:

- O número de parcelas é utilizado para emissão do comprovante não fiscal vinculado. Poderá ser

emitido um comprovante para cada parcela.

- O fechamento do cupom com formas de pagamento deve ter sido iniciado.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_EfetuaFormaPagamentoMFD("Cartão VISA", "50,00", "2", "Compra parcelada")
```

#### // Exemplo em Delphi

```
cFormaPgto := 'Cartão VISA';
```

```
cValorPago := '50,00';
```

```
cNumeroParcelas := '2';
```

```
cMsg := 'Compra parcelada';
```

```
iResultado := Elgin_EfetuaFormaPagamentoMFD(pchar ( cFormaPgto ), pchar( cValorPago ), pchar( cNumeroParcelas ), pchar( cMsg ) );
```

### 3.2.8.28 Elgin\_EfetuaRecebimentoNaoFiscalMFD

Efetua o recebimento não fiscal.

#### Parâmetros:

**ÍndiceTotalizador:** STRING com o Índice do Totalizador com até 2 dígitos para o recebimento.

**ValorRecebimento:** STRING com o Valor do recebimento com até 14 dígitos (duas casas decimais).

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função. .

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Observações:

- O comprovante deve ter sido aberto anteriormente.

- O totalizador não fiscal deve estar programado.

- Caso não seja passado a virgula para definir as casas decimais do valor, o função irá considerar os dois últimos dígitos como casas decimais. Ex.: ValorRecebimento = "2000" significa 20,00.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_EfetuaRecebimentoNaoFiscalMFD("03", "35,00")
```

#### // Exemplo em Delphi

```
cIndice := '03';
```

```
cValor := '35,00';
```

```
iResultado := Elgin_EfetuaRecebimentoNaoFiscalMFD( pchar ( cIndice ), pchar( cValor ) );
```

### 3.2.8.29 Elgin\_EstornoNaoFiscalVinculadoMFD

Estorna os lançamentos de um comprovante de crédito ou débito vinculado. Deve ser executado imediatamente após a impressão do comprovante vinculado.

#### Parâmetros:

**CGC:** STRING até 29 caracteres com o CGC ou CPF do cliente

**Nome:** STRING até 30 caracteres com o nome do cliente.

**Endereco:** STRING até 80 caracteres com o endereço do cliente.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

- 1: indica que nenhum erro ocorreu.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

#### Observações:

- Pode se usar a função [Elgin\\_UsaComprovanteNaoFiscalVinculado](#), para imprimir um texto qualquer, dentro deste cupom de estorno.
- Deve-se usar a função [Elgin\\_FechaComprovanteNaoFiscalVinculado](#), para fechar este cupom de estorno.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_EstornoNaoFiscalVinculadoMFD("11.111.111-11", "Maria", "Rua 11")
```

##### // Exemplo em Delphi

```
cCPF := '11.111.111-11';
cNomeCliente := 'Maria';
cEnderecoCliente := 'Rua 11';
iResultado := Elgin_EstornoNaoFiscalVinculadoMFD( pchar( cCPF ), pchar( cNomeCliente ), pchar(
cEnderecoCliente ) );
```

#### 3.2.8.30 Elgin\_FechaRecebimentoNaoFiscalMFD

Termina o fechamento do recebimento não fiscal.

#### Parâmetro:

**Mensagem:** String com a Mensagem promocional com até 490 caracteres.

#### Retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_FechaRecebimentoNaoFiscalMFD("Obrigado, volte sempre !!!")
```

##### // Exemplo em Delphi

```
cMsgPromocional := 'Obrigado, volte sempre !!!';
iResultado := Elgin_FechaRecebimentoNaoFiscalMFD(pchar( cMsgPromocional ) );
```

#### 3.2.8.31 Elgin\_FormatoDadosMFD

Gera os dados da MFD (Memória de Fita Detalhe) em formato TXT ou RTF.

#### Parâmetros:

**cArquivoOrigem:** String com o nome do arquivo MFD que contém a origem dos dados MFD, exemplo: "c:\DOWNLOAD.MFD".

**cDestino:** String com o nome do arquivo que será gerado, exemplo: "c:\SAIDA.txt".

**cFormatoDados:** String com o tipo do formato de dados, onde:

- "0" (zero) para formato .TXT (é criado um arquivo TXT com a imagem dos cupons referentes ao movimento)
- "1" (um) para formato .RTF (é criado um arquivo RTF com a imagem dos cupons referentes

ao movimento)

**cTipoDownload:** String com o tipo de download, onde:

"0" (zero): download total

"1" (um): download por data

"2" (dois): download por COO

**cDadoInicial:** String com a data ou o COO inicial (data no formato DDMMAA, COO com no máximo 6 dígitos).

**cDadoFinal:** String com a data ou o COO final (data no formato DDMMAA, COO com no máximo 6 dígitos).

**cUsuario:** String com o número de ordem do proprietário do ECF, exemplo: primeiro proprietário "cUsuario = 1". Pois para cada proprietário o COO dos cupons serão diferentes.

#### **Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-2: Parâmetro inválido.

-8: Erro ao criar ou gravar o arquivo.

-50: Número de série inválido.

#### **Observações:**

- Executa-se Elgin\_DownloadMFD com os parâmetros desejados e gera-se um arquivo

- Gerado o arquivo, executa-se Elgin\_FormatoDadosMFD para conversão em TXT ou RTF

- Os parâmetros "TipoDownload", "ParametroInicial", "ParametroFinal" e "UsuarioECF" desta função não tem prioridade sob os mesmos parâmetros da função Elgin\_DownloadMFD, ou seja, sempre serão considerados os parâmetros setados na geração do arquivo pela função Elgin\_DownloadMFD.

#### **Exemplos**

##### **' Exemplo em Visual Basic**

```
iResultado = Elgin_FormatoDadosMFD("c:\DOWNLOAD.MFD", "c:\SAIDA.RTF", "1", "2", "000001", "000010", "1")
```

##### **// Exemplo em Delphi**

```
cArquivoOrigem := 'c:\DOWNLOAD.MFD';
```

```
cArquivoDestino := 'c:\SAIDA.RTF';
```

```
cTipoFormato := '1';
```

```
cTipoDownload := '2';
```

```
cCOOInicial := '000001';
```

```
cCOOFinal := '000010';
```

```
cUsuario := '1';
```

```
iResultado := Elgin_FormatoDadosMFD(pchar( cArquivoOrigem ), pchar( cArquivoDestino ), pchar( cTipoFormato ), pchar( cTipoDownload ), pchar( cCOOInicial ), pchar( cCOOFinal ), pchar( cUsuario ) );
```

#### **3.2.8.32 Elgin\_GrandeTotalUltimaReducaoMFD**

Retorna o grande total (GT) da última redução Z.

#### **Parâmetro:**

**cGT:** Variável String para receber a informação.

#### **Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
cGT = space(18)
iResultado = Elgin_GrandeTotalUltimaReducaoMFD(cGT)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 18 do cGT := cGT + ' ';
iResultado := Elgin_GrandeTotalUltimaReducaoMFD( cGT );
```

**3.2.8.33 Elgin\_HabilitaDesabilitaRetornoEstendidoMFD**

Habilita e desabilita o retorno estendido na MFD. O retorno estendido é ACK, ST1, ST2 e ST3. Caso não seja habilitado, será retornado apenas ACK, ST1 e ST2.

**Parâmetro:**

**FlagRetorno:** String com o valor um (1) para habilitar ou zero (0) para desabilitar o retorno estendido.

**Retornos da Função (INTEIRO):**

1: indica que nenhum erro ocorreu

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_HabilitaDesabilitaRetornoEstendidoMFD("1")
```

**// Exemplo em Delphi**

```
cFlag := '1';
iResultado := Elgin_HabilitaDesabilitaRetornoEstendidoMFD(pchar ( cFlag ) );
```

**Tabela de retornos para o ST3:**

ST3	Informação de Retorno
0	COMANDO OK
1	COMANDO INVÁLIDO
2	ERRO DESCONHECIDO
3	NÚMERO DE PARÂMETRO INVÁLIDO
4	TIPO DE PARÂMETRO INVÁLIDO
5	TODAS ALÍQUOTAS JÁ PROGRAMADAS
6	TOTALIZADOR NÃO FISCAL JÁ PROGRAMADO
7	CUPOM FISCAL ABERTO
8	CUPOM FISCAL FECHADO
9	ECF OCUPADO
10	IMPRESSORA EM ERRO
11	IMPRESSORA SEM PAPEL
12	MPRESSORA COM CABEÇA LEVANTADA
13	MPRESSORA OFF LINE
14	ALÍQUOTA NÃO PROGRAMADA
15	TERMINADOR DE STRING FALTANDO
16	ACRÉSCIMO OU DESCONTO MAIOR QUE O TOTAL DO CUPOM FISCAL
17	CUPOM FISCAL SEM ITEM VENDIDO
18	COMANDO NÃO EFETIVADO
19	SEM ESPAÇO PARA NOVAS FORMAS DE PAGAMENTO
20	FORMA DE PAGAMENTO NÃO PROGRAMADA
21	ÍNDICE MAIOR QUE NÚMERO DE FORMA DE PAGAMENTO
22	FORMAS DE PAGAMENTO ENCERRADAS
23	CUPOM NÃO TOTALIZADO
24	COMANDO MAIOR QUE 7Fh (127d)
25	CUPOM FISCAL ABERTO E SEM ÍTEM
26	CANCELAMENTO NÃO IMEDIATAMENTE APÓS
27	CANCELAMENTO JÁ EFETUADO
28	COMPROVANTE DE CRÉDITO OU DÉBITO NÃO PERMITIDO OU JÁ EMITIDO
29	MEIO DE PAGAMENTO NÃO PERMITE TEF
30	SEM COMPROVANTE NÃO FISCAL ABERTO

31	COMPROVANTE DE CRÉDITO OU DÉBITO JÁ ABERTO
32	REIMPRESSÃO NÃO PERMITIDA
33	COMPROVANTE NÃO FISCAL JÁ ABERTO
34	TOTALIZADOR NÃO FISCAL NÃO PROGRAMADO
35	CUPOM NÃO FISCAL SEM ÍTEM VENDIDO
36	ACRÉSCIMO E DESCONTO MAIOR QUE TOTAL CNF
37	MEIO DE PAGAMENTO NÃO INDICADO
38	MEIO DE PAGAMENTO DIFERENTE DO TOTAL DO RECEBIMENTO
39	NÃO PERMITIDO MAIS DE UMA SANGRIA OU SUPRIMENTO
40	RELATÓRIO GERENCIAL JÁ PROGRAMADO
41	RELATÓRIO GERENCIAL NÃO PROGRAMADO
42	RELATÓRIO GERENCIAL NÃO PERMITIDO
43	MFD NÃO INICIALIZADA
44	MFD AUSENTE
45	MFD SEM NÚMERO DE SÉRIE
46	MFD JÁ INICIALIZADA
47	MFD LOTADA
48	CUPOM NÃO FISCAL ABERTO
49	MEMÓRIA FISCAL DESCONECTADA
50	MEMÓRIA FISCAL SEM NÚMERO DE SÉRIE DA MFD
51	MEMÓRIA FISCAL LOTADA
52	DATA INICIAL INVÁLIDA
53	DATA FINAL INVÁLIDA
54	CONTADOR DE REDUÇÃO Z INICIAL INVÁLIDO
55	CONTADOR DE REDUÇÃO Z FINAL INVÁLIDO
56	ERRO DE ALOCAÇÃO
57	DADOS DO RTC INCORRETOS
58	DATA ANTERIOR AO ÚLTIMO DOCUMENTO EMITIDO
59	FORA DE INTERVENÇÃO TÉCNICA
60	EM INTERVENÇÃO TÉCNICA
61	ERRO NA MEMÓRIA DE TRABALHO
62	JÁ HOVE MOVIMENTO NO DIA
63	BLOQUEIO POR RZ
64	FORMA DE PAGAMENTO ABERTA
65	AGUARDANDO PRIMEIRO PROPRIETÁRIO
66	AGUARDANDO RZ
67	ECF OU LOJA IGUAL A ZERO
68	CUPOM ADICIONAL NÃO PERMITIDO
69	DESCONTO MAIOR QUE TOTAL VENDIDO EM ICMS
70	RECEBIMENTO NÃO FISCAL NULO NÃO PERMITIDO
71	ACRÉSCIMO OU DESCONTO MAIOR QUE TOTAL NÃO FISCAL
72	MEMÓRIA FISCAL LOTADA PARA NOVO CARTUCHO
73	ERRO DE GRAVAÇÃO NA MF
74	ERRO DE GRAVAÇÃO NA MFD
75	DADOS DO RTC ANTERIORES AO ÚLTIMO DOC ARMAZENADO
76	MEMÓRIA FISCAL SEM ESPAÇO PARA GRAVAR LEITURAS DA MFD
77	MEMÓRIA FISCAL SEM ESPAÇO PARA GRAVAR VERSAO DO SB
78	DESCRIÇÃO IGUAL A DEFAULT NÃO PERMITIDO
79	EXTRAPOLADO NÚMERO DE REPETIÇÕES PERMITIDAS
80	SEGUNDA VIA DO COMPROVANTE DE CRÉDITO OU DÉBITO NÃO PERMITIDO
81	PARCELAMENTO FORA DA SEQUÊNCIA
82	COMPROVANTE DE CRÉDITO OU DÉBITO ABERTO
83	TEXTO COM SEQUÊNCIA DE ESC INVÁLIDA
84	TEXTO COM SEQUÊNCIA DE ESC INCOMPLETA
85	VENDA COM VALOR NULO
86	ESTORNO DE VALOR NULO
87	FORMA DE PAGAMENTO DIFERENTE DO TOTAL DA SANGRIA
88	REDUÇÃO NÃO PERMITIDA EM INTERVENÇÃO TÉCNICA
89	AGUARDANDO RZ PARA ENTRADA EM INTERVENÇÃO TÉCNICA
90	FORMA DE PAGAMENTO COM VALOR NULO NÃO PERMITIDO
91	ACRÉSCIMO E DESCONTO MAIOR QUE VALOR DO ÍTEM
92	AUTENTICAÇÃO NÃO PERMITIDA
93	TIMEOUT NA VALIDAÇÃO
94	COMANDO NÃO EXECUTADO EM IMPRESSORA BILHETE DE PASSAGEM
95	COMANDO NÃO EXECUTADO EM IMPRESSORA DE CUPOM FISCAL
96	CUPOM NÃO FISCAL FECHADO
97	PARÂMETRO NÃO ASCII EM CAMPO ASCII

98	PARÂMETRO NÃO ASCII NUMÉRICO EM CAMPO ASCII NUMÉRICO
99	TIPO DE TRANSPORTE INVÁLIDO
100	DATA E HORA INVÁLIDA
101	SEM RELATÓRIO GERENCIAL OU COMPROVANTE DE CRÉDITO OU DÉBITO ABERTO
102	NÚMERO DO TOTALIZADOR NÃO FISCAL INVÁLIDO
103	PARÂMETRO DE ACRÉSCIMO OU DESCONTO INVÁLIDO
104	ACRÉSCIMO OU DESCONTO EM SANGRIA OU SUPRIMENTO NÃO PERMITIDO
105	NÚMERO DO RELATÓRIO GERENCIAL INVÁLIDO
106	FORMA DE PAGAMENTO ORIGEM NÃO PROGRAMADA
107	FORMA DE PAGAMENTO DESTINO NÃO PROGRAMADA
108	ESTORNO MAIOR QUE FORMA PAGAMENTO
109	CARACTER NUMÉRICO NA CODIFICAÇÃO GT NÃO PERMITIDO
110	ERRO NA INICIALIZAÇÃO DA MF
111	NOME DO TOTALIZADOR EM BRANCO NÃO PERMITIDO
112	DATA E HORA ANTERIORES AO ÚLTIMO DOC ARMAZENADO
113	PARÂMETRO DE ACRÉSCIMO OU DESCONTO INVÁLIDO
114	ÍTEM ANTERIOR AOS TREZENTOS ÚLTIMOS
115	ÍTEM NÃO EXISTE OU JÁ CANCELADO
116	CÓDIGO COM ESPAÇOS NÃO PERMITIDO
117	DESCRICAÇÃO SEM CARACTER ALFABÉTICO NÃO PERMITIDO
118	ACRÉSCIMO MAIOR QUE VALOR DO ÍTEM
119	DESCONTO MAIOR QUE VALOR DO ÍTEM
120	DESCONTO EM ISS NÃO PERMITIDO
121	ACRÉSCIMO EM ÍTEM JÁ EFETUADO
122	DESCONTO EM ÍTEM JÁ EFETUADO
123	ERRO NA MEMÓRIA FISCAL CHAMAR CREDENCIADO
124	AGUARDANDO GRAVAÇÃO NA MEMÓRIA FISCAL
125	CARACTER REPETIDO NA CODIFICAÇÃO DO GT
126	VERSÃO JÁ GRAVADA NA MEMÓRIA FISCAL
127	ESTOURO DE CAPACIDADE NO CHEQUE
128	TIMEOUT NA LEITURA DO CHEQUE
129	MÊS INVÁLIDO
130	COORDENADA INVÁLIDA
131	SOBREPOSIÇÃO DE TEXTO
132	SOBREPOSIÇÃO DE TEXTO NO VALOR
133	SOBREPOSIÇÃO DE TEXTO NO EXTENSO
134	SOBREPOSIÇÃO DE TEXTO NO FAVORECIDO
135	SOBREPOSIÇÃO DE TEXTO NA LOCALIDADE
136	SOBREPOSIÇÃO DE TEXTO NO OPCIONAL
137	SOBREPOSIÇÃO DE TEXTO NO DIA
138	SOBREPOSIÇÃO DE TEXTO NO MÊS
139	SOBREPOSIÇÃO DE TEXTO NO ANO
140	USANDO MFD DE OUTRO ECF
141	PRIMEIRO DADO DIFERENTE DE ESC OU 1C
142	NÃO PERMITIDO ALTERAR SEM INTERVENÇÃO TÉCNICA
143	DADOS DA ÚLTIMA RZ CORROMPIDOS
144	COMANDO NÃO PERMITIDO NO MODO INICIALIZAÇÃO
145	AGUARDANDO ACERTO DE RELÓGIO
146	MFD JÁ INICIALIZADA PARA OUTRA MF
147	AGUARDANDO ACERTO DO RELÓGIO OU DESBLOQUEIO PELO TECLADO
148	VALOR FORMA DE PAGAMENTO MAIOR QUE MÁXIMO PERMITIDO
149	RAZÃO SOCIAL EM BRANCO
150	NOME DE FANTASIA EM BRANCO
151	ENDEREÇO EM BRANCO
152	ESTORNO DE CDC NÃO PERMITIDO
153	DADOS DO PROPRIETÁRIO IGUAIS AO ATUAL
154	ESTORNO DE FORMA DE PAGAMENTO NÃO PERMITIDO
155	DESCRIÇÃO FORMA DE PAGAMENTO IGUAL JÁ PROGRAMADA
156	ACERTO DE HORÁRIO DE VERÃO SÓ IMEDIATAMENTE APÓS RZ
157	IT NÃO PERMITIDA MF RESERVADA PARA RZ
158	SENHA CNPJ INVÁLIDA
159	TIMEOUT NA INICIALIZAÇÃO DA NOVA MF
160	NÃO ENCONTRADO DADOS NA MFD
161	SANGRIA OU SUPRIMENTO DEVEM SER ÚNICOS NO CNF
162	ÍNDICE DA FORMA DE PAGAMENTO NULO NÃO PERMITIDO
163	UF DESTINO INVÁLIDA
164	TIPO DE TRANSPORTE INCOMPATÍVEL COM UF DESTINO

165	DESCRIÇÃO DO PRIMEIRO ÍTEM DO BILHETE DE PASSAGEM DIFERENTE DE "TARIFA"
166	AGUARDANDO IMPRESSÃO DE CHEQUE OU AUTENTICAÇÃO
167	NÃO PERMITIDO PROGRAMAÇÃO CNPJ IE COM ESPAÇOS EM BRANCO
168	NÃO PERMITIDO PROGRAMAÇÃO UF COM ESPAÇOS EM BRANCO
169	NÚMERO DE IMPRESSÕES DA FITA DETALHE NESTA INTERVENÇÃO TÉCNICA ESGOTADO
170	CF JÁ SUBTOTALIZADO
171	CUPOM NÃO SUBTOTALIZADO
172	ACRÉSCIMO EM SUBTOTAL JÁ EFETUADO
173	DESCONTO EM SUBTOTAL JÁ EFETUADO
174	ACRÉSCIMO NULO NÃO PERMITIDO
175	DESCONTO NULO NÃO PERMITIDO
176	CANCELAMENTO DE ACRÉSCIMO OU DESCONTO EM SUBTOTAL NÃO PERMITIDO
177	DATA INVÁLIDA
178	VALOR DO CHEQUE NULO NÃO PERMITIDO
179	VALOR DO CHEQUE INVÁLIDO
180	CHEQUE SEM LOCALIDADE NÃO PERMITIDO
181	CANCELAMENTO ACRÉSCIMO EM ÍTEM NÃO PERMITIDO
182	CANCELAMENTO DESCONTO EM ÍTEM NÃO PERMITIDO
183	NÚMERO MÁXIMO DE ÍTENS ATINGIDO
184	NÚMERO DE ÍTEM NULO NÃO PERMITIDO
185	MAIS QUE DUAS ALÍQUOTAS DIFERENTES NO BILHETE DE PASSAGEM NÃO PERMITIDO
186	ACRÉSCIMO OU DESCONTO EM ÍTEM NÃO PERMITIDO
187	CANCELAMENTO DE ACRÉSCIMO OU DESCONTO EM ÍTEM NÃO PERMITIDO
188	CLICHE JÁ IMPRESSO
189	TEXTO OPCIONAL DO CHEQUE EXCEDEU O MÁXIMO PERMITIDO
190	IMPRESSÃO AUTOMÁTICA NO VERSO NÃO PERMITIDO NESTE EQUIPAMENTO
191	TIMEOUT NA INSERÇÃO DO CHEQUE
192	OVERFLOW NA CAPACIDADE DE TEXTO DO COMPROVANTE DE CRÉDITO OU DÉBITO
193	PROGRAMAÇÃO DE ESPAÇOS ENTRE CUPONS MENOR QUE O MÍNIMO PERMITIDO
194	EQUIPAMENTO NÃO POSSUI LEITOR DE CHEQUE
195	PROGRAMAÇÃO DE ALÍQUOTA COM VALOR NULO NÃO PERMITIDO
196	PARÂMETRO BAUD RATE INVÁLIDO
197	CONFIGURAÇÃO PERMITIDA SOMENTE PELA PORTA DOS FISCO
198	VALOR TOTAL DO ÍTEM EXCEDE 11 DÍGITOS
199	PROGRAMAÇÃO DA MOEDA COM ESPAÇOS EM BRANCO NÃO PERMITIDO
200	CASAS DECIMAIS DEVEM SER PROGRAMADAS COM 2 OU 3
201	NÃO PERMITE CADASTRAR USUÁRIOS DIFERENTES NA MESMA MFD
202	IDENTIFICAÇÃO DO CONSUMIDOR NÃO PERMITIDA PARA SANGRIA OU SUPRIMENTO
203	CASAS DECIMAIS EM QUANTIDADE MAIOR DO QUE A PERMITIDA
204	CASAS DECIMAIS DO UNITÁRIO MAIOR DO QUE O PERMITIDA
205	POSIÇÃO RESERVADA PARA ICMS
206	POSIÇÃO RESERVADA PARA ISS
207	TODAS AS ALÍQUOTAS COM A MESMA VINCULAÇÃO NÃO PERMITIDO
208	DATA DE EMBARQUE ANTERIOR A DATA DE EMISSÃO
208	RETORNO_DATA_DE_EMBARQUE_ANTERIOR_A_DATA_DA_EMISSAO
209	RETORNO_ALIQUOTA_DE_ISS_NAO_PERMITIDO_SEM_INSCRICAO_MUNICIPAL
210	RETORNO_PACOTE_CLICHE_FORA_DA_SEQUENCIA
211	RETORNO_ESPACO_PARA_PARA_ARMAZENAMENTO_CLICHE_ESGOTADO
212	RETORNO_CLICHE_GRAFICO_NAO_DISPONIVEL_PARA_CONFIRMACAO
213	RETORNO_CRC_DO_CLICHE_GRAFICO_DIFERENTE_DO_INFORMADO

### 3.2.8.34 Elgin\_IniciaFechamentoCupomMFD

Inicia o fechamento do cupom fiscal. Permite acréscimo e desconto no fechamento do cupom.

#### Parâmetros:

**AcrescimoDesconto:** String que indica se haverá acréscimo no cupom, desconto ou ambos. "A" para acréscimo, "D" para desconto e "X" para acréscimo e desconto.

**TipoAcrescimoDesconto:** String que indica se o acréscimo ou desconto é por valor ou por percentual. "\$" para desconto por valor e "%" para percentual.

**ValorAcrescimo:** String com no máximo 14 dígitos para acréscimo por valor e 4 dígitos para acréscimo percentual.

**ValorDesconto:** String com no máximo 14 dígitos para desconto por valor e 4 dígitos para desconto percentual.

**Retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observação:**

- Para efetuar a forma de pagamento use a função [Elgin\\_EfetuaFormaPagamentoMFD](#).
- O valor "X" para o parâmetro *A acrescimoDesconto* não é suportado pelos ECFs FIT 1E e IF 6000TH

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_IniciaFechamentoCupomMFD("D", "%", "1200", "0000")
```

**// Exemplo em Delphi**

```
cAcresDesc := 'D';
cTipoAcresDesc := '%';
cValorAcres := '1200';
cValorDesc := '0000';
iResultado := Elgin_IniciaFechamentoCupomMFD(pchar( cAcresDesc), pchar( cTipoAcresDesc ), pchar(
cValorAcres ), pchar( cValorDesc ) );
```

**3.2.8.35 Elgin\_IniciaFechamentoRecebimentoNaoFiscalMFD**

Inicia o fechamento do recebimento não fiscal. Permite acréscimo e desconto no fechamento do recebimento.

**Parâmetros:**

***A acrescimoDesconto***: String que indica se haverá acréscimo no cupom, desconto ou ambos. "A" para acréscimo, "D" para desconto e "X" para acréscimo e desconto.

***TipoAcrescimoDesconto***: String que indica se o acréscimo ou desconto é por valor ou por percentual. "\$" para desconto por valor e "%" para percentual.

***ValorAcrescimo***: String com no máximo 14 dígitos para acréscimo por valor e 4 dígitos para acréscimo percentual.

***ValorDesconto***: String com no máximo 14 dígitos para desconto por valor e 4 dígitos para desconto percentual.

**Retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Observação:**

- Para efetuar a forma de pagamento use a função [Elgin\\_EfetuaFormaPagamentoMFD](#).
- O valor "X" para o parâmetro *A acrescimoDesconto* não é suportado pelos ECFs FIT 1E e IF 6000TH

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_IniciaFechamentoRecebimentoNaoFiscalMFD("D", "%", "1200", "0000")
```

**// Exemplo em Delphi**

```
cAcresDesc := 'D';
```

```
cTipoAcresDesc := '$';  
cValorAcres := '0200';  
cValorDesc := '0000';  
iResultado := Elgin_IniciaFechamentoRecebimentoNaoFiscalMFD(pchar ( cAcresDesc ), pchar(  
cTipoAcresDesc ), pchar( cValorAcres ), pchar( cValorDesc ) );
```

### 3.2.8.36 Elgin\_InicioFimCOOsMFD

Retorna o COO (Contador de Ordem de Operação) inicial e final.

#### Parâmetros:

**cCOOInicial:** Variável String para receber a informação do COO inicial.

**cCOOFinal:** Variável String para receber a informação do COO final.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Observação:

- Esta função deve ser usada após a redução Z para que o COO inicial e final tenha referência ao dia do movimento atual.

#### Exemplos

##### ' Exemplo em Visual Basic

```
cCOOInicial = space(6)
```

```
cCOOFinal = space(6)
```

```
iResultado = Elgin_InicioFimCOOsMFD(cCOOInicial, cCOOFinal)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 6 do
```

```
begin
```

```
cCOOInicial := cCOOInicial + ' ';
```

```
cCOOFinal := cCOOFinal + ' ';
```

```
end;
```

```
iResultado := Elgin_InicioFimCOOsMFD( cCOOInicial, cCOOFinal );
```

### 3.2.8.37 Elgin\_InicioFimGTsMFD

Retorna o valor do grande total (GT) inicial e final.

#### Parâmetros:

**cGTInicial:** Variável String para receber a informação do GT inicial.

**cGTFinal:** Variável String para receber a informação do GT final.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Observação:

- Esta função deve ser usada após a redução Z para que o GT inicial e final tenha referência ao dia do

movimento atual.

### Exemplos

#### ' Exemplo em Visual Basic

```
cGTInicial = space(18)
cGTFinal = space(18)
iResultado = Elgin_InicioFimGTsMFD(cGTInicial, cGTFinal)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 18 do
begin
cGTInicial := cGTInicial + ' ';
cGTFinal := cGTFinal + ' ';
end;
iResultado := Elgin_InicioFimGTsMFD( cGTInicial, cGTFinal );
```

### 3.2.8.38 Elgin\_InscricaoEstadualMFD

Retorna a inscrição estadual do cliente cadastrada na impressora.

#### Parâmetro:

**InscricaoEstadual:** Variável String com o tamanho de 20 posições para receber a informação.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu.  
-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.  
-5: Erro ao abrir a porta de comunicação.  
-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
sInscricaoEstadual = space(20)
iResultado = Elgin_InscricaoEstadualMFD(sInscricaoEstadual)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 20 do sInscricaoEstadual := sInscricaoEstadual + ' ';
iResultado := Elgin_InscricaoEstadualMFD( sInscricaoEstadual );
```

### 3.2.8.39 Elgin\_InscricaoMunicipalMFD

Retorna a inscrição municipal do cliente cadastrada na impressora.

#### Parâmetro:

**InscricaoMunicipal:** Variável String com o tamanho de 20 posições para receber a informação.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu.  
-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.  
-5: Erro ao abrir a porta de comunicação.  
-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
sInscricaoMunicipal = space(20)
iResultado = Elgin_InscricaoMunicipalMFD(sInscricaoMunicipal)
```

```
// Exemplo em Delphi
for iConta := 1 to 20 do sInscricaoMunicipal := sInscricaoMunicipal + ' ';
iResultado := Elgin_InscricaoMunicipalMFD( sInscricaoMunicipal );
```

### 3.2.8.40 Elgin\_MapaResumoMFD

Gera o relatório "Mapa Resumo" referente ao movimento do dia. As informações serão geradas no arquivo RETORNO.TXT no diretório configurado no parâmetro "path" do arquivo ini. O diretório default configurado é o diretório raiz (C:\).

#### Parâmetro:

Não há.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Observações:

- As informações contidas no mapa resumo podem variar de Estado para Estado. Essa função gera o relatório com as informações padrões que são usadas na maioria dos Estados. O relatório gerado terá o seguinte layout:

Contador de Redução Z	0312
COO	000026
Venda Bruta	43,73
Venda Líquida	25,50
Cancelamentos	11,14
Acréscimos	2,33
Descontos	7,09
ISS	0,00
Isenção	0,00
Não Incidência	1,95
Substituição Tributária	23,55
1200	0,00
1700	0,00

- Após a linha "Substituição Tributária" serão gravadas as informações de vendas referente as alíquotas de ICMS. Portanto, o número de linhas irá variar de acordo com o número de alíquotas de ICMS cadastradas na impressora.

- As informações contidas no mapa resumo serão referentes aos dados da última redução Z, portanto, para gerar o mapa resumo referente ao movimento do dia você deve executar essa função após a redução Z, caso contrário o mapa resumo será gerado com as informações referente ao movimento do dia anterior.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_MapaResumoMFD()
```

```
// Exemplo em Delphi
iResultado := Elgin_MapaResumoMFD();
```

### 3.2.8.41 Elgin\_MarcaModeloTipoImpressoraMFD

Retorna a marca, o modelo e o tipo da impressora.

#### Parâmetro:

**Marca:** Variável STRING com 15 posições para receber a marca da impressora.

**Modelo:** Variável STRING com 20 posições para receber o modelo.

**Tipo:** Variável STRING com 7 posições para receber o tipo da impressora.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
sMarca = space(15)
```

```
sModelo = space(20)
```

```
sTipo = space(7)
```

```
iResultado = Elgin_MarcaModeloTipoImpressoraMFD(sMarca,sModelo,sTipo)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 15 do sMarca := sMarca + ' ';
```

```
for iConta := 1 to 20 do sModelo := sModelo + ' ';
```

```
for iConta := 1 to 7 do sTipo := sTipo + ' ';
```

```
iResultado := Elgin_MarcaModeloTipoImpressoraMFD( sMarca, sModelo, sTipo );
```

### 3.2.8.42 Elgin\_MinutosEmitindoDocumentosFiscaisMFD

Retorna o tempo em que a impressora emitiu documentos fiscais.

#### Parâmetro:

**Minutos:** Variável String com o tamanho de 4 posições para receber a informação.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
sTempo = space(4)
```

```
iResultado = Elgin_MinutosEmitindoDocumentosFiscaisMFD(sTempo)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 4 do sTempo := sTempo + ' ';
```

```
iResultado := Elgin_MinutosEmitindoDocumentosFiscaisMFD( sTempo );
```

### 3.2.8.43 Elgin\_NomeiaRelatorioGerencialMFD

Programa Relatório Gerencial.

**Parâmetros:**

**Índice:** String numérica com valor entre 0 e 19 para o índice do relatório.

**Descricao:** String com até 17 caracteres com o nome do relatório.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- Só será possível nomear um relatório gerencial, caso a impressora ainda não tenha iniciado o seu movimento.

- Não é possível alterar ou apagar um relatório gerencial já gravado.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_NomeiaRelatorioGerencialMFD("2","Pagamento de carnê")
```

**// Exemplo em Delphi**

```
cIndice := '2';
```

```
cDescricao := 'Pagamento de carnê';
```

```
iResultado := Elgin_NomeiaRelatorioGerencialMFD(pchar ( cIndice ), pchar( cDescricao ) );
```

### 3.2.8.44 Elgin\_NumeroSerieMemoriaMFD

Retorna o número de série da memória de fita detalhe (MFD).

**Parâmetro:**

**NumeroSerieMFD:** Variável String com o tamanho de 20 posições para receber a informação.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
sMFD = space(20)
```

```
iResultado = Elgin_NumeroSerieMemoriaMFD(sMFD)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 20 do sMFD := sMFD + ' ';
```

```
iResultado := Elgin_NumeroSerieMemoriaMFD( sMFD );
```

### 3.2.8.45 Elgin\_PercentualLivreMFD

Retorna o percentual livre da Memória Fita Detalhe (MFD) no formato XX,XX% (com a vírgula e o %).

**Parâmetro:**

**MemoriaLivre:** Variável String com o tamanho de 6 posições para receber os dados.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
sMemoriaLivre = space(6)
```

```
iResultado = Elgin_PercentualLivreMFD(sMemoriaLivre)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 6 do sMemoriaLivre := sMemoriaLivre + ' ';
```

```
iResultado := Elgin_PercentualLivreMFD( sMemoriaLivre );
```

**3.2.8.46 Elgin\_ProgramaFormaPagamentoMFD**

Programa as formas de pagamento.

**Parâmetros:**

**FormaPagto:** STRING até 16 caracteres com a forma de pagamento.

**OperacaoTef:** STRING com 0 (zero) ou 1 (um) indicando se a forma de pagamento permite operação TEF ou não, onde:

1 - permite operação TEF

0 - não permite operação TEF.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Observações:**

- A forma de pagamento a ser programada é tratada como case insensitive se esta não existir. Caso a mesma já esteja cadastrada esta será sobreposta em sua posição atual.(Ex.: Cheque e CHEQUE são considerados a mesma forma de pagamento).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ProgramaFormaPagamentoMFD("Cartão", "1")
```

**// Exemplo em Delphi**

```
cFormaPgto := 'Cartao';
```

```
cTef := '1';
```

```
iResultado := Elgin_ProgramaFormaPagamentoMFD(pchar ( cFormaPgto ), pchar( cTef ) );
```

**3.2.8.47 Elgin\_ReducoesRestantesMFD**

Retorna o número de reduções restantes na impressora.

**Parâmetro:**

**Reduções:** Variável String com o tamanho de 4 posições para receber a informação.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
sRed = space(4)
```

```
iResultado = Elgin_ReducoesRestantesMFD(sRed)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 4 do sRed := sRed + ' ';
```

```
iResultado := Elgin_ReducoesRestantesMFD( sRed );
```

**3.2.8.48 Elgin\_RegistrosTipo60**

Esta função retorna os registro Tipo 60 Mestre e Analítico completos da impressora.

**Parâmetro:**

não há.

O retorno das informações geradas por esta função, serão gravadas no arquivo **RETORNO.TXT**, na seguinte ordem:

**- para o Registro Tipo 60 Mestre:**

Campo	Conteúdo	Tamanho
Tipo	60	2
Subtipo	M	1
Data de Emissão	Data de Emissão dos Cupons Fiscais	8
Número de Série do ECF	Número de Série do ECF	20
Número do ECF	Número do ECF	3
Modelo do Documento Fiscal	Código do Modelo Documento Fiscal	2
COO Inicial	Primeiro Cupom Fiscal Emitido	6
COO Final	Último Cupom Fiscal Emitido	6
Contador de Redução Z	Reduções Z	6
Contador de Reinício de Operação	Reinício de Operações	3
Venda Bruta	Valor da Venda Bruta	16
Grande Total	Valor do GT	16
Branco		37

**- para o Registro Tipo 60 Analítico:**

<b>Campo</b>	<b>Conteúdo</b>	<b>Tamanho</b>
Tipo	60	2
Subtipo	A	1
Data de Emissão	Data de Emissão dos Cupons Fiscais	8
Número de Série do ECF	Número de Série do ECF	20
Situação Tributária/Alíquota	Identificador da Situação Tributária/Alíquota do ICMS	4
Valor Acumulado no Totalizador Parcial	Valor acumulado no final do dia no totalizador parcial da situação tributária	12
Branços		79

**Observações:**

- Esta função deve ser utilizada diariamente, após a emissão da Redução Z.
- Os campos já estão formatados no tamanho que o relatório do Sintegra exige.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_RegistrosTipo60()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_RegistrosTipo60();
```

**3.2.8.49 Elgin\_ReimpressaoNaoFiscalVinculadoMFD**

Reimprime o comprovante não fiscal vinculado. Será executado, somente, se o comando for enviado imediatamente após a impressão do comprovante.

**Parâmetro:**

Não há

**Retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ReimpressaoNaoFiscalVinculadoMFD()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_ReimpressaoNaoFiscalVinculadoMFD();
```

### 3.2.8.50 Elgin\_RelatorioTipo60AnaliticoMFD

Gera o relatório "Tipo 60 analítico" exigido pelo convênio de ICMS 85/2001. As informações serão geradas no arquivo RETORNO.TXT no diretório configurado no parâmetro "path" do arquivo ini. O diretório default é o diretório raiz (C:\).

#### Parâmetro:

Não há.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Observações:

- O relatório gerado terá o seguinte layout:

Tipo do relatório	60
Subtipo	A
Data de emissão	12/05/2003
Número de série	570899040002
	8
Cancelamentos	11,14
Descontos	7,09
F	18,55
I	0,00
N	1,95
1200	0,00
1700	0,00
ISS	0,00

- Após a linha "N" serão gravadas as informações de vendas referentes às alíquotas tributárias. Portanto, o número de linhas irá variar de acordo com o número de alíquotas cadastradas na impressora.

- As informações contidas no relatório Tipo 60 Analítico são referentes aos dados da última Redução Z. Portanto, para gerar o relatório referente ao movimento do dia você deve executar essa função após a Redução Z, caso contrário as informações geradas serão referentes ao movimento do dia anterior.

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_RelatorioTipo60AnaliticoMFD()
```

##### // Exemplo em Delphi

```
iResultado := Elgin_RelatorioTipo60AnaliticoMFD();
```

### 3.2.8.51 Elgin\_SegundaViaNaoFiscalVinculadoMFD

Imprime a segunda via do comprovante não fiscal vinculado. Deve ser executada imediatamente após a emissão da primeira via.

**Parâmetro:**

Não há

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_SegundaViaNaoFiscalVinculadoMFD()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_SegundaViaNaoFiscalVinculadoMFD();
```

### 3.2.8.52 Elgin\_StatusEstendidoMFD

Retorna o status de relatório gerencial ou comprovante não-fiscal aberto.

**Parâmetro:**

*iStatus*: Variável INTEGER para receber a informação.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_StatusEstendidoMFD(iStatus)
```

**// Exemplo em Delphi**

```
iResultado := Elgin_StatusEstendidoMFD( iStatus );
```

**Análise do iStatus:**

...

```
if iStatus >= 128 then
```

```
  iStatus = iStatus - 128
```

```
endif
```

```
if iStatus >= 64 then
```

```
  iStatus = iStatus - 64
```

```
  cMensagem = cMensagem + "Estorno de Comprovante de Débito ou Crédito
```

```
  permitido" + chr(13) + chr(10)
```

```
endif
```

```
if iStatus >= 32 then
```

```
  iStatus = iStatus - 32
```

```
  cMensagem = cMensagem + "Permite cancelamento do CNF" + chr(13) + chr(10)
```

```
endif
```

```

if iStatus >= 16 then
iStatus = iStatus - 16
endif

if iStatus >= 8 then
iStatus = iStatus - 8
cMensagem = cMensagem + "Totalizando Cupom" + chr(13) + chr(10)
endif

if iStatus >= 4 then
iStatus = iStatus - 4
cMensagem = cMensagem + "Relatório Gerencial Aberto" + chr(13) + chr(10)
endif

if iStatus >= 2 then
iStatus = iStatus - 2
cMensagem = cMensagem + "Comprovante de Débito ou Crédito
Aberto" + chr(13) + chr(10)
endif

if iStatus >= 1 then
iStatus = iStatus - 1
cMensagem = cMensagem + "Comprovante Não-Fiscal Aberto" + chr(13) + chr(10)
endif
...

```

### 3.2.8.53 Elgin\_SubTotalComprovanteNaoFiscalMFD

Retorna o subtotal do comprovante não fiscal não vinculado (recebimento).

#### Parâmetro:

**cSubtotal:** Variável String para receber a informação.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
cSubtotal = space(14)
```

```
iResultado = Elgin_SubTotalComprovanteNaoFiscalMFD(cSubtotal)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 14 do cSubtotal := cSubtotal + ' ';
```

```
iResultado := Elgin_SubTotalComprovanteNaoFiscalMFD( cSubtotal );
```

### 3.2.8.54 Elgin\_TamanhoTotalMFD

Retorna o tamanho total da MFD em bytes.

#### Parâmetro:

**MemoriaLivre:** Variável String com o tamanho de 10 posições para receber os dados.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.  
-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.  
-5: Erro ao abrir a porta de comunicação.  
-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
sMemoriaLivre = space(10)
iResultado = Elgin_TamanhoTotalMFD(sMemoriaLivre)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 10 do sMemoriaLivre := sMemoriaLivre + ' ';
iResultado := Elgin_TamanhoTotalMFD( sMemoriaLivre );
```

### 3.2.8.55 Elgin\_TempoOperacionalMFD

Retorna o tempo em que a impressora está operacional.

#### Parâmetro:

**TempoOperacional:** Variável String com o tamanho de 4 posições para receber a informação.

#### Valores de Retorno:

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu.  
-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.  
-5: Erro ao abrir a porta de comunicação.  
-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
sTempo = space(4)
iResultado = Elgin_TempoOperacionalMFD(sTempo)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 4 do sTempo := sTempo + ' ';
iResultado := Elgin_TempoOperacionalMFD( iTempo );
```

### 3.2.8.56 Elgin\_TotalLivreMFD

Retorna o quantidade de bytes livres na MFD.

#### Parâmetro:

**MemoriaLivre:** Variável String com o tamanho de 10 posições para receber os dados.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu.  
-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.  
-5: Erro ao abrir a porta de comunicação.  
-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.  
-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
sMemoriaLivre = space(10)
iResultado = Elgin_TotalLivreMFD(sMemoriaLivre)
```

```
// Exemplo em Delphi
for iConta := 1 to 10 do sMemoriaLivre := sMemoriaLivre + ' ';
iResultado := Elgin_TotalLivreMFD( sMemoriaLivre );
```

### 3.2.8.57 Elgin\_UsaRelatorioGerencialMFD

Imprime as informações do Relatório Gerencial.

#### Parâmetro:

**Texto:** String Texto a ser impresso no relatório com até 492 caracteres.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu.  
-2: Parâmetro inválido.  
-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.  
-5: Erro ao abrir a porta de comunicação.  
-50: Número de série inválido.

#### Observações:

- O relatório permanecerá aberto por 2 minutos, se não for enviado o comando de fechamento durante esse período, o relatório será fechado automaticamente.  
- Para abrir o relatório gerencial use a função [Elgin\\_AbreRelatorioGerencialMFD](#).  
- Para fechar o relatório use a função [Elgin\\_FechaRelatorioGerencial](#).

#### Exemplos

##### ' Exemplo em Visual Basic

```
iResultado = Elgin_UsaRelatorioGerencialMFD("Entre com o texto aqui !!!")
```

##### // Exemplo em Delphi

```
cMsg := 'Entre com o texto aqui !!!';
iResultado := Elgin_UsaRelatorioGerencialMFD(pchar ( cMsg ) );
```

### 3.2.8.58 Elgin\_ValorFormaPagamentoMFD

Retorna o valor acumulado em uma determinada forma de pagamento.

#### Parâmetros:

**FormaPagamento:** Variável String com até 16 posições com a descrição ou posição da Forma de Pagamento que deseja retornar o seu valor.

**Valor:** Variável String com 14 posições para receber a informação.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu.  
-1: Erro de execução da função.  
-2: Parâmetro inválido na função.  
-24: Forma de pagamento não programada.  
-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
Valor = Space(14)
iResultado = Elgin_ValorFormaPagamentoMFD("Cheque", Valor)
```

##### // Exemplo em Delphi

```
cFormaPgto := 'Cheque';
for iConta := 1 to 14 do Valor := Valor + ' ';
```

iResultado := Elgin\_ValorFormaPagamentoMFD(pchar ( cFormaPgto ), Valor );

### 3.2.8.59 Elgin\_ValorTotalizadorNaoFiscalMFD

Retorna o valor acumulado em um determinado totalizador não fiscal.

#### Parâmetros:

**Totalizador:** Variável String com até 19 posições com a descrição do Totalizador.

**Valor:** Variável String com 14 posições para receber a informação.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-2: Parâmetro inválido na função.

-25: Totalizador não fiscal não programado.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

Valor = Space(14)

iResultado = Elgin\_ValorTotalizadorNaoFiscalMFD("Rec. Prestacao", Valor)

##### // Exemplo em Delphi

cTotalizador := 'Cheque';

for iConta := 1 to 14 do Valor := Valor + ' ';

iResultado := Elgin\_ValorTotalizadorNaoFiscalMFD(pchar ( cTotalizador ), Valor );

### 3.2.8.60 Elgin\_VerificaEstadoImpressoraMFD

Retorna o estado da impressora.

#### Parâmetros:

**ACK:** Variável inteira para receber o primeiro byte.

**ST1:** Variável inteira para receber o segundo byte.

**ST2:** Variável inteira para receber o terceiro byte.

**ST3:** Variável inteira para receber o quarto byte (caso seja habilitado o **retorno estendido**).

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução da função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Observação:

- Essa função devolve o status da impressora. Maiores detalhes do retorno podem ser encontrados em:

[Elgin\\_HabilitaDesabilitaRetornoEstendidoMFD](#) e [Retorno Impressora](#).

#### Exemplos

##### ' Exemplo em Visual Basic

iResultado = Elgin\_VerificaEstadoImpressoraMFD( iAck, iSt1, iSt2, St3 )

##### // Exemplo em Delphi

iResultado := Elgin\_VerificaEstadoImpressoraMFD( iAck, iSt1, iSt2, St3 );



- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

#### Observações:

- Essa função retorna as informações para os 30 totalizadores não fiscais não vinculados na seguinte ordem:

Descrição do recebimento	19 bytes
Valor acumulado para cada recebimento (duas casas decimais)	14 bytes

#### Exemplo:

```
Conta de água 00000000005460,
Conta de Luz 00000000145850,
Conta de Telefone 00000000078437,
.
.
.
Sangria 00000000000000,
Suprimento 00000000010000,
```

- A impressora permite programar até 15 totalizadores não fiscais.

#### Exemplos

##### ' Exemplo em Visual Basic

```
sRec = space(1019)
iResultado = Elgin_VerificaRecebimentoNaoFiscalMFD(sRec)
```

##### // Exemplo em Delphi

```
for iConta := 1 to 1019 do sRec := sRec + ' ';
iResultado := Elgin_VerificaRecebimentoNaoFiscalMFD( sRec );
```

### 3.2.8.63 Elgin\_VerificaRelatorioGerencialMFD

Retorna os relatórios gerenciais programados e seus valores acumulados.

#### Parâmetro:

**Relatorios:** Variável String com 659 posições para receber as informações.

#### Retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

#### Observações:

- Essa função retorna as informações para os 30 relatórios gerenciais na seguinte ordem:

Número de vezes que o relatório foi utilizado	4 bytes
Descrição do relatório	17 bytes

Exemplo:

```
0005Relatorio Geral ,
0002Relatorio 3 ,
0001Relatorio 4 ,
.
.
.
0004Relatorio 30 ,
```

### Exemplos

#### ' Exemplo em Visual Basic

```
sRel = space(659)
iResultado = Elgin_VerificaRelatorioGerencialMFD(sRec)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 659 do sRel := sRel + ' ';
iResultado := Elgin_VerificaRelatorioGerencialMFD( sRec );
```

### 3.2.8.64 Elgin\_VerificaSensorPoucoPapelMFD

Verificar se o sensor de pouco papel está habilitado ou desabilitado.

#### Parâmetro:

**cFlag:** Variável STRING para receber a informacao, onde:

- 0: sensor desabilitado.
- 1: sensor habilitado.

#### Retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
cFlag = " "
iResultado = Elgin_VerificaSensorPoucoPapelMFD( cFlag )
```

#### // Exemplo em Delphi

```
cFlag := ' ';
iResultado := Elgin_VerificaSensorPoucoPapelMFD( cFlag );
```

### 3.2.8.65 Elgin\_VerificaTotalizadoresNaoFiscaisMFD

Retorna a descrição dos totalizadores não fiscais programados na impressora.

#### Parâmetro:

**Totalizadores:** Variável String com 599 posições para receber a descrição dos totalizadores não fiscais programados.

#### Retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 1 - Erro de execução.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.

- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

**Observações:**

- A impressora MFD permite programar até 28 totalizadores não fiscais e possui dois (2) pré-programados: Sangria e Suprimento.
- Os totalizadores virão separados por vírgula.

**Exemplos****' Exemplo em Visual Basic**

```
sTot = space(599)
iResultado = Elgin_VerificaTotalizadoresNaoFiscaisMFD(sTot)
```

**// Exemplo em Delphi**

```
for iConta := 1 to 599 do sTot := sTot + ' ';
iResultado := Elgin_VerificaTotalizadoresNaoFiscaisMFD( sTot );
```

**3.2.8.66 Elgin\_VerificaTotalizadoresParciaisMFD**

Retorna os totalizadores parciais da impressora.

**Parâmetro:**

**Totalizadores:** Variável String com o tamanho de 889 posições para receber os totalizadores parciais.

**Retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 1 - Erro de execução.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT
- 50: Número de série inválido.

**Observações:**

- São retornadas as seguintes informações separadas por vírgulas:

Totalizadores parciais tributados	224 bytes
Isenção de ICMS	14 bytes
Não incidência de ICMS	14 bytes
Substituição de ICMS	14 bytes
Isenção de ISSQN	14 bytes
Não incidência de ISSQN	14 bytes
Substituição de ISSQN	14 bytes
Descontos sobre ICMS	14 bytes
Acréscimos sobre ICMS	14 bytes
Cancelamentos sobre ICMS	14 bytes
Descontos sobre ISSQN	14 bytes
Acréscimos sobre ISSQN	14 bytes
Cancelamentos sobre ISSQN	14 bytes
Totalizadores não fiscais + sangria e suprimento	210 bytes
Descontos sobre não fiscais	14 bytes
Acréscimos sobre não fiscais	14 bytes
Cancelamentos sobre não fiscais	14 bytes
Grande Total	18 bytes

### Exemplos

#### ' Exemplo em Visual Basic

```
sTot = space(889)
iResultado = Elgin_VerificaTotalizadoresParciaisMFD(sTot)
```

#### // Exemplo em Delphi

```
for iConta := 1 to 889 do sTot := sTot + ' ';
iResultado := Elgin_VerificaTotalizadoresParciaisMFD( sTot );
```

### 3.2.9 Autenticação

#### 3.2.9.1 Elgin\_Autenticacao

Permite que seja feita a autenticação de documentos.

#### Parâmetro:

Não há.

#### Observações:

- Deverá ser executada imediatamente após um recebimento não fiscal ou o término de um cupom fiscal.
- Poderá ser repetido até 5 vezes para cada recebimento, após isso o comando será ignorado.
- A autenticação documento a data, número do caixa e o número do último cupom fiscal.

#### Retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

```
' Exemplo em Visual Basic  
iResultado = Elgin_Autenticacao( )
```

```
// Exemplo em Delphi  
iResultado := Elgin_Autenticacao( );
```

### 3.2.9.2 Elgin\_AutenticacaoMFD

Permite que seja feita a autenticação de documentos.

#### Parâmetro:

**Linhas:** STRING numérica com valor entre 0 e 49 representando o número de linhas que serão saltadas para imprimir o texto.

**Texto:** STRING com até 50 caracteres com o texto a ser impresso.

#### Observações:

- **Só poderá ser utilizado 50 caracteres no campo texto se não houver nenhum salto de linhas, ou seja, cada salto de linha corresponde a 1 caracter do parâmetro texto. Ex.: Para se saltar 5 linhas, o campo texto só poderá ter no máximo 45 caracteres, conforme o exemplo abaixo.**

- Deverá ser executada imediatamente após um recebimento não fiscal ou o término de um cupom fiscal.

- Poderá ser repetido até 5 vezes para cada recebimento, após isso o comando será ignorado.

- A autenticação conterá a data, número do caixa e o número do último cupom fiscal.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

```
' Exemplo em Visual Basic  
iResultado = Elgin_AutenticacaoMFD("5" , "0123456789012345678901234567890123456789012345678901234")
```

```
// Exemplo em Delphi  
iResultado := Elgin_AutenticacaoMFD('5' , '0123456789012345678901234567890123456789012345678901234' );
```

### 3.2.10 Funções de Código de Barras

#### 3.2.10.1 Configura Código de Barras

Esta função configura os seguintes itens do código de barras: Altura, Largura e Posição dos caracteres.

#### Parâmetros:

**Altura:** Inteiro entre 1 à 255. (default 162)

**Largura:** Inteiro entre 0 à 2, onde:

Largura = 0 (barras finas)

Largura = 1 (barras médias) - default

Largura = 2 (barras grossas)

**Posição dos Caracteres:** Inteiro entre 0 à 3, onde:

Posição = 0 (não imprime os caracteres do código)

Posição = 1 (imprime os caracteres acima do código)

Posição = 2 (imprime os caracteres abaixo do código) - default

Posição = 3 (imprime os caracteres acima e abaixo do código)

**Fonte:** Inteiro entre 0 à 1, onde:

Fonte = 0 (normal)

Fonte = 1 (condensado)

**Margem:** Inteiro entre 0 à 575 (dots pitch), onde

Margem = 0 (sem margem) default

**Observação:**

- A configuração do código de barras deve ser realizada dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
iRetorno01 = 150
```

```
iRetorno02 = 0
```

```
iRetorno03 = 2
```

```
iRetorno04 = 0
```

```
iRetorno05 = 250
```

```
iResultado = Elgin_ConfiguraCodigoBarrasMFD( iRetorno01, iRetorno02, iRetorno03, iRetorno04, iRetorno05 )
```

**// Exemplo em Delphi**

```
iRetorno01 := 150;
```

```
iRetorno02 := 0;
```

```
iRetorno03 := 2;
```

```
iRetorno04 := 0;
```

```
iRetorno05 := 250;
```

```
iResultado := Elgin_ConfiguraCodigoBarrasMFD( iRetorno01, iRetorno02, iRetorno03, iRetorno04, iRetorno05 );
```

**3.2.10.2 Elgin\_CodigoBarrasCODABARMFD**

Esta função faz a impressão do código de barras CODABAR.

**Parâmetro:**

**Código:** STRING do código que será gerado.

O tamanho da string é dada pela tabela abaixo.

Largura das Barras	Quantidade de Caracteres
0	20
1	12
2	8

A Largura das Barras é 1 (default).

**Observações:**

- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do

Comprovante Não Fiscal Vinculado.

- Será acrescentado, automaticamente, o dígito verificador.
- Aceita dígitos entre 0 à 9.
- Aceita as letras A, B, C e D (maiúsculas ou minúsculas).
- Aceita os caracteres: "\$", "+", "-", ".", "/" e ":".

#### **Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### **Exemplos**

##### **' Exemplo em Visual Basic**

```
iResultado = Elgin_CodigoBarrasCODABARMFD("012-345-ABC")
```

##### **// Exemplo em Delphi**

```
iResultado := Elgin_CodigoBarrasCODABARMFD('012-345-ABC');
```

#### **3.2.10.3 Elgin\_CodigoBarrasCODE39MFD**

Esta função faz a impressão do código de barras CODE39.

#### **Parâmetro:**

**Código:** STRING do código que será gerado.

O tamanho da string é dada pela tabela abaixo.

<b>Largura das Barras</b>	<b>Quantidade de Caracteres</b>
0	15
1	9
2	6

A Largura das Barras é 1 (default).

#### **Observações:**

- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.
- Será acrescentado, automaticamente, o dígito verificador.
- Aceita dígitos entre 0 à 9.
- Aceita as letras A, B, C e D (maiúsculas ou minúsculas).
- Aceita os caracteres: "\$", "+", "-", ".", "/" e ":".

#### **Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_CodigoBarrasCODE39MFD("ABC+123")
```

**// Exemplo em Delphi**

```
iResultado := Elgin_CodigoBarrasCODABARMFD('ABC=123');
```

**3.2.10.4 Elgin\_CodigoBarrasEAN13MFD**

Esta função faz a impressão do código de barras EAN13.

**Parâmetro:**

**Codigo:** STRING com o tamanho de 12 dígitos de 0 à 9

**Observações:**

- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.
- Será acrescentado, automaticamente, o dígito verificador.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_CodigoBarrasEAN13MFD("012345678901")
```

**// Exemplo em Delphi**

```
iResultado := Elgin_CodigoBarrasEAN13MFD('012345678901');
```

**3.2.10.5 Elgin\_CodigoBarrasEAN8MFD**

Esta função faz a impressão do código de barras EAN8.

**Parâmetro:**

**Codigo:** STRING com o tamanho de 7 dígitos de 0 à 9.

**Observações:**

- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.
- Será acrescentado, automaticamente, o dígito verificador.

**Retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos**

```
' Exemplo em Visual Basic
iResultado = Elgin_CodigoBarrasEAN8MFD("0123456")
```

```
// Exemplo em Delphi
iResultado := Elgin_CodigoBarrasEAN8MFD('0123456');
```

### 3.2.10.6 Elgin\_CodigoBarrasITFMFD

Esta função faz a impressão do código de barras ITF (Intercalado 2/5).

#### Parâmetro:

**Código:** STRING do código que será gerado.

O tamanho da string é dada pela tabela abaixo.

Largura das Barras	Quantidade de Caracteres
0	30
1	20
2	14

A Largura das Barras é 1 (default).

#### Observações:

- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.
- Aceita dígitos entre 0 à 9.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-1: Erro de execução.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Exemplos

```
' Exemplo em Visual Basic
iResultado = Elgin_CodigoBarrasITFMFD("0123456789012345")
```

```
// Exemplo em Delphi
iResultado := Elgin_CodigoBarrasITFMFD('0123456789012345');
```

### 3.2.10.7 Elgin\_CodigoBarrasUPCAMFD

Esta função faz a impressão do código de barras UPCA.

#### Parâmetro:

**Código:** STRING com o tamanho de 11 dígitos de 0 à 9

#### Observações:

- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.
- Será acrescentado, automaticamente, o dígito verificador.

#### Retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 1: Erro de execução.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_CodigoBarrasUPCAMFD("01234567890")
```

#### // Exemplo em Delphi

```
iResultado := Elgin_CodigoBarrasUPCAMFD('01234567890');
```

### 3.2.10.8 Elgin\_CodigoBarrasUPCEMFD

Esta função faz a impressão do código de barras UPCE.

#### Parâmetro:

**Codigo:** STRING com o tamanho de 6 dígitos de 0 à 9

#### Observações:

- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.
- Será acrescentado, automaticamente, o dígito verificador.

#### Retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 1: Erro de execução.
- 4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_CodigoBarrasUPCAMFD("012345")
```

#### // Exemplo em Delphi

```
iResultado := Elgin_CodigoBarrasUPCAMFD('012345');
```

### 3.2.11 Outras Funções

#### 3.2.11.1 Elgin\_AberturaDoDia

Faz a abertura do caixa emitindo um suprimento e uma leitura X.

#### Parâmetros:

**Valor:** String com o Valor do suprimento com até 14 dígitos (2 casas decimais). Só imprime cupom de Suprimento se o Valor for diferente de 0,00.

**FormaPagamento:** String com a Forma de pagamento com até 16 caracteres. Se não for informado, o suprimento será feito em Dinheiro.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_AberturaDoDia("50,00", "Dinheiro")
```

#### // Exemplo em Delphi

```
cValor := '50,00';  
cFormaPgto := 'Dinheiro';  
iResultado := Elgin_AberturaDoDia(pchar ( cValor ), pchar( cFormaPgto ) );
```

### 3.2.11.2 Elgin\_AbrePortaSerial

Abre a porta serial para comunicação entre a impressora e o micro.

#### Parâmetro:

Não há.

#### Possíveis retornos da Função (INTEIRO):

- 1: Indica que nenhum erro ocorreu
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 50: Número de série inválido.

#### Observação:

- A função lê o nome da porta a ser aberta no arquivo [Elgin.ini](#). Se o parâmetro "porta" estiver configurado com a palavra "Default" a função localiza onde a impressora está conectada e configura o arquivo INI.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_AbrePortaSerial()
```

#### // Exemplo em Delphi

```
iResultado := Elgin_AbrePortaSerial();
```

### 3.2.11.3 Elgin\_AcionaGuilhotinaMFD

Aciona a guilhotina da impressora.

#### Parâmetros:

*TipoCorte*: variável INTEIRA para definir o tipo de corte do papel. Podendo ser 0 ou 1.

#### Possíveis retornos da Função (INTEIRO):

- 0 (zero): Erro de Comunicação.
- 1 (um): OK.
- 2 (menos dois): Parâmetro Inválido.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 10: Impressora offline ou desligada
- 50: Número de série inválido.

### Exemplos

**' Exemplo em Visual Basic**

Retorno = Elgin\_AcionaGuilhotinaMFD(1)

**// Exemplo em Delphi**

Retorno := Elgin\_AcionaGuilhotinaMFD(1);

**3.2.11.4 Elgin\_FechamentoDoDia**

Faz o fechamento do dia emitindo uma Redução Z. Essa função grava o COO final e o Grande Total final.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

iResultado = Elgin\_FechamentoDoDia()

**// Exemplo em Delphi**

iResultado := Elgin\_FechamentoDoDia();

**3.2.11.5 Elgin\_FechaPortaSerial**

Fecha a porta serial.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

1: Indica que nenhum erro ocorreu.

0: Erro ao fechar a porta.

**Exemplos****' Exemplo em Visual Basic**

iResultado = Elgin\_FechaPortaSerial()

**// Exemplo em Delphi**

iResultado := Elgin\_FechaPortaSerial();

**3.2.11.6 Elgin\_IdentificaConsumidor**

Grava as informações do consumidor em um arquivo de texto e imprime os dados do consumidor no fechamento do Cupom.

**Parâmetro:**

**CNPJ\_CPF:** STRING com o CNPJ ou CPF do Consumidor.

**Nome:** STRING com o NOME do Consumidor.

**Endereco:** STRING com o ENDEREÇO do Consumidor.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 6: Impressora desligada ou cabo de comunicação desconectado.
- 50: Número de série inválido.

**Observação:**

- Esse comando deve ser utilizado após a abertura do cupom.
- Só pode ser usado quando não são passados os dados do consumidor na abertura do cupom.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_IdentificaConsumidor("74.199.852/0001-58", "Vinicyus Leme", "Rua São João, 45")
```

**// Exemplo em Delphi**

```
iResultado = Elgin_IdentificaConsumidor('74.199.852/0001-58', 'Vinicyus Leme', 'Rua São João, 45');
```

**3.2.11.7 Elgin\_ImprimeConfiguracoesImpressora**

Imprime as configurações da impressora fiscal em um relatório gerencial. Será emitida uma leitura X antes. Veja abaixo em "Observações" as informações que serão impressas.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 2: Parâmetro inválido na função.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

**Observações:**

Serão impressas as seguintes configurações:

Data da impressora	27/06/2001
Hora da impressora	11:38:37
Versão do firmware	0310
Número de série	4708991023070
Número da loja	0001
Número do caixa	0001
Símbolo da moeda	R\$
Modo de operação	Arredondamento
Horário de verão	Não
Já houve redução Z	Não
Memória fiscal	Com espaço
Versão da dll	1.9
log da dll	Desabilitado
Lin. Imp. após pouco papel	0
Tipo Impressora	fiscal + gaveta+autenticação

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ImprimeConfiguracoesImpressora()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_ImprimeConfiguracoesImpressora();
```

**3.2.11.8 Elgin\_ImprimeDepartamentos**

Imprime os departamentos e seus valores acumulados em um relatório gerencial.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-2: Parâmetro inválido na função.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ImprimeDepartamentos()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_ImprimeDepartamentos();
```

**3.2.11.9 Elgin\_IncluiCidadeFavorecido**

Permite incluir o nome da cidade e do favorecido no arquivo Elgin.ini.

**Parâmetros:**

**Cidade:** STRING com o Nome da cidade com até 27 caracteres.

**Favorecido:** STRING com o Nome do favorecido com até 45 caracteres.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Observações:**

- Esta função pode ser executada em qualquer momento;

- Após gravados a Cidade e Favorecido no arquivo Elgin.ini ao executar a função

[Elgin\\_ImprimeCheque](#) com o valor da cidade e favorecido em branco serão passados os valores que foram gravados no Elgin.ini.

**Exemplo:****Exemplo em Visual Basic**

```
iRetorno = Elgin_IncluiCidadeFavorecido("Manaus", "Mirian")
```

**Exemplo em Delphi**

```
sCidade := 'Manaus';  
sFavorecido := 'Mirian';  
iRetorno := Elgin_IncluiCidadeFavorecido(sCidade,sFavorecido);
```

**3.2.11.10 Elgin\_LeArquivoRetorno**

Le o conteúdo do arquivo RETORNO.TXT, com a informação solicitada da impressora, e grava na variável de retorno.

**Parâmetro:**

**sRetorno:** Variável do tipo STRING, com o mesmo tamanho da variável usada pela função anterior (função que solicitou a informação).

**Possíveis retornos da Função (INTEIRO):**

0: Erro de execução da função.  
1: OK.

**Observações:**

- Esta função só deverá ser usada, caso esteja trabalhando com "Cliente/Servidor" (Windows Terminal Server ou Frame-Relay).
- Esta função deve ser usada logo após a função que originou o retorno da informação da impressora.

**' Exemplo em Visual Basic**

```
strRetorno01 = Space(6)  
strRetorno02 = Space(1024)  
strRetorno03 = Space(6)  
iResultado = Elgin_NumeroCupom( strRetorno01 )  
iResultado = Elgin_RetornoImpressora(iRetorno01, strRetorno02)  
iResultado = Elgin_LeArquivoRetorno(strRetorno03)
```

**// Exemplo em Delphi**

```
strRetorno01 := StringofChar(' ', 6);  
strRetorno02 := StringofChar(' ', 1024);  
strRetorno03 := StringofChar(' ', 6);  
iResultado := Elgin_NumeroCupom( strRetorno01 );  
iResultado := Elgin_RetornoImpressora(iRetorno01, strRetorno02);  
iResultado := Elgin_LeArquivoRetorno(strRetorno03);
```

**3.2.11.11 Elgin\_LeBalanca**

Le a informação contida na balança.

**Parâmetro:**

**Porta:** Variável do tipo STRING, com o nome da porta serial a qual está conectada a balança.

**Baud:** Variável do tipo STRING, com a velocidade de conexão com a porta serial.

**Valor:** Variável do tipo STRING que retorna o valor contido na balança.

**Possíveis retornos da Função (INTEIRO):**

0: Erro de execução da função.  
1: OK.

**Observações:**

- Balanças Elgin (DIGIPESO).

**'Exemplo em Visual Basic**

```
Dim strPorta as String  
Dim strBaud as String
```

```
Dim strValor as String
Dim iRetorno as Integer

strValor = Space(5)
strPorta = "COM2"
strBaud = "9600"
iRetorno = Elgin_LeBalanca( strPorta, strBaud, strValor )
```

#### // Exemplo em Delphi

```
var
  strPorta, strBaud, strValor: String;
  iRetorno: Integer;
begin
  strValor := StringofChar(' ', 5);
  strPorta := 'COM2';
  strBaud := '9600';
  iRetorno := Elgin_LeBalanca( strPorta, strBaud, strValor );
end;
```

#### 3.2.11.12 Elgin\_MapasResumo

Gera o relatório "Mapa Resumo" referente ao movimento do dia. As informações serão geradas no arquivo RETORNO.TXT no diretório configurado no parâmetro "path" do arquivo ini. O diretório default configurado é o diretório raiz (C:\).

#### Parâmetro:

Não há.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.
- 50: Número de série inválido.

#### Observações:

- As informações contidas no mapa resumo podem variar de Estado para Estado. Essa função gera o relatório com as informações padrões que são usadas na maioria dos Estados. O relatório gerado terá o seguinte layout:

Contador de Redução Z	0312
COO	000026
Venda Bruta	43,73
Venda Líquida	25,50
Cancelamentos	11,14
Acréscimos	2,33
Descontos	7,09
ISS	0,00
Isenção	0,00
Não Incidência	1,95
Substituição Tributária	23,55
1200	0,00
1700	0,00

- Após a linha "Substituição Tributária" serão gravadas as informações de vendas referente as alíquotas de ICMS. Portanto, o número de linhas irá variar de acordo com o número de alíquotas de ICMS cadastradas na impressora.

- As informações contidas no mapa resumo serão referentes aos dados da última redução Z, portanto, para gerar o mapa resumo referente ao movimento do dia você deve executar essa função após a redução Z, caso contrário o mapa resumo será gerado com as informações referente ao movimento do dia anterior.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_MapaResumo()
```

#### // Exemplo em Delphi

```
iResultado := Elgin_MapaResumo();
```

### 3.2.11.13 Elgin\_ModeloImpressora

Retorna o MODELO da impressoracadastrado na impressora.

#### Parâmetro:

**MODELO:** Variável String para receber o MODELO da impressora.

#### Retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: Arquivo ini não encontrado ou parâmetro inválido para o nome da porta.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_ModeloImpressora(sMODELO)
```

#### // Exemplo em Delphi

```
iResultado := Elgin_ModeloImpressora( sMODELO );
```

### 3.2.11.14 Elgin\_NumeroSerieCriptografado

Criptografa o número de serie do ECF(Emissor de Cupom Fiscal).

**Parâmetro:**

**NumeroSerie:** variável STRING para receber o Número de Série do ECF criptografado.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_NumeroSerieCriptografado(sNumeroSerie)
```

**// Exemplo em Delphi**

```
iResultado = Elgin_NumeroSerieCriptografado(sNumeroSerie);
```

### 3.2.11.15 Elgin\_NumeroSerieDescriptografado

Descriptografa o número de serie criptografado do ECF(Emissor de Cupom Fiscal).

**Parâmetro:**

**NumeroSerieCriptografado:** variável STRING com o número de série da impressora criptografado.

**NumeroSerieDescriptografado:** variável STRING para receber o Número de Série descriptografado da impressora.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Observação:**

- O Numero de Serie deve ter sido criptografado na função [Elgin\\_NumeroSerieCriptografado](#).

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Elgin_NumeroSerieCriptografado(sNumeroSerie)
```

**// Exemplo em Delphi**

```
iResultado = Elgin_NumeroSerieCriptografado(sNumeroSerie);
```

### 3.2.11.16 Elgin\_ProgramaCaracterAutenticacao

Programa um caracter gráfico para autenticação.

**Parâmetro:**

**coordenadas:** STRING com os 18 valores para programação do caracter gráfico, separados por vírgula.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Observações:**

Exemplo de programação do caracter gráfico:

- Para programar o caracter acima deve-se passar a seguinte string de parâmetros:

```
"001,002,004,008,016,032,064,128,064,032,016,008,004,002,129,129,129,129"
```

- Uma vez programado, este caracter será mantido na memória da impressora mesmo que a mesma seja desligada.

**Exemplo:****Exemplo em Visual Basic**

```
iRetorno = Elgin_ProgramaCaracterAutenticacao("001,002,004,008,016,032,064,128,064,032,016,008,004,002,129,129,129,129")
```

**Exemplo em Delphi**

```
sCoordenadas := '001,002,004,008,016,032,064,128,064,032,016,008,004,002,129,129,129,129';  
iRetorno := Elgin_ProgramaCaracterAutenticacao(sCoordenadas);
```

**3.2.11.17 Elgin\_ProgramaIdAplicativoMFD**

Programa a identificação do aplicativo no cupom fiscal.

**Parâmetro:**

**NomeAplicativo:** STRING com o nome do aplicativo com até 100 caracteres.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Observações:**

- O Nome do Aplicativo será salvo no arquivo Elgin.ini.

- Não há uma sequência pré estabelecida para a execução da função.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ProgramaIdAplicativoMFD("Nome do Aplicativo")
```

**// Exemplo em Delphi**

```
iResultado = Elgin_ProgramaIdAplicativoMFD('Nome do Aplicativo');
```

**3.2.11.18 Elgin\_ProgramaOperador**

Programa o nome do operador a ser utilizado na LeituraX, ReduçãoZ, Relatorios Gerenciais e em Cupons Fiscais. O nome do operador será mostrado no rodapé do documento.

**Parâmetro:**

**NomeOperador:** STRING com o nome do operador com até 8 caracteres.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_ProgramaOperador("Joao")
```

**// Exemplo em Delphi**

```
iResultado = Elgin_ProgramaOperador('Joao');
```

**Observações:**

- A informação não será armazenada permanentemente, ou seja, caso a dll seja reiniciada este comando deve ser executado novamente.

**3.2.11.19 Elgin\_RelatorioTipo60Analitico**

Gera o relatório "Tipo 60 analítico" exigido pelo convênio de ICMS 85/2001. As informações serão geradas no arquivo RETORNO.TXT no diretório configurado no parâmetro "path" do arquivo ini. O diretório default é o diretório raiz (C:\).

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

**Observações:**

- O relatório gerado terá o seguinte layout:

Tipo do relatório	60
Subtipo	A
Data de emissão	12/05/03
Número de série	5708990400028
Cancelamentos	11,14
Descontos	7,09
F	18,55
I	0,00
N	1,95
1200	0,00
1700	0,00
ISS	0,00

- Após a linha "N" serão gravadas as informações de vendas referentes às alíquotas tributárias. Portanto, o número de linhas irá variar de acordo com o número de alíquotas cadastradas na impressora.

- As informações contidas no relatório Tipo 60 Analítico são referentes aos dados da última Redução Z. Portanto, para gerar o relatório referente ao movimento do dia você deve executar essa função após a Redução Z, caso contrário as informações geradas serão referentes ao movimento do dia anterior.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_RelatorioTipo60Analitico()
```

#### // Exemplo em Delphi

```
iResultado := Elgin_RelatorioTipo60Analitico();
```

### 3.2.11.20 Elgin\_RelatorioTipo60Mestre

Gera o relatório "Tipo 60 Mestre" exigido pelo convênio de ICMS 85/2001. As informações serão geradas no arquivo RETORNO.TXT no diretório configurado no parâmetro "path" do arquivo ini. O diretório default é o diretório raiz (C:\).

#### Parâmetro:

Não há.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-8: Erro ao criar ou gravar no arquivo STATUS.TXT ou RETORNO.TXT.

-50: Número de série inválido.

#### Observações:

- O relatório gerado terá o seguinte layout:

Tipo do relatório	60
Subtipo	M
Data de emissão	12/05/03
Número de série	5708990400028
Número do equipamento	0001
Modelo do documento fiscal	2D
COO inicial	000001
COO final	000012
Contador de reduções	0307
Reinício de Operacao	0129
Venda Bruta	38,73
Totalizador geral	6169,21

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_RelatorioTipo60Mestre()
```

#### // Exemplo em Delphi

```
iResultado := Elgin_RelatorioTipo60Mestre();
```

### 3.2.11.21 Elgin\_VerificaImpressoraLigada

Verifica se a impressora está ligada ou conectada no computador.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_VerificaImpressoraLigada()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_VerificaImpressoraLigada();
```

### 3.2.11.22 Elgin\_VerificaReducaoZAutomatica

Verifica se a redução Z está configurada como automática.

**Parâmetro:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Observação:**

- Será retornado em Flag sempre "0", uma vez que não é possível fazer Z automática.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Elgin_VerificaReducaoZAutomatica()
```

**// Exemplo em Delphi**

```
iResultado := Elgin_VerificaReducaoZAutomatica();
```

### 3.2.11.23 RFD\_ConvertedaMFD

Permite gerar o arquivo RFD para reduções Z anteriores ou aplicável quando houver problema na geração automática do arquivo RFD(Registro de Fita-Detalhe).

**Parâmetro:**

**CRZ:** variável STRING que deve contar o contador da Redução Z que se deseja gerar o RFD(Registro de Fita-Detalhe).

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Observação:**

- Será gerado o diretório "RFD" com os arquivos no diretório onde estiver localizado o aplicativo.

- Deve estar setado para "1" o parâmetro "Gera\_RFD\_REDZ" no arquivo [Elgin.ini](#).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = RFD_ConverteMFD("0003")
```

**// Exemplo em Delphi**

```
iResultado := RFD_ConverteMFD('0003');
```

**3.2.11.24 RFD\_ConvertedaMFDData**

Permite gerar o arquivo RFD para reduções Z anteriores ou aplicável quando houver problema na geração automática do arquivo RFD(Registro de Fita-Detalhe).

**Parâmetro:**

**DataInicial:** variável STRING onde deve ser informada a data no formato: DDMMAA, DDMMAAAA, DD/MM/AA ou D/MM/AAAA.

**DataFinal:** variável STRING onde deve ser informada a data no formato: DDMMAA, DDMMAAAA, DD/MM/AA ou D/MM/AAAA.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = RFD_ConvertedaMFDData("01042008", "10042008")
```

**// Exemplo em Delphi**

```
iResultado := RFD_ConvertedaMFDData('01042008', '10042008');
```

**3.2.11.25 Elgin\_RelatorioSintegraMFD**

Permite gerar o Relatório Sintegra.

**Parâmetro:**

**iRelatorios:** variável INTEIRA com o tamanho de um byte, onde:

4: gera o relatório tipo 60D (Diário);

8: gera o relatório tipo 60I (Item);

16: gera o relatório tipo 60R (Resumo Mensal).

**cArquivo:** variável STRING com o path e nome do arquivo onde o relatório será gerado. Exemplo: "C:\SINTEGRA.TXT".

**cMes:** variável STRING com o mês, no formato MM.

**cAno:** variável STRING com o ano, no formato AAAA.  
**cRazaoSocial:** variável STRING com a razão social, no tamanho de até 35 caracteres.  
**cEndereco:** variável STRING com o endereço, no tamanho de até 34 caracteres.  
**cNumero:** variável STRING com o número, no tamanho de até 5 caracteres.  
**cComplemento:** variável STRING com o complemento, no tamanho de até 22 caracteres.  
**cBairro:** variável STRING com o bairro, no tamanho de até 15 caracteres.  
**cCidade:** variável STRING com a cidade, no tamanho de até 30 caracteres.  
**cCEP:** variável STRING com o CEP, com o tamanho de 8 caracteres.  
**cTelefone:** variável STRING com o telefone, no tamanho de até 12 caracteres.  
**cFax:** variável STRING com o fax, no tamanho de até 10 caracteres.  
**cContato:** variável STRING com o nome do contato, no tamanho de até 18 caracteres.

#### **Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.  
1: indica que nenhum erro ocorreu.  
-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.  
-5: Erro ao abrir a porta de comunicação.  
-6: Impressora desligada ou cabo de comunicação desconectado.  
-50: Número de série inválido.

#### **Observações:**

- Sempre serão gerados os registros 60M(Mestre), 60A(Analítico) e 75.  
- Para gerar mais de um relatório, deve-se enviar a soma dos valores dos tipos de registros, por exemplo: Se desejar gerar os relatórios tipo 60I e tipo 60D, a variável deve ter o valor 12.  
-Se desejar gerar os relatório tipo 60D, tipo 60I e tipo 60R, a variável deve ter o valor 28.

#### **Exemplos**

##### **' Exemplo em Visual Basic**

```
iRetorno = Elgin_RelatorioSintegraMFD (28, "SINTEGRA.TXT", "05", "2008", "Elgin S/A", "Campos Elíseos", "123", "Teste", "Centro", "São Paulo", "123456789", "0800 707 9890", "0800 707 9890", "Suporte" );
```

##### **// Exemplo em Delphi**

```
cArquivo := 'SINTEGRA.TXT';  
cMes := '05';  
cAno := '2008';  
cRazaoSocial := 'Elgin S/A';  
cEndereco := 'Campos Elíseos';  
cNumero := '123';  
cComplemento := 'Teste';  
cBairro := 'Centro';  
cCidade := 'São Paulo';  
cCEP := '123456789';  
cTelefone := '0800 707 9890';  
cFax := '0800 707 9890';  
cContato := 'Suporte';
```

```
iRetorno := Elgin_RelatorioSintegraMFD ( 28, pchar ( cArquivo ), pchar ( cMes ), pchar ( cAno ), pchar ( cRazaoSocial ), pchar ( cEndereco ), pchar ( cNumero, pchar ( cComplemento ), pchar ( cBairro ), pchar ( cCidade ), pchar ( cCEP ), pchar ( cTelefone ), pchar ( cFax ), pchar ( cContato ) );
```

#### **3.2.11.26 Elgin\_LeParametrosPAF**

Realiza a leitura automática de alguns parâmetros de inicialização do PAF ECF.

##### **Parâmetro:**

**CNPJ:** variável STRING (18 caracteres) que recebe o CNPJ do proprietário do ECF.

**Data:** variável STRING (06 caracteres) que recebe a data atual do ECF.

**Hora:** variável STRING (06 caracteres) que recebe a hora atual do ECF.

**Número de série:** variável STRING (21 caracteres) que recebe o Número de série do ECF.

**NúmeroECF:** variável STRING (4 caracteres) que recebe o número do ECF.

**Grande Total:** variável STRING (20 caracteres) que recebe o Grande total do ECF.

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

#### Exemplos

##### ' Exemplo em Visual Basic

```
Dim strRetorno01 As String
Dim strRetorno02 As String
Dim strRetorno03 As String
Dim strRetorno04 As String
Dim strRetorno05 As String
Dim strRetorno06 As String
strRetorno01 = Space(18)
strRetorno02 = Space(8)
strRetorno03 = Space(8)
strRetorno04 = Space(21)
strRetorno05 = Space(4)
strRetorno06 = Space(20)
iResultado = Elgin_LeParametrosPAF(strRetorno01, strRetorno02, strRetorno03, strRetorno04,
strRetorno05, strRetorno06)
```

##### // Exemplo em Delphi

```
Var
strRetorno01, strRetorno02, strRetorno03, strRetorno04, strRetorno05, strRetorno06: string
Begin
    strRetorno01 := StringOfChar(' ',18);
    strRetorno02 := StringOfChar(' ',8);
    strRetorno03 := StringOfChar(' ',8);
    strRetorno04 := StringOfChar(' ',21);
    strRetorno05 := StringOfChar(' ',4);
    strRetorno06 := StringOfChar(' ',20);
    mmLogComandos.Lines.Append('Elgin_LeParametrosPAF(CNPJ, Data, Hora, NumeroSerie,
NumeroECF,GrandeTotal)');
    iResultado := Elgin_LeParametrosPAF( strRetorno01, strRetorno02,
    strRetorno03, strRetorno04, strRetorno05, strRetorno06);
End;
```

#### 3.2.11.27 Elgin\_LeCodigoNacionalIdentificacaoECF

Retorna o Código Nacional de Identificação do ECF (CNI).

#### Parâmetro:

**CNI:** variável STRING (6 caracteres) que recebe o CNI (Código Nacional de Identificação do ECF).

#### Possíveis retornos da Função (INTEIRO):

0: indica erro na execução da função.

- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 6: Impressora desligada ou cabo de comunicação desconectado.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
Dim strRetorno01 As String
strRetorno01 = Space(6)
iResultado = Elgin_LeCodigoNacionalIdentificacaoECF(strRetorno01)
```

#### // Exemplo em Delphi

```
Var
strRetorno01: string
Begin
    strRetorno01 := StringOfChar(' ',6);
    iResultado := Elgin_LeCodigoNacionalIdentificacaoECF( strRetorno01 );
End;
```

### 3.2.11.28 Elgin\_ExecutaComando

Possibilita a execução de comandos do protocolo do ECF.

#### Parâmetro:

**Comando:** variável STRING com o comando a ser executado pelo protocolo do ECF.

**Parâmetros:** variável STRING com os parâmetros que serão pelo comando a ser executado pelo protocolo do ECF.

#### Possíveis retornos da Função (INTEIRO):

- 0: indica erro na execução da função.
- 1: indica que nenhum erro ocorreu.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 6: Impressora desligada ou cabo de comunicação desconectado.
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Elgin_ExecutaComando("AvancaPapel", "Avanco=100")
```

#### // Exemplo em Delphi

```
iResultado := Elgin_ExecutaComando( 'AvancaPapel','Avanco=100' );
```

#### Observações:

- Os comandos e parâmetros do protocolo do ECF são encontrados no manual de comunicação direta do ECF.

### 3.2.11.29 Elgin\_ExecutaLeitura

Possibilita a execução de leitura de registradores do protocolo do ECF.

**Parâmetro:**

**Comando:** variável STRING com o comando a ser executado pelo protocolo do ECF.

**Parâmetros:** variável STRING com os parâmetros que serão pelo comando a ser executado pelo protocolo do ECF.

**Retorno:** variável STRING com até 4000 caracteres que recebe o retorno do comando enviado para o ECF.

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
dim strRetorno01 as String
strRetorno01 = Space(256)
iResultado = Elgin_ExecutaLeitura("LeInteiro", "NomeInteiro=" & Chr(34) &
"EspacamentoDocumentos" & Chr(34), strRetorno01)
```

**// Exemplo em Delphi**

```
Var
strRetorno01: string;
Begin
    strRetorno01 := StringOfChar(' ',256);
    iResultado := Elgin_ExecutaLeitura(
'LeInteiro','NomeInteiro="'EspacamentoDocumentos"',strRetorno01 );
End;
```

**Observações:**

- Os comandos e parâmetros do protocolo do ECF são encontrados no manual de comunicação direta do ECF.

- O tamanho do retorno vai depender do comando a ser utilizado na função.

### 3.2.11.30 Elgin\_TotalIcmsCupom

Retorna o valor total do ICMS emitido no cupom fiscal.

**Parâmetro:**

**TotalICMS:** variável STRING (14 caracteres) que recebe o total do ICMS emitido no cupom

**Possíveis retornos da Função (INTEIRO):**

0: indica erro na execução da função.

1: indica que nenhum erro ocorreu.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-6: Impressora desligada ou cabo de comunicação desconectado.

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
dim strRetorno01 as String
strRetorno01 = Space(14)
iResultado = Elgin_TotalIcmsCupom(strRetorno01)
```

**// Exemplo em Delphi**

```
Var
strRetorno1: string;
Begin
    strRetorno01 := StringOfChar(' ',14);
    iResultado := Elgin_TotalIcmsCupom( strRetorno01 );
End;
```

**Observações:**

- A flag 'HabilitaICMSCupom na área [Sistema] no elgin.ini deve ser igual a 1 para habilitar o retorno o valor do ICMS do ultimo culpom.
- Caso em [ECF] no elgin.ini o estado esteja RJ, a impressão do Cupom Mania sera feita automaticamente.

**3.2.12 Funções para impressora Wind****3.2.12.1 Wind\_AcionaGaveta**

Gera um pulso para a gaveta.

**Parâmetros:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

- 0 (zero): Erro de Comunicação.
- 1 (um): OK.
- 2 (menos dois): Parâmetro Inválido.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 10: Impressora offline ou desligada
- 50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
Retorno = Wind_AcionaGaveta:integer()
```

**// Exemplo em Delphi**

```
Retorno := Wind_AcionaGaveta:integer();
```

**3.2.12.2 Wind\_AcionaGuilhotina**

Aciona a guilhotina da impressora.

**Parâmetros:**

**Modo:** variável INTEIRA para definir o tipo de corte do papel. Onde:

- 0: para corte parcial ou;
- 1: para corte total.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

Retorno = Wind\_AcionaGuilhotina(1)

**// Exemplo em Delphi**

Retorno := Wind\_AcionaGuilhotina(1);

**3.2.12.3 Wind\_AcionaGuilhotinaParcial**

Aciona a guilhotina da impressora para cortes parciais.

**Parâmetros:**

**Modo:** variável INTEIRA para definir o tipo de corte do papel. Podendo variar de 1 a 10.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

Retorno = Wind\_AcionaGuilhotinaParcial (5)

**// Exemplo em Delphi**

Retorno := Wind\_AcionaGuilhotinaParcial (10);

**3.2.12.4 Wind\_AjustaLarguraPapel**

Ajusta a largura do papel.

**Parâmetros:**

**LarguraPapel:** Variável inteira que recebe o valor largura do papel.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

- 5: Erro ao abrir a porta de comunicação.
- 10: Impressora offline ou desligada
- 50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
Retorno = Wind_AjustaLarguraPapel(50)
```

**// Exemplo em Delphi**

```
Retorno := Wind_AjustaLarguraPapel(50);
```

**3.2.12.5 Wind\_ConfiguraCodigoBarras**

Configura a altura, largura, posição dos caracteres, fonte e margem do código de barras.

**Parâmetros:**

**Altura:** Variável inteira que pode receber valores entre 1 à 255. (default 162).

**Largura:** Variável inteira que pode receber valores entre 0 à 2. Largura = 0 (barras finas), Largura = 1 (barras médias) - default, Largura = 2 (barras grossas).

**PosicaoCaracteres:** Variável inteira que pode receber valores entre 0 à 3. Posição = 0 (não imprime os caracteres do código), Posição = 1 (imprime os caracteres acima do código), Posição = 2 (imprime os caracteres abaixo do código) - default, Posição = 3 (imprime os caracteres acima e abaixo do código).

**Fonte:** Variável inteira que pode receber valores entre 0 à 1. Fonte = 0 (normal), Fonte = 1 (condensado).

**Margem:** Variável inteira que pode receber valores entre 0 à 575 (dots pitch). Margem = 0 (sem margem) default.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
Retorno = Wind_ConfiguraCodigoBarras(100, 1, 2, 1, 5)
```

**// Exemplo em Delphi**

```
Retorno := Wind_ConfiguraCodigoBarras(100, 1, 2, 1, 5);
```

**3.2.12.6 Wind\_EnviaBuffer**

Envia uma sequência de caracteres a impressora.

**Parâmetros:**

**Buffer:** Variável string que armazena o buffer a ser enviado a impressora.

**Possíveis retornos da Função (INTEIRO):**

- 0 (zero): Erro de Comunicação.
- 1 (um): OK.
- 2 (menos dois): Parâmetro Inválido.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 10: Impressora offline ou desligada
- 50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
Retorno = Wind_EnviaBuffer("Wind_ImprimeCodigoBarrasUPCA" + chr(10))
```

**// Exemplo em Delphi**

```
Retorno := Wind_EnviaBuffer('Wind_ImprimeCodigoBarrasUPCA' + chr(10));
```

**3.2.12.7 Wind\_EnviaBufferFormatado**

Envia uma sequência de caracteres formatados a impressora.

**Parâmetros:**

**Buffer:** Variável string que armazena o buffer a ser enviado a impressora.

**TipoLetra:** Variável inteira que armazena o tipo de de letra, podendo ser 0 ou 1.

**Italico:** Variável inteira que informa se a fonte vai ficar em itálico. Ela pode assumir os valores 0 ou 1.

**Sublinhado:** Variável inteira que informa se a fonte vai ficar sublinhada. Ela pode assumir os valores 0 ou 1.

**Expandido:** Variável inteira que informa se a fonte vai ficar expandida. Ela pode assumir os valores 0 ou 1.

**Enfatizado:** Variável inteira que informa se a fonte vai ficar enfatizada. Ela pode assumir os valores 0 ou 1.

**Possíveis retornos da Função (INTEIRO):**

- 0 (zero): Erro de Comunicação.
- 1 (um): OK.
- 2 (menos dois): Parâmetro Inválido.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 10: Impressora offline ou desligada
- 50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
Retorno = Wind_EnviaBufferFormatado("Wind_ImprimeCodigoBarrasUPCA" + chr(10), 0, 0, 1, 1, 0)
```

**// Exemplo em Delphi**

```
Retorno := Wind_EnviaBufferFormatado('Wind_ImprimeCodigoBarrasUPCA' + chr(10), 0, 0, 1, 1, 0);
```

### 3.2.12.8 Wind\_EnviaComando

Serve para enviar comandos em hexadecimal ou decimal para a impressora.

**Parâmetros:**

**Buffer:** STRING que armazena o buffer a ser enviado a impressora.

**TamanhoBuffer:** INTEIRO que armazena o tamanho do buffer.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
strBuffer = "Wind_ImprimeCodigoBarrasUPCA"
```

```
Retorno = Wind_EnviaComando(strBuffer, len(strBuffer))
```

**// Exemplo em Delphi**

```
strBuffer = 'Wind_ImprimeCodigoBarrasUPCA';
```

```
Retorno := Wind_EnviaComando(strBuffer, length(strBuffer));
```

### 3.2.12.9 Wind\_ImprimeCodigoBarrasCODABAR

Esta função faz a impressão do código de barras CODABAR.

**Parâmetro:**

**Código:** STRING do código que será gerado.

O tamanho da string é dada pela tabela abaixo.

**Largura das Barras    Quantidade de Caracteres**

0	20
1	12
2	8

A Largura das Barras é 1 (default).

**Observações:**

- Será acrescentado, automaticamente, o dígito verificador.

- Aceita dígitos entre 0 à 9.

- Aceita as letras A, B, C e D (maiúsculas ou minúsculas).

- Aceita os caracteres: "\$", "+", "-", ".", "/" e ":".

- Ver o tópico [DLL para wind paralela](#).

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

- 2 (menos dois): Parâmetro Inválido.
- 4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.
- 5: Erro ao abrir a porta de comunicação.
- 10: Impressora offline ou desligada
- 50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Wind_ImprimeCodigoBarrasCODABAR("123-ABC/001")
```

#### // Exemplo em Delphi

```
cCodigo := '100-ZWK/001';
```

```
iResultado := Wind_ImprimeCodigoBarrasCODABAR( pchar ( cCodigo ) );
```

### 3.2.12.10 Wind\_ImprimeCodigoBarrasCODE128

Esta função faz a impressão do código de barras CODE128.

#### Parâmetro:

**Código:** STRING com o código que será gerado.

O tamanho da string é dada pela tabela abaixo:

#### Largura das Barras    Quantidade de Caracteres

0	42
1	28
2	16

A Largura das Barras é 1 (default).

#### Observações:

- Aceita os caracteres da tabela ASCII, na faixa de valores de 001 à 127.
- Ver o tópico [DLL para wind paralela](#).

#### Possíveis retornos da Função (INTEIRO):

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

### Exemplos

#### ' Exemplo em Visual Basic

```
iResultado = Wind_ImprimeCodigoBarrasCODE128("Elgin")
```

#### // Exemplo em Delphi

```
cCodigo := 'Código';
```

```
iResultado := Wind_ImprimeCodigoBarrasCODE128( pchar ( cCodigo ) );
```

### 3.2.12.11 Wind\_ImprimeCodigoBarrasCODE39

Esta função faz a impressão do código de barras CODE39.

#### Parâmetro:

**Código:** STRING com o código que será gerado.

O tamanho da string é dada pela tabela abaixo:

**Largura das Barras    Quantidade de Caracteres**

0	15
1	9
2	6

A Largura das Barras é 1 (default).

**Observações:**

- Será acrescentado, automaticamente, o dígito verificador.
- Aceita dígitos entre 0 à 9.
- Aceita letras de A à Z (maiúsculas ou minúsculas).
- Aceita os caracteres: "espaço em branco", "\$", "%", "+", "-", "." e "/".
- As letras não podem se maiúsculas e minúsculas simultaneamente.
- Ver o tópico [DLL para wind paralela](#).

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Wind_ImprimeCodigoBarrasCODE39("4ab*TEST80")
```

**// Exemplo em Delphi**

```
cCodigo := 'abc-123';
```

```
iResultado := Wind_ImprimeCodigoBarrasCODE39( pchar ( cCodigo ) );
```

**3.2.12.12 Wind\_ImprimeCodigoBarrasCODE93**

Esta função faz a impressão do código de barras CODE93.

**Parâmetro:**

**Código:** STRING com o código que será gerado.

O tamanho da string é dada pela tabela abaixo:

**Largura das Barras    Quantidade de Caracteres**

0	15
1	9
2	6

A Largura das Barras é 1 (default).

**Observações:**

- Será acrescentado, automaticamente, o dígito verificador.
- Aceita os caracteres da tabela ASCII, na faixa de valores de 001 à 127.
- Ver o tópico [DLL para wind paralela](#).

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Wind_ImprimeCodigoBarrasCODE93("AbCdEfGh12")
```

**// Exemplo em Delphi**

```
cCodigo := '123-ABC';
```

```
iResultado := Wind_ImprimeCodigoBarrasCODE93( pchar ( cCodigo ) );
```

**3.2.12.13 Wind\_ImprimeCodigoBarrasEAN13**

Esta função faz a impressão do código de barras EAN13.

**Parâmetro:**

**Codigo:** STRING com o tamanho de 12 dígitos de 0 à 9.

**Observação:**

- Será acrescentado, automaticamente, o dígito verificador.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Wind_ImprimeCodigoBarrasEAN13("123456789012")
```

**// Exemplo em Delphi**

```
cCodigo := '123456789012';
```

```
iResultado := Wind_ImprimeCodigoBarrasEAN13( pchar ( cCodigo ) );
```

**3.2.12.14 Wind\_ImprimeCodigoBarrasEAN8**

Esta função faz a impressão do código de barras EAN8.

**Parâmetro:**

**Codigo:** STRING com o tamanho de 7 dígitos de 0 à 9.

**Observação:**

- Será acrescentado, automaticamente, o dígito verificador.
- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Wind_ImprimeCodigoBarrasEAN8("3551234")
```

**// Exemplo em Delphi**

```
cCodigo := '1234567';
```

```
iResultado := Wind_ImprimeCodigoBarrasEAN8( pchar ( cCodigo ) );
```

**3.2.12.15 Wind\_ImprimeCodigoBarrasISBN**

Esta função faz a impressão do código de barras ISBN.

**Parâmetro:**

**Código:** STRING com 9 dígitos, composto por dígitos de 0 à 9, "-" (hifen) e "X". O "-" (hifen) e o "X" não são somados. Após o 9º dígito, podem aparecer ainda hifens seguidos por "X" ou algum número com o tamanho de 5 caracteres, não somando o espaço após o "-X" ou após o número. Exemplo: "1-56592-292-X 9000" ou "1-56592-291-1 900000".

**Observações:**

- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Wind_ImprimeCodigoBarrasISBN("51234-56-789 00000")
```

**// Exemplo em Delphi**

```
cCodigo := '1-56592-292-X 90000';
```

```
iResultado := Wind_ImprimeCodigoBarrasISBN( pchar ( cCodigo ) );
```

### 3.2.12.16 Wind\_ImprimeCodigoBarrasITF

Esta função faz a impressão do código de barras ITF (Intercalado 2/5).

**Parâmetro:**

**Código:** STRING com o código que será gerado.

O tamanho da string é dada pela tabela abaixo:

Largura das Barras	Quantidade de Caracteres
0	30
1	20
2	14

A Largura da Barras é 1 (default).

**Observações:**

- Aceita dígitos entre 0 à 9.
- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.
- Ver o tópico [DLL para wind paralela](#).

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Wind_ImprimeCodigoBarrasITF("12345678901234567890")
```

**// Exemplo em Delphi**

```
cCodigo := '0123456789012345';
```

```
iResultado := Wind_ImprimeCodigoBarrasITF( pchar ( cCodigo ) );
```

### 3.2.12.17 Wind\_ImprimeCodigoBarrasMSI

Esta função faz a impressão do código de barras MSI.

**Parâmetro:**

**Código:** STRING com o código que será gerado.

O tamanho da string é dada pela tabela abaixo:

**Largura das Barras    Quantidade de Caracteres**

0	16
1	10
2	7

A Largura da Barras é 1 (default).

**Observações:**

- Será acrescentado, automaticamente, o dígito verificador.
- Aceita dígitos entre 0 à 9.
- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.
- Ver o tópico [DLL para wind paralela](#).

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Wind_ImprimeCodigoBarrasMSI("69876543")
```

**// Exemplo em Delphi**

```
cCodigo := '123';
```

```
iResultado := Wind_ImprimeCodigoBarrasMSI( pchar ( cCodigo ) );
```

**3.2.12.18 Wind\_ImprimeCodigoBarrasPDF417**

Esta função faz a impressão do código de barras PDF417.

**Parâmetros:**

**nível:** Inteiro entre 0 à 8. Quanto mais alto o nível, melhor a leitura do código, maior a impressão e menor o número de informações que poderão ser impressas.

**altura:** Inteiro entre 1 à 8. Altura do caracter do código (pitch). 1 pitch = altura de 0,125 mm.

**largura:** Inteiro entre 1 à 4. Largura do caracter do código (pitch). 1 pitch = altura de 0,125 mm.

**númColunas:** Inteiro entre 0 à 30. "0" (zero) utiliza o máximo de colunas que o mecanismo permite para a largura informada (pitch). Caso não caiba na linha a impressora ajusta, automaticamente, para o máximo de colunas permitido na linha.

**codigo:** String do código que deseja gerar.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

- 5: Erro ao abrir a porta de comunicação.
- 10: Impressora offline ou desligada
- 50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Wind_ImprimeCodigoBarrasPDF417(4, 3, 2, 0, "Impressoras Elgin !!!")
```

**// Exemplo em Delphi**

```
cCodigo := 'Impressoras Elgin !!!';
```

```
iResultado := Wind_ImprimeCodigoBarrasPDF417( 4, 3, 2, 0, pchar ( cCodigo ) );
```

**3.2.12.19 Wind\_ImprimeCodigoBarrasPLESSEY**

Esta função faz a impressão do código de barras PLESSEY.

**Parâmetro:**

**Código:** STRING com o código que será gerado.

O tamanho da string é dada pela tabela abaixo:

**Largura das Barras    Quantidade de Caracteres**

0	13
1	7
2	4

A Largura das Barras é 1 (default).

**Observações:**

- Será acrescentado, automaticamente, o dígito verificador.
- Aceita dígitos entre 0 à 9.
- Aceita, somente, as letras ABCDEF (maiúsculas e minúsculas).
- As letras não podem ser maiúsculas e minúsculas simultaneamente.
- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.
- Ver o tópico [DLL para wind paralela](#).

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Wind_ImprimeCodigoBarrasPLESSEY("ABC0123")
```

**// Exemplo em Delphi**

```
cCodigo := '123-ABC';
```

```
iResultado := Wind_ImprimeCodigoBarrasPLESSEY( pchar ( cCodigo ) );
```

### 3.2.12.20 Wind\_ImprimeCodigoBarrasUPCA

Esta função faz a impressão do código de barras UPCA.

**Parâmetro:**

**Código:** STRING com o tamanho de 11 dígitos de 0 à 9.

**Observações:**

- Será acrescentado, automaticamente, o dígito verificador.
- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos**

**' Exemplo em Visual Basic**

```
iResultado = Wind_CodigoBarrasUPCA("12345678901")
```

**// Exemplo em Delphi**

```
ret := Wind_EnviaBuffer('Wind_ImprimeCodigoBarrasUPCA'+chr(10));
```

```
ret := Wind_ImprimeCodigoBarrasUPCA('12345678901');
```

### 3.2.12.21 Wind\_ImprimeCodigoBarrasUPCE

Esta função faz a impressão do código de barras UPCE.

**Parâmetro:**

**Código:** STRING com o tamanho de 6 dígitos de 0 à 9.

**Observação:**

- Será acrescentado, automaticamente, o dígito verificador.
- A impressão do código de barras só será executado dentro do Relatório Gerencial ou dentro do Comprovante Não Fiscal Vinculado.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): OK.

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
iResultado = Wind_CodigoBarrasUPCE("123456")
```

**// Exemplo em Delphi**

```
cCodigo := '123456';  
iResultado := Wind_CodigoBarrasUPCE( pchar ( cCodigo ) );
```

**3.2.12.22 Wind\_VerificaEstadoGaveta**

Verifica se a gaveta está aberta ou fechada.

**Parâmetros:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0: Erro de Comunicação

1: Gaveta Aberta

2: Gaveta Fechada

-2 (menos dois): Parâmetro Inválido.

-4: O arquivo de inicialização Elgin.ini não foi encontrado no diretório de sistema do Windows.

-5: Erro ao abrir a porta de comunicação.

-10: Impressora offline ou desligada

-50: Número de série inválido.

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
Retorno = Wind_VerificaEstadoGaveta()
```

**// Exemplo em Delphi**

```
Retorno := Wind_VerificaEstadoGaveta();
```

**3.2.12.23 Wind\_VerificaFimPapel**

Verifica se a impressora está sem papel.

**Parâmetros:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): Impressora com papel

2 (dois): Impressora sem papel

-10: Impressora offline ou desligada

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
Retorno = Wind_VerificaFimPapel()
```

**// Exemplo em Delphi**

```
Retorno := Wind_VerificaFimPapel();
```

### 3.2.12.24 Wind\_VerificaOnLine

Verifica se a impressora está ligada ou desligada.

**Parâmetros:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Impressora está Offline.

1 (um): Impressora está Online.

-10: Impressora offline ou desligada

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
Retorno = Wind_VerificaOnLine()
```

**// Exemplo em Delphi**

```
Retorno := Wind_VerificaOnLine();
```

### 3.2.12.25 Wind\_VerificaPoucoPapel

Verifica se a impressora está com pouco papel.

**Parâmetros:**

Não há.

**Possíveis retornos da Função (INTEIRO):**

0 (zero): Erro de Comunicação.

1 (um): Papel suficiente

2 (dois): Pouco papel

-10: Impressora offline ou desligada

**Observação:**

- Ver o tópico [DLL para wind paralela](#).

**Exemplos****' Exemplo em Visual Basic**

```
Retorno = Wind_VerificaPoucoPapel()
```

**// Exemplo em Delphi**

```
Retorno := Wind_VerificaPoucoPapel();
```

## 4 Configuração do arquivo elgin.ini

A DLL Elgin.dll possui um arquivo de configuração chamado Elgin.ini onde algumas opções podem ser configuradas. Esse arquivo de configuração possui seções com alguns parâmetros de configuração, a saber:

**Seção [Sistema]**

Essa seção possui as seguintes opções de configuração:

**Porta=** Indica a porta serial ou paralela (no caso da Wind paralela) de comunicação entre a impressora e o computador. Se o valor for igual a "DEFAULT", a DLL buscará a porta onde estiver a impressora, automaticamente.

**Path=** Configura o diretório de escrita para arquivo "RETORNO.TXT". O path default é o diretório raiz na unidade C:\.

**HabilitaICMSCupom** - Quando esse parâmetro é igual a 1 habilita o valor do ICMS do ultimo cupom (válido somente para o Cupom Mania UF = RJ).

**DINHEIRO** Identifica o índice da Finalizadora "Dinheiro", mas só poderá ser utilizado para as impressoras IF400 e IF500. A DLL ainda não dá suporte a esses modelos de impressora.

**EspacamentoDocumentos** Indica o número de linhas para serem avançadas ao final dos cupons.

**Sangria** Indica o índice do Não Vinculado que corresponde a Sangria.

**Suprimento** Indica o índice do Não Vinculado que corresponde ao Suprimento.

**Homologar** Se está opção estiver com o valor igual a "1", a DLL irá gerar dois arquivos 'LOG.DAT' e 'HOMOL.DAT', os dois trazem a seqüência de comandos e respostas do ECF, e tem como objetivo validar a homologação da software house. O arquivo 'LOG.DAT' traz a informação pura, enquanto o 'HOMOL.DAT' traz a mesma criptografada. Estes arquivos podem ser utilizados também para se saber o que de fato está sendo enviado e recebido da Impressora, facilitando o debug por parte do suporte e até mesmo do usuário.

**Baud** Este parâmetro recebe o valor da velocidade programada na impressora. As velocidades pode ser: 115200, 9600, 57600 etc.

**HabilitaCorte** Este parâmetro é utilizado somente para a IF 600 TH e X-5. Quando este parâmetro for igual a 0 o papel não será cortado após a emissão do cupom e quando o parâmetro for diferente de 0, p papel será cortado após a emissão de um cupom. Ex: habilitaCorte=1.

**GERA\_RFD\_REDZ** Esta chave habilita (1) ou desabilita (0) a geração do arquivo para a CAT-52 ou Ato Cotepe 17/04. Ao emitir a Redução Z do dia a impressora piscará o LED "On Line" por tempo indeterminado (proporcional ao número de transações do dia). No diretório da aplicação será criado uma pasta chamada "RFD" onde serão colocados todos os arquivos gerados, isto de acordo com o padrão estabelecido na legislação.

**TIMEOUTREDZ** Está chave é utilizada na configuração do tempo de espera (em minutos), após a execução da função Elgin\_ReducãoZ.

**ENDereco** Este parâmetro só é utilizado para a Wind paralela. Ela serve para informar o endereço base para a porta paralela. O valor default é 378H, mas caso a porta possua outro endereço, por exemplo 278H, será necessário colocar 278 no valor desse parâmetro. Veja o tópico [DLL para wind paralela](#).

**PathRFD** Diretório de escrita para a geração do arquivo de RFD (Registro da Fita Detalhe).

**MODOGAVETA** É utilizada para compatibilizar o sinal lógico que a gaveta envia para a impressora, quando a mesma é acionada (aberta) ou não. O Default é 0 (zero), ou seja, o sinal enviado pela gaveta é o mesmo retornado pela DLL. Em alguns modelos de gaveta, este sinal poderá vir invertido, ou seja, o sinal que corresponde a "gaveta aberta", pode ser o sinal para "gaveta fechada". Neste caso habilite esta chave, substituindo o valor 0 (zero) para (1) um.

**TimeOutComandos** Está chave configura o tempo para time outs gerais, o valor default é 10 segundos (Mínimo 5 segundos).

**ChecaCTS** Flags que indica se o CTS do ECF deve ser checado ou não. Caso esteja marcado com "1", a DLL verifica se o CTS está ligado por 2 segundos antes de informar erro de comunicação. Com isto, pode ocorrer uma agilidade muito maior no retorno de timeout nos casos em que o ECF esteja desligado. Default 0 (desabilitado).

**CorrigeRelogioZ** Este flag tem como default o valor 1 e deve ser utilizado para acerto automático do relógio durante a Redução Z, usando como referência a hora do PC, lembrando que o ECF só permite o acerto de 5 em 5 minutos.

**LMFAutomatico** Flags que ativa ou desativa a emissão automatica da Leitura da Memória Fiscal automatica na virada do mês.

**NumPkt** - Número do pacote enviado para a impressora.

### Seção [ECF]

Esta seção corresponde aos dados do ECF pertinentes à geração dos registros da portaria CAT-52

**IM=** Refere-se à Inscrição Municipal do estabelecimento que possui a ECF, com até 20 posições (opcional).

**UF=** Refere-se à Unidade da Federação do estabelecimento que possui a ECF, com até 2 posições.

### Conteúdo do arquivo

[Sistema]

ModeloImp=ELGIN\_FIT

Porta=COM1

Path=C:\

HabilitaICMSCupom=1

DINHEIRO=0

EspacamentoDocumentos=7

SANGRIA=1

```
SUPRIMENTO=0
HOMOLOGAR=1
BAUD=9600
HabilitaCorte=1
Gera_RFD_REDZ=1
TimeOutRedZ=40
Endereco=278
PathRFD=C:\RFD
TimeOutComandos=10
ChecaCTS=0
CorrigeRelogioZ=1
LMFAutomatico=1
[Flags]
NumPkt=133
```

**ModeloImp** - Modelo da Impressora em desenvolvimento

Possibilidades:

```
ModeloImp=ELGIN_X5
ModeloImp=ELGIN_WIND_LPT
ModeloImp=ELGIN_WIND_SERIAL
ModeloImp=ELGIN_FIT
ModeloImp=ELGIN_IF6000TH
```

#### **Observação:**

- O arquivo ELGIN.ini pode estar no diretório da aplicação, C:\WINDOWS\system32 ou C:\WINDOWS. Note que se não existir um novo arquivo é criado no diretório da aplicação.
- A emissão da LMF não é realizada ao carregar a DLL, ou seja, para disparar a sua emissão, algum comando precisa ser executado (Ex: Emissão de Cupom Fiscal, Relatório Gerencial, etc) exceto Redução Z; A verificação da "virada" do mes, é executada na primeira vez que a DLL foi carregada. Logo, em situações em que o PC não é desligado, o software deverá ser reinicializado.

## **5 Configuração do arquivo cheques.ini**

O arquivo CHEQUES.ini deve estar no diretório da aplicação e só será utilizado quando a impressora utilizada for a IF 6000 TH.

#### **Conteúdo do arquivo**

```
[21]
Observacao= Banco Real S.A.
HPosAno=1640
HPosCidade=900
HPosDia=1100
HPosExtensoLinha1=320
HPosExtensoLinha2=31
HPosFavorecido=52
HPosMes=1343
HPosMsgLinha1=900
HPosValor=1400
VPosCidade=399
VPosExtensoLinha1=220
VPosExtensoLinha2=240
VPosFavorecido=280
VPosMsgLinha1=650
VPosValor=120
```

[21] - Parâmetro com o número do banco. Este só pode ter no máximo 3 dígitos.

**Observacao** - Parâmetro para informar o nome do banco.

**HPosAno** - Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm,

para impressão do campo Ano no documento. A posição vertical usada para este campo é a VPosCidade.

**HPosCidade** - Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm, para impressão do campo Cidade no documento.

**HPosDia** - Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm, para impressão do campo Dia no documento. A posição vertical usada para este campo é VPosCidade.

**HPosExtensoLinha1** - Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm, para impressão da primeira linha do Extenso, gerado pelo ECF, referente ao valor do cheque.

**HPosExtensoLinha2** - Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm, para impressão da segunda linha do Extenso, gerado pelo ECF, referente ao valor do cheque.

**HPosFavorecido** - Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm, para impressão do campo Favorecido no documento.

**HPosMes** - Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm, para impressão do campo Mes no documento. A posição vertical usada para este campo é VPosCidade.

**HPosMsgLinha1** - Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm, para impressão do campo MensagemDocLinha1 no documento.

**HPosValor** - Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm, para impressão do campo Valor no documento.

**VPosCidade** - Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo Cidade no documento.

**VPosExtensoLinha1** - Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão da primeira linha do Extenso no documento.

**VPosExtensoLinha2** - Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão da segunda linha do Extenso no documento.

**VPosFavorecido** - Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo Favorecido no documento.

**VPosMsgLinha1** - Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo MensagemDocLinha1 no documento.

**VPosValor** - Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo Valor no documento.

## 6 Sequência de comandos

### 6.1 Venda Completa

Para realizar uma venda completa utilizando as funções da Elgin.dll devem ser executados os comandos seguindo os exemplos apresentados nas seqüências:

#### Seqüência 1:

1. [Elgin\\_AbreCupom](#) ou [Elgin\\_AbreCupomMFD](#)
3. [Elgin\\_VendeItem](#) ou [Elgin\\_VendeItemDepartamento](#)
4. [Elgin\\_InciaFechamentoCupom](#)
5. [Elgin\\_EfetuaFormaPagamento](#)
6. [Elgin\\_TerminaFechamentoCupom](#)

#### Seqüência 2:

1. [Elgin\\_AbreCupom](#) ou [Elgin\\_AbreCupomMFD](#)
2. [Elgin\\_VendeItem](#) ou [Elgin\\_VendeItemDepartamento](#)
3. [Elgin\\_EfetuaFormaPagamento](#)
4. [Elgin\\_TerminaFechamentoCupom](#)

**Sequência 3:**

1. [Elgin\\_AbreCupom](#) ou [Elgin\\_AbreCupomMFD](#)
2. [Elgin\\_VendeItem](#) ou [Elgin\\_VendeItemDepartamento](#)
3. [Elgin\\_FechaCupom](#)

**Sequência 4:**

1. [Elgin\\_AbreCupom](#) ou [Elgin\\_AbreCupomMFD](#)
2. [Elgin\\_VendeItem](#) ou [Elgin\\_VendeItemDepartamento](#)
3. [Elgin\\_FechaCupomResumido](#)

**Sequência 5 (Identificação do consumidor no final do cupom):**

1. [Elgin\\_AbreCupom](#) (Não passar o CPF do consumidor)
2. [Elgin\\_VendeItem](#) ou [Elgin\\_VendeItemDepartamento](#)
3. [Elgin\\_IdentificaConsumidor](#)
4. Executar as funções de fechamento do cupom

## 7 Procedimentos para usar o Emulador Log2 com a DLL Elgin

Atualmente as únicas maneiras de usar a DLL Elgin com o emulador (simulador de ECF) é fazendo a conexão de duas portas seriais usando um cabo serial NullModemDB9-DB9 para fazer um "loop" entre essas portas.

**Como fazer usando um PC com duas seriais:**

Quando for usado um PC com duas portas seriais ( Com1 e Com2 ) Você vai precisar de 1 cabo serial NullModemDB9-DB9 (verificar pinagem no tópico: Pinagem do cabo NullModem) que faça o "loop" para ligar a COM1 com a COM2. Depois das portas conectadas fisicamente, você deve abrir o simulador e na opção "Canal p/ Responder comandos" você coloca COM2, por exemplo, no seu aplicativo você usa a porta COM1 (configurada no arquivo Elgin.ini), em seguida clique no botão "conectar-se ao ECF" ao lado do canal.

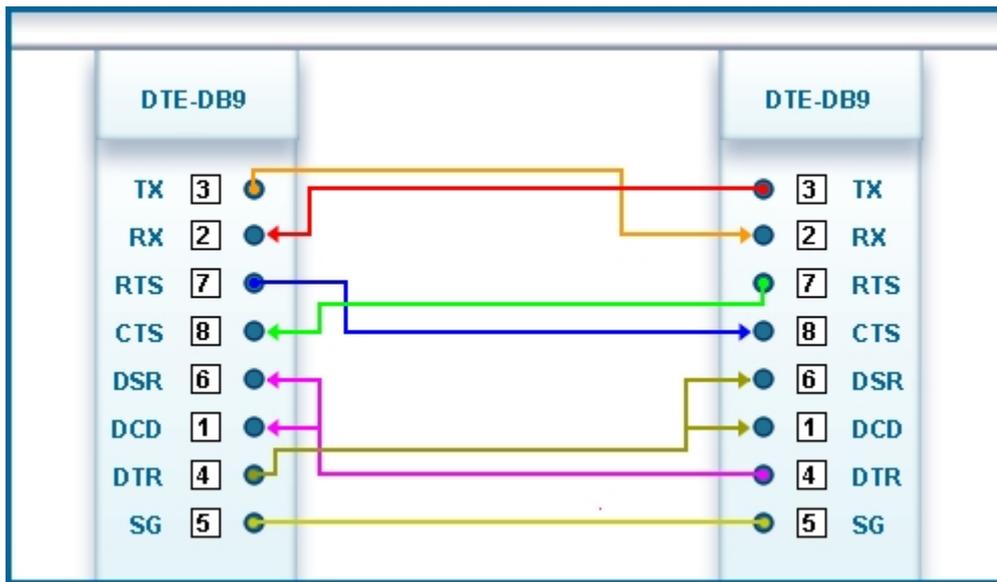
Isso fará com que os comandos executados pelo seu aplicativo sejam enviados para o emulador, através da porta COM1, e o emulador responda pela porta COM2 para seu aplicativo.

**Como fazer usando dois PCs:**

Quando forem usados dois PCs (PC1 e PC2) cada um com uma porta serial Você irá precisar do cabo serial NullModemDB9-DB9 (verificar pinagem no tópico: Pinagem do cabo NullModem), mas agora vai conectar a COM1 do PC1 com a COM1 do PC2. No emulador você deve colocar a COM1 na opção "Canal p/ Responder comandos" e seu aplicativo deve trabalhar também com a porta COM1, em seguida clique no botão "conectar-se ao ECF" ao lado do canal.

Neste caso no PC1 rodará o emulador e no PC2 a aplicação.

**Pinagem do cabo NullModem:**

**OBSERVAÇÕES:**

1. Futuramente será implementado a comunicação por software entre a Dll e o emulador.
2. O emulador de ECF (LOG2) pode ser encontrado no site da Elgin [www.elgin.com.br](http://www.elgin.com.br)

## 8 Retornos da Impressora

0	NoError	Não houve erro algum.
1	ErroGeralFaltaRAM	Não foi possível alocar mais memória.
2	ErroGeralPerdaRAM	Memória RAM foi corrompida.
1000	ErroMFDesconectada	Memória Fiscal foi desconectada.
1001	ErroMFLeitura	Erro de leitura na Memória Fiscal.
1002	ErroMFApenasLeitura	Memória está setada apenas para leitura.
1003	ErroMFTamRegistro	Registro fora dos padrões (erro interno).
1004	ErroMFCheia	Memória Fiscal está lotada.
1005	ErroMFCartuchosExcedidos	Número máximo de cartuchos excedidos.
1006	ErroMFJaInicializada	Tentativa de gravar novo modelo de ECF.
1007	ErroMFNaoInicializada	Tentativa de gravação de qualquer dado antes da inicialização da Memória Fiscal.
1008	ErroMFUsuariosExcedidos	Número máximo de usuários foi atingido.
1009	ErroMFIntervencoesExcedidas	Número máximo de intervenções foi atingido.
1010	ErroMFVersoesExcedidas	Número máximo de versões foi atingido.
1011	ErroMFReducoesExcedidas	Número máximo de reduções foi atingido.
1012	ErroMFGravacao	Erro na gravação de registro na memória fiscal
2000	ErroTransactDrvrLeitura	Erro de leitura no dispositivo físico.
2001	ErroTransactDrvrEscrita	Erro de leitura no dispositivo.
2002	ErroTransactDrvrDesconectado	Dispositivo de transações foi desconectado.
3000	ErroTransactRegInvalido	Tipo de registro a ser gravado inválido.
3001	ErroTransactCheio	Registro de transações está esgotado.
3002	ErroTransactTransAberta	Tentativa de abrir nova transação com transação já aberta.
3003	ErroTransactTransNaoAberta	Tentativa de fechar uma transação que não se encontrava
4000	ErroContextDrvrLeitura	Erro de leitura no dispositivo físico.
4001	ErroContextDrvrEscrita	Erro de escrita no dispositivo.
4002	ErroContextDrvrDesconectado	Dispositivo de contexto foi desconectado.
4003	ErroContextDrvrLeituraAposFim	Leitura após final do arquivo.
4004	ErroContextDrvrEscritaAposFim	Escrita após final do arquivo.
5000	ErroContextVersaoInvalida	Versão de contexto fiscal no dispositivo não foi reconhecida.
5001	ErroContextCRC	CRC do dispositivo está incorreto.
5002	ErroContextLimitesExcedidos	Tentativa de escrita fora da área de contexto.
6000	ErroRelogioInconsistente	Relógio do ECF inconsistente.
6001	ErroRelogioDataHoraInvalida	Data/hora informadas não estão consistentes.
7000	ErroPrintSemMecanismo	Nenhum mecanismo de impressão presente.
7001	ErroPrintDesconectado	Atual mecanismo de impressão está desconectado.
7002	ErroPrintCapacidadeInexistente	Mecanismo não possui capacidade suficiente para realizar esta operação.

7003	ErroPrintSemPapel	Impressora está sem papel para imprimir.
7004	ErroPrintFaltouPapel	Faltou papel durante a impressão do comando.
8000	ErroCMDForaDeSequencia	Comando fora de seqüência.
8001	ErroCMDCodigoInvalido	Código mercadoria não válido.
8002	ErroCMDDescricaoInvalida	Descrição inválida.
8003	ErroCMDQuantidadeInvalida	Quantidade não inválida.
8004	ErroCMDAliquotaInvalida	Índice da alíquota não válido.
8005	ErroCMDAliquotaNaoCarregada	Alíquota não carregada.
8006	ErroCMDValorInvalido	Valor contém caracter inválido.
8007	ErroCMDMontanteOperacao	Total da operação igual a 0 (zero).
8008	ErroCMDAliquotaIndisponivel	Alíquota não disponível para carga.
8009	ErroCMDValorAliquotaInvalido	Valor da alíquota não válido.
8010	ErroCMDTrocaSTAposFechamento	Troca de situação tributária somente após Redução Z.
8011	ErroCMDFormaPagamentoInvalida	Índice do Meio de Pagamento não válido.
8012	ErroCMDPayIndisponivel	Meio de Pagamento indisponível para carga.
8013	ErroCMDCupomTotalizadoEmZero	Cupom totalizado em 0 (zero).
8014	ErroCMDFormaPagamentoIndefinida	Meio de Pagamento não definido.
8015	ErroCMDTrocaUsuarioAposFechamento	Carga de usuário permitido somente após Redução Z.
8016	ErroCMDSemMovimento	Dia sem movimento.
8017	ErroCMDPagamentoIncompleto	Total pago inferior ao total do cupom.
8018	ErroCMDGerencialNaoDefinido	Gerencial não definido.
8019	ErroCMDGerencialInvalido	Índice do Gerencial fora da faixa.
8020	ErroCMDGerencialIndisponivel	Gerencial não disponível para carga.
8021	ErroCMDNomeGerencialInvalido	Nome do Gerencial inválido.
8022	ErroCMDNaoHaMaisRelatoriosLivres	Esgotado número de Gerenciais.
8023	ErroCMDAcertoHVPermitidoAposZ	Acerto do horário de verão somente após a Redução Z.
8024	ErroCMDHorarioVeraoJaRealizado	Já acertou horário de verão.
8025	ErroCMDAliquotasIndisponiveis	Sem Alíquotas disponíveis para carga.
8026	ErroCMDItemInexistente	Item não vendido no cupom.
8027	ErroCMDQtdCancInvalida	Quantidade a ser cancelada maior do que a quantidade vendida.
8028	ErroCMDCampoCabeçalhoInvalido	Cabeçalho possui campo(s) inválido(s).
8029	ErroCMDNomeDepartamentoInvalido	Nome do Departamento não válido.
8030	ErroCMDDepartamentoNaoEncontrado	Departamento não encontrado.
8031	ErroCMDDepartamentoIndefinido	Departamento não definido.
8032	ErroCMDFormasPagamentosIndisponiveis	Não há Meio de Pagamento disponível.
8033	ErroCMDAltPagamentoSoAposZ	Alteração de Meio de Pagamento somente após a Redução Z.
8034	ErroCMDNomeNaoFiscalInvalido	Nome do Documento Não Fiscal não pode ser vazio.
8035	ErroCMDDocsNaoFiscaisIndisponiveis	Não há mais Documentos Não Fiscais disponíveis.

8036	ErroCMDNaoFiscalIndisponivel	Documento Não Fiscal indisponível para carga.
8037	ErroCMDReducaoInvalida	Número da redução inicial inválida.
8038	ErroCMDCabecalhoJaImpresso	Cabeçalho do documento já foi impresso.
8039	ErroCMDLinhasSuplementaresExcedidas	Número máximo de linhas de propaganda excedidas.
8040	ErroCMDHorarioVeraoJaAtualizado	Relógio já está no estado desejado.
8041	ErroCMDValorAcrescimoInvalido	Valor do acréscimo inconsistente.
8042	ErroCMDNaoHaMeiodePagamento	Não há meio de pagamento definido.
8043	ErroCMDCOOVinculadoInvalido	COO do documento vinculado inválido.
8044	ErroCMDIndiceItemInvalido	Índice do item inexistente no contexto.
8045	ErroCMDCodigoNaoEncontrado	Código de item não encontrado no cupom atual.
8046	ErroCMDPercentualDescontoInvalido	Percentual do desconto ultrapassou 100%.
8047	ErroCMDDescontoItemInvalido	Desconto do item inválido.
8048	ErroCMDFaltaDefinirValor	Falta definir valor percentual ou absoluto em operação de desconto/acrécimo.
8049	ErroCMDItemCancelado	Tentativa de operação sobre item cancelado.
8050	ErroCMDCancelaAcrDescInvalido	Cancelamento de acréscimo/desconto inválidos.
8051	ErroCMDAcrDescInvalido	Operação de acréscimo/desconto inválida.
8052	ErroCMDNaoHaMaisDepartamentosLivres	Número de Departamentos esgotados.
8053	ErroCMDIndiceNaoFiscalInvalido	Índice de Documento Não Fiscal fora da faixa.
8054	ErroCMDTrocaNaoFiscalAposZ	Troca de Documento Não Fiscal somente após a Redução Z.
8055	ErroCMDInscricaoInvalida	CNPJ e/ou Inscrição Estadual inválida(s).
8056	ErroCMDVinculadoParametrosInsuficientes	Falta(m) parâmetro(s) no comando de abertura de Comprovante Crédito ou Débito.
8057	ErroCMDNaoFiscalIndefinido	Código e Nome do Documento Não Fiscal indefinidos.
8058	ErroCMDFaltaAliquotaVenda	Alíquota não definida no comando de venda.
8059	ErroCMDFaltaMeioPagamento	Código e Nome do Meio de Pagamento não definidos.
8060	ErroCMDFaltaParametro	Parâmetro de comando não informado.
8061	ErroCMDNaoHaDocNaoFiscaisDefinidos	Não há Documentos Não Fiscais definidos.
8062	ErroCMDOperacaoJaCancelada	Acrécimo/Desconto de item já cancelado.
8063	ErroCMDNaoHaAcrescDescItem	Não há acréscimo/desconto em item.
8064	ErroCMDItemAcrescido	Item já possui acréscimo.
8065	ErroCMDOperSoEmICMS	Operação de acréscimo em item ou subtotal só é válido para ICMS
8066	ErroCMDFaltaInformarValor	Valor do Comprovante Crédito ou Débito não informado.
8067	ErroCMDCOOInvalido	COO inválido.
8068	ErroCMDIndiceInvalido	Índice do Meio de Pagamento no cupom inválido.
8069	ErroCMDCupomNaoEncontrado	Documento Não Fiscal não encontrado.
8070	ErroCMDSequenciaPagamentoNaoEncontrada	Seqüência de pagamento não encontrada no cupom.
8071	ErroCMDPagamentoNaoPermiteCDC	Meio de pagamento não permite CDC.
8072	ErroCMDUltimaFormaPagamentoInv	Valor insuficiente para pagar o cupom.
8073	ErroCMDMeioPagamentoNEncontrado	Meio de pagamento origem ou destino não encontrado no último cupom emitido
8074	ErroCMDValorEstornoInvalido	Valor do estorno não pode exceder o valor do pagamento no meio origem.

8075	ErroCMDMeiosPagamentoOrigemDestinoIguais	Meios de pagamento origem e destino devem ser diferentes no estorno.
8076	ErroCMDPercentualInvalido	Percentual da alíquota inválido.
8077	ErroCMDNaoHouveOpSubtotal	Não houve operação em subtotal para ser cancelada.
8078	ErroCMDOpSubtotalInvalida	Só é permitida uma operação de acréscimo em subtotal por cupom.
8079	ErroCMDTextoAdicional	Texto adicional do meio de pagamento deve ter no máximo 2 linhas
8080	ErroCMDPrecoUnitarioInvalido	Preço unitário ultrapassou o número máximo de dígitos permitido
8081	ErroCMDDepartamentoInvalido	Código do departamento fora da faixa.
8082	ErroCMDDescontoInvalido	O valor do desconto não pode zerar o valor do cupom ou ser maior que o item.
8083	ErroCMDPercentualAcrescimoInvalido	Percentual de acréscimo não pode ser superior a 999,99%
8084	ErroCMDAcrescimoInvalido	Valor do acréscimo ultrapassa o número máximo de dígitos permitido (13 dígitos)
8085	ErroCMDNaoHouveVendaEmICMS	Cupom sem venda em alíquota de ICMS.
8086	ErroCMDCancelamentoInvalido	Cancelamento inválido.
8087	ErroCMDCliche	Texto de clique do usuário deve ter no máximo três linhas.
8088	ErroCMDNaoHouveVendaNaoFiscal	Não houve venda de item não fiscal
8089	ErroCMDDataInvalida	A data não pode ser inferior a data do último documento emitido.
8090	ErroCMDHoraInvalida	A hora informada no comando não pode ser inferior ao horário do último documento
8091	ErroCMDEstorno	Sem função
8092	ErroCMDAcertoRelogio	Estado inválido para ajuste de relógio ou horário de verão
8093	ErroCMDCDCInvalido	A operação de CDC deve preceder as operações de estorno de meio de pagamento.
8094	ErroCMDSenhaInvalida	Senha inválida para inicialização do proprietário.
8095	ErroCMDMecanismoCheque	Erro gerado pelo mecanismo de cheques
8096	ErroFaltaIniciarDia	Comando válido somente após a abertura do dia
9000	ErroMFDNenhumCartuchoVazio	Não foi encontrado nenhum cartucho de dados vazio para ser inicializado.
9001	ErroMFDCartuchoInexistente	Cartucho com o número de série informado não foi encontrado.
9002	ErroMFDNumSerie	Número de série do ECF é inválido na inicialização.
9003	ErroMFDCartuchoDesconectado	Cartucho de MFD desconectado ou com problemas.
9004	ErroMFDEscrita	Erro de escrita no dispositivo de MFD.
9005	ErroMFDSeek	Erro na tentativa de posicionar ponteiro de leitura.
9006	ErroMFDBadBadSector	Endereço do Bad Sector informado é inválido.
9007	ErroMFDLeitura	Erro de leitura na MFD.
9008	ErroMFDLeituraAlemEOF	Tentativa de leitura além dos limites da MFD.
9009	ErroMFDEsgotada	MFD não possui mais espaço para escrita.
9010	ErroMFDLeituraInterrompida	Leitura da MFD serial é interrompida por comando diferente de LeImpressao
10000	ErroBNFEstadoInvalido	Estado inválido para registro sendo codificado.
10001	ErroBNFParametroInvalido	Inconsistência nos parâmetros lidos no Logger.
10002	ErroBNFRegistroInvalido	Registro inválido detectado no Logger.
10003	ErroBNFErroMFD	Erro interno.

11000	ErroProtParamInvalido	Parâmetro repassado ao comando é inválido.
11001	ErroProtParamSintaxe	Erro de sintaxe na lista de parâmetros.
11002	ErroProtParamValorInvalido	Valor inválido para parâmetro do comando.
11003	ErroProtParamStringInvalido	String contém seqüência de caracteres inválidos.
11004	ErroProtParamRedefinido	Parâmetro foi declarado 2 ou mais vezes na lista.
11005	ErroProtParamIndefinido	Parâmetro obrigatório ausente na lista.
11006	ErroProtComandoInexistente	Não existe o comando no protocolo.
11007	ErroProtSequenciaComando	Estado atual não permite a execução deste comando.
11008	ErroProtAborta2aVia	Sinalização indicando que comando aborta a impressão da segunda via.
11009	ErroProtSemRetorno	Sinalização indicando que comando não possui retorno.
11010	ErroProtTimeout	Tempo de execução esgotado.
11011	ErroProtNomeRegistrador	Nome de registrador inválido.
11012	ErroProtTipoRegistrador	Tipo de registrador inválido.
11013	ErroProtSomenteLeitura	Tentativa de escrita em registrador de apenas leitura.
11014	ErroProtSomenteEscrita	Tentativa de leitura em registrador de apenas escrita.
11015	ErroProtComandoDiferenteAnterior	Comando recebido diferente do anterior no buffer de recepção.
11016	ErroProtFilaCheia	Fila de comandos cheia.
11017	ErroProtIndiceRegistrador	Índice de registrador indexado fora dos limites.
11018	ErroProtNumEmissoesExcedido	Número de emissões do Logger foi excedido na Intervenção Técnica.
11019	ErroMathDivisaoPorZero	Divisão por 0 (zero) nas rotinas de BDC.
15001	ErroApenasIntTecnica	Comando aceito apenas em modo de Intervenção Técnica.
15002	ErroECFIntTecnica	Comando não pode ser executado em modo de Intervenção Técnica.
15003	ErroMFDPresente	Já existe MFD presente neste ECF.
15004	ErroSemMFD	Não existe MFD neste ECF.
15005	ErroRAMInconsistente	Memória RAM do ECF não está consistente.
15006	ErroMemoriaFiscalDesconectada	Memória fiscal não encontrada.
15007	ErroDiaFechado	Dia já fechado.
15008	ErroDiaAberto	Dia aberto.
15009	ErroZPendente	Falta reducao Z.
15010	ErroMecanismoNaoConfigurado	Mecanismo impressor não selecionado.
15011	ErroSemPapel	Sem bobina de papel na estação de documento fiscal.
15012	ErroDocumentoEncerrado	Tentativa de finalizar documento já encerrado.
15013	ErroSemSinalDTR	Não há sinal de DTR.
15014	ErroSemInscricoes	Sem inscrições do usuário no ECF.
15015	ErroSemCliche	Sem dados do proprietário no ECF.
15016	ErroEmLinha	ECF encontra-se indevidamente em linha.
15017	ErroForaDeLinha	ECF não encontra-se em linha para executar o comando.
15018	ErroMecanismoBloqueado	Mecanismo está indisponível para impressão.

### [Voltar para a função Elgin RetornoImpressora](#)

## 9 Retorno Impressora (ACK, ST1 e ST2)

Todos as funções, executadas na impressora, possuem bytes de retorno, que significam o estado atual após o envio. Cada byte é tratado da seguinte maneira:

ACK (06h ou 6d) : Byte indicativo de recebimento correto.

ST1 e ST2 : Bytes de estado da impressora.

NAK (15h ou 21d) : Byte indicativo de recebimento incorreto.

**Observação:**

- Nas **impressoras fiscais Matriciais**, os retornos de execução são obtidos, apenas, pelos **bytes ACK, ST1 e ST2**.

- Nas **impressoras fiscal MFD (somente)**, existe um outro retorno chamado **ST3 (retorno estendido)**, onde informações mais detalhadas podem ser obtidas, mas este retorno só é habilitado pela função [Elgin\\_HabilitaDesabilitaRetornoEstendidoMFD](#).

**ST1**

Byte Retorno	Mensagem da impressora	Valor de retorno
bit 7	fim de papel	128
bit 6	pouco papel	64
bit 5	erro no relógio	32
bit 4	impressora em estado de erro	16
bit 3	o primeiro dado do cmd não foi esc (1Bh)	8
bit 2	comando inexistente	4
bit 1	cupom fiscal aberto	2
bit 0	número do parâmetro do cmd inválido	1

**ST2**

Byte Retorno	Mensagem da impressora	Valor de retorno
bit 7	tipo de parâmetro de cmd inválido	128
bit 6	memória fiscal lotada	64
bit 5	erro na memória ram	32
bit 4	alíquota não programada	16
bit 3	capacidade de alíquota esgotada	8
bit 2	cancelamento não permitido	4
bit 1	cnpj/ie do proprietário não programados	2
bit 0	comando não executado	1

**Lógica de Tratamento**

Cada byte é composto de 8 bits. Cada bit, dentro do byte, tem um valor, conforme a tabela acima:

O valor recebido da impressora para **ST1** e/ou **ST2** deve ser comparado com cada bit.

As linhas abaixo são uma forma de você desenvolver um pedaço de código para saber quais bits estão "setados":

**Codificando o ST1**

```

If St1 >= 128 Then // bit 7
    St1 = St1 - 128
    sMsg = "Fim de Papel"
End If
If St1 >= 64 Then // bit 6
    St1 = St1 - 64
    sMsg = "Pouco Papel"
End If
If St1 >= 32 Then // bit 5
    St1 = St1 - 32
    sMsg = "Erro no Relógio"
End If
If St1 >= 16 Then // bit 4
    St1 = St1 - 16
    sMsg = "Impressora em Erro"
End If
If St1 >= 8 Then // bit 3
    St1 = St1 - 8
    sMsg = "Comando não iniciado com ESC"
End If
If St1 >= 4 Then // bit 2
    St1 = St1 - 4
    sMsg = "Comando Inexistente"
End If
If St1 >= 2 Then // bit 1
    St1 = St1 - 2
    sMsg = "Cupom Aberto"
End If
If St1 >= 1 Then // bit 0
    St1 = St1 - 1
    sMsg = "Número de Parâmetro(s) Inválido(s)"
End If

```

**Codificando o ST2**

```

If St2 >= 128 Then // bit 7
    St2 = St2 - 128
    sMsg = "Tipo de Parâmetro de Comando
Inválido"
End If
If St2 >= 64 Then // bit 6
    St2 = St2 - 64
    sMsg = "Memória Fiscal Lotada"
End If
If St2 >= 32 Then // bit 5
    St2 = St2 - 32
    sMsg = "Erro na Memória RAM"
End If
If St2 >= 16 Then // bit 4
    St2 = St2 - 16
    sMsg = "Alíquota Não Programada"
End If
If St2 >= 8 Then // bit 3
    St2 = St2 - 8
    sMsg = "Capacidade de Alíquotas Lotada"
End If
If St2 >= 4 Then // bit 2
    St2 = St2 - 4
    sMsg = "Cancelamento Não Permitido"
End If
If St2 >= 2 Then // bit 1
    St2 = St2 - 2
    sMsg = "CNPJ/IE do Proprietário Não
Programado"
End If
If St2 >= 1 Then // bit 0
    St2 = St2 - 1
    sMsg = "Comando Não Executado"
End If

```

O exemplo acima com **ST1 = 66**, apresentará dois erros, **pouco papel (64)** e **cupom aberto (2)**.

**10 Comprovante Não Fiscal Vinculado**

Para realizar a emissão de um Comprovante Não Fiscal Vinculado utilizando as funções da Elgin.dll devem ser executados os seguintes comandos seguindo a sequência:

Pré-Requisito: Ter sido emitido anteriormente um Cupom Fiscal cujo pagamento aceite vinculo.

1. [Elgin\\_AbreComprovanteNaoFiscalVinculado](#) ou [Elgin\\_AbreComprovanteNaoFiscalVinculadoMFD](#)
2. [Elgin\\_UsaComprovanteNaoFiscalVinculado](#)
3. [Elgin\\_FechaComprovanteNaoFiscalVinculado](#)

**11 Declaração das Funções da Elgin.dll em Delphi**

```

{=====
*****}

```

```

DECLARAÇÃO DAS FUNÇÕES DA Elgin.DLL
Ultima atualização: 23/11/2009
Elgin.dll v.1.0.0.0
*****
=====
function Elgin_AberturaDoDia( ValorCompra: string; FormaPagamento: string ): Integer; StdCall; External
'Elgin.DLL';
function Elgin_AbreComprovanteNaoFiscalVinculado( FormaPagamento: String; Valor: String; NumeroCupom:
String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_AbreComprovanteNaoFiscalVinculadoMFD(FormaPagamento, Valor, NumeroCupom, CGC,
nome, Endereco : string): Integer; StdCall; External 'Elgin.DLL'
function Elgin_AbreCupom( CGC_CPF: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_AbreCupomMFD(CGC: string; Nome: string; Endereco : string): Integer; StdCall; External
'Elgin.DLL'
function Elgin_AbrePortaSerial: Integer; StdCall; External 'Elgin.DLL';
function Elgin_AbreRecebimentoNaoFiscalMFD(CGC, Nome, Endereco : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_AbreRelatorioGerencial(): Integer; StdCall; External 'Elgin.DLL';
function Elgin_AbreRelatorioGerencialMFD(Indice : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_AcionaGaveta:Integer; StdCall; External 'Elgin.DLL'
function Elgin_AcionaGuilhotinaMFD (TipoCorte:integer): Integer; StdCall; External 'Elgin.DLL';
function Elgin_AcrescimoDescontoItemMFD (Item, AcrescimoDesconto,TipoAcrescimoDesconto,
ValorAcrescimoDesconto: string): Integer; StdCall; External 'Elgin.DLL'
function Elgin_AcrescimoDescontoSubtotalMFD( cFlag, cTipo, cValor: string): integer; StdCall; External
'Elgin.DLL'
function Elgin_AcrescimoDescontoSubtotalRecebimentoMFD( cFlag, cTipo, cValor: string ): integer; StdCall;
External 'Elgin.DLL';
function Elgin_AcrescimoItemNaoFiscalMFD(strNroItem:string; strAcrescDesc:string;
strTipoAcrescDesc:string; strValor:string): integer; StdCall; External 'Elgin.DLL';
function Elgin_Acrescimos( ValorAcrescimos: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_AlteraSimboloMoeda( SimboloMoeda: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_AtivaDesativaVendaUmaLinhaMFD( iFlag: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_AumentaDescricaoItem( Descricao: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_Autenticacao:Integer; StdCall; External 'Elgin.DLL'
function Elgin_AutenticacaoMFD(Linhas, Texto : string):Integer; StdCall; External 'Elgin.DLL'
function Elgin_CancelaAcrescimoDescontoItemMFD( cFlag, cItem: string ): integer; StdCall; External
'Elgin.DLL';
function Elgin_CancelaAcrescimoDescontoSubtotalMFD( cFlag: string): integer; StdCall; External 'Elgin.DLL'
function Elgin_CancelaAcrescimoDescontoSubtotalRecebimentoMFD( cFlag: string ): integer; StdCall; External
'Elgin.DLL';
function Elgin_CancelaAcrescimoNaoFiscalMFD(strNumeroItem: String; strAcrescDesc: String): Integer;
StdCall; External 'Elgin.DLL';
function Elgin_CancelaCupom: Integer; StdCall; External 'Elgin.DLL';
function Elgin_CancelaCupomMFD(CGC, Nome, Endereco: string): Integer; StdCall; External 'Elgin.DLL'
function Elgin_CancelaImpressaoCheque: Integer; StdCall; External 'Elgin.DLL';
function Elgin_CancelaItemAnterior: Integer; StdCall; External 'Elgin.DLL';
function Elgin_CancelaItemGenerico( NumeroItem: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_CancelaItemNaoFiscalMFD(strNroItem:string): integer; StdCall; External 'Elgin.DLL';
function Elgin_Cancelamentos( ValorCancelamentos: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_CancelaRecebimentoNaoFiscalMFD(CGC, Nome, Endereco : string): Integer; StdCall; External
'Elgin.DLL'
function Elgin_CGC_IE( CGC: String; IE: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ClicheProprietario( Cliche: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_CNPJ_IE(CNPJ: string; IE: string):Integer;stdcall; External 'Elgin.DLL';
function Elgin_CNPJMFD(CNPJ : string): Integer; StdCall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasCODABARMFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasCODE128MFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasCODE39MFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasCODE93MFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasEAN13MFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';

```

```
Function Elgin_CodigoBarrasEAN8MFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasISBNMFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasITFMFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasMSIMFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasPLESSEYMFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasUPCAMFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Elgin_CodigoBarrasUPCEMFD(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
function Elgin_ComprovantesNaoFiscaisNaoEmitidosMFD(Comprovantes : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_ConfiguraCodigoBarrasMFD (Var Altura: Integer; var Largura: Integer; var pos: Integer; var
Fonte: Integer; var Margem: Integer): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ContadorComprovantesCreditoMFD(Comprovantes : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_ContadorCupomFiscalMFD(CuponsEmitidos : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ContadoresTotalizadoresNaoFiscais( Contadores: String ) : Integer; StdCall; External 'Elgin.DLL';
function Elgin_ContadoresTotalizadoresNaoFiscaisMFD(Contadores : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_ContadorFitaDetalheMFD(ContadorFita : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ContadorOperacoesNaoFiscaisCanceladasMFD(OperacoesCanceladas : string): Integer; StdCall;
External 'Elgin.DLL'
function Elgin_ContadorRelatoriosGerenciaisMFD( Relatorios : String): Integer; StdCall; External 'Elgin.DLL';
function Elgin_CupomAdicionalMFD(): Integer; StdCall; External 'Elgin.DLL';
function Elgin_DadosIntegra( DataInicial: string; DataFinal: string ) : integer; StdCall; External 'Elgin.DLL';
function Elgin_DadosUltimaReducao( DadosReducao: String ) : Integer; StdCall; External 'Elgin.DLL';
function Elgin_DadosUltimaReducaoMFD(DadosReducao : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_DataHoraImpressora( Data: String; Hora: String ) : Integer; StdCall; External 'Elgin.DLL';
function Elgin_DataHoraReducao( Data: String; Hora: String ) : Integer; StdCall; External 'Elgin.DLL';
function Elgin_DataHoraSoftwareBasico(DataSW, HoraSW: PChar ) : Integer;stdcall; External 'Elgin.DLL';
function Elgin_DataHoraUltimoDocumentoMFD( cDataHora: string ) : integer; StdCall; External 'Elgin.DLL';
function Elgin_DataMovimento( Data: String ) : Integer; StdCall; External 'Elgin.DLL';
function Elgin_DataMovimentoUltimaReducaoMFD( cDataMovimento: String ) : Integer; StdCall; External
'Elgin.DLL';
function Elgin_Descontos( ValorDescontos: String ) : Integer; StdCall; External 'Elgin.DLL';
function Elgin_DownloadMF( Arquivo: String ) : Integer; StdCall; External 'Elgin.DLL'
function Elgin_DownloadMFD( Arquivo: String; TipoDownload: String; ParametroInicial: String;
ParametroFinal: String; UsuarioECF: String ) : Integer; StdCall; External 'Elgin.DLL'
function Elgin_EfetuaFormaPagamento( FormaPagamento: String; ValorFormaPagamento: String ) : Integer;
StdCall; External 'Elgin.DLL';
function Elgin_EfetuaFormaPagamentoDescricaoForma( FormaPagamento: string; ValorFormaPagamento:
string; DescricaoFormaPagto: string ) : integer; StdCall; External 'Elgin.DLL';
function Elgin_EfetuaFormaPagamentoMFD(FormaPagamento, ValorFormaPagamento, Parcelas,
DescricaoFormaPagto: string): Integer; StdCall; External 'Elgin.DLL'
function Elgin_EfetuaRecebimentoNaoFiscalMFD(IndiceTotalizador, ValorRecebimento : string): Integer;
StdCall; External 'Elgin.DLL';
function Elgin_EspacoEntreLinhas( Dots: Integer ) : Integer; StdCall; External 'Elgin.DLL';
function Elgin_EstornoFormasPagamento( FormaOrigem: String; FormaDestino: String; Valor: String ) : Integer;
StdCall; External 'Elgin.DLL';
function Elgin_EstornoNaoFiscalVinculadoMFD(CGC, Nome, Endereco : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_ExecutaComando(Comandos: Pchar; Parametros: PChar):integer;stdcall;External 'ELGIN.DLL';
function Elgin_ExecutaLeitura(Comandos,Parametros,Retorno:string):integer;stdcall;External 'ELGIN.DLL';
function Elgin_FechaComprovanteNaoFiscalVinculado: Integer; StdCall; External 'Elgin.DLL';
function Elgin_FechaCupom( FormaPagamento: String; AcrescimoDesconto: String; TipoAcrescimoDesconto:
String; ValorAcrescimoDesconto: String; ValorPago: String; Mensagem: String): Integer; StdCall; External
'Elgin.DLL';
function Elgin_FechaCupomResumido( FormaPagamento: String; Mensagem: String ) : Integer; StdCall; External
'Elgin.DLL';
function Elgin_FechamentoDoDia: Integer; StdCall; External 'Elgin.DLL'
function Elgin_FechaPortaSerial: Integer; StdCall; External 'Elgin.DLL'
function Elgin_FechaRecebimentoNaoFiscalMFD(Mensagem : string): Integer; StdCall; External 'Elgin.DLL'
function Elgin_FechaRelatorioGerencial: Integer; StdCall; External 'Elgin.DLL';
```

```
function Elgin_FlagsFiscais( Var Flag: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_FlagsFiscaisStr(FlagFiscal: String): Integer; StdCall; External 'Elgin.DLL';
Function Elgin_FormatoDadosMFD(ArquivoOrigem: String;
    ArquivoDestino: String;
    TipoFormato: String;
    TipoDownload: String;
    ParametroInicial: String;
    ParametroFinal: String;
    UsuarioECF: String): Integer; StdCall; External 'Elgin.DLL';
function Elgin_GrandeTotal( GrandeTotal: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_GrandeTotalUltimaReducaoMFD( cGT: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_HabilitaDesabilitaRetornoEstendidoMFD(FlagRetorno : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_IdentificaConsumidor(CNPJ_CPF: String; Nome: String; Endereco : String): Integer; StdCall;
External 'Elgin.DLL';
function Elgin_ImprimeCheque( Banco: String; Valor: String; Favorecido: String; Cidade: String; Data: String;
Mensagem: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ImprimeChequeMFD(NumeroBanco, Valor, Favorecido, Cidade, Data, Mensagem,
ImpressaoVerso, Linhas : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ImprimeConfiguracoesImpressora:Integer; StdCall; External 'Elgin.DLL'
function Elgin_ImprimeCopiaCheque: Integer; StdCall; External 'Elgin.DLL';
function Elgin_ImprimeDepartamentos: Integer; StdCall; External 'Elgin.DLL';
function Elgin_IncluiCidadeFavorecido( Cidade: String; Favorecido: String ): Integer; StdCall; External
'Elgin.DLL';
function Elgin_IniciaFechamentoCupom( AcrescimoDesconto: String; TipoAcrescimoDesconto: String;
ValorAcrescimoDesconto: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_IniciaFechamentoCupomMFD(AcrescimoDesconto,TipoAcrescimoDesconto, ValorAcrescimo,
ValorDesconto : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_IniciaFechamentoRecebimentoNaoFiscalMFD(AcrescimoDesconto,TipoAcrescimoDesconto,
ValorAcrescimo, ValorDesconto : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_InicioFimCOOsMFD( cCOOIni, cCOOFim: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_InicioFimGTsMFD( cGTInicial, cGTFinal: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_InscricaoEstadualMFD(InscricaoEstadual : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_InscricaoMunicipalMFD(InscricaoMunicipal : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_LeArquivoRetorno(sCupom: String): Integer; StdCall; External 'Elgin.DLL';
function Elgin_LeCodigoNacionalIdentificacaoECF(CNI:string): Integer; stdcall; External 'Elgin.DLL';
function Elgin_LeIndicadores( var Indicador: Integer): Integer; StdCall; External 'Elgin.DLL';
function Elgin_LeituraCheque(CodigoCMC7 : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_LeituraMemoriaFiscalData(DataInicial, DataFinal, FlagLeitura : string): Integer; StdCall;
External 'Elgin.DLL';
function Elgin_LeituraMemoriaFiscalReducao(ReducaoInicial, ReducaoFinal, FlagLeitura : string): Integer;
StdCall; External 'Elgin.DLL';
function Elgin_LeituraMemoriaFiscalSerialData(DataInicial, DataFinal, FlagLeitura : string): Integer; StdCall;
External 'Elgin.DLL';
function Elgin_LeituraMemoriaFiscalSerialReducao(ReducaoInicial, ReducaoFinal, FlagLeitura : string):
Integer; StdCall; External 'Elgin.DLL';
function Elgin_LeituraX: Integer; StdCall; External 'Elgin.DLL' ;
function Elgin_LeituraXSerial: Integer; StdCall; External 'Elgin.DLL';
function Elgin_LeNomeRelatorioGerencial(Codigo, descricao: string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_LeParametrosPAF(CNPJ, Data, Hora, NumeroSerie, NumeroECF,GrandeTotal:string)
:integer;stdcall; External 'Elgin.DLL';
function Elgin_LinhasEntreCupons( Linhas: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_MapaResumo:Integer; StdCall; External 'Elgin.DLL';
function Elgin_MapaResumoMFD:Integer; StdCall; External 'Elgin.DLL';
function Elgin_MarcaModeloTipoImpressoraMFD(Marca, Modelo, Tipo : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_MinutosEmitindoDocumentosFiscaisMFD(Minutos : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_MinutosImprimindo( Minutos: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_MinutosLigada( Minutos: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ModeloImpressora(ModeloImpressora: String): Integer; StdCall; External 'Elgin.DLL';
```

```
function Elgin_NomeiaDepartamento( Indice: Integer; Departamento: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NomeiaRelatorioGerencialMFD (Indice, Descricao : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NomeiaTotalizadorNaoSujeitoIcms( Indice: Integer; Totalizador: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroCaixa( NumeroCaixa: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroCupom( NumeroCupom: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroCuponsCancelados( NumeroCancelamentos: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroIntervencoes( NumeroIntervencoes: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroLoja( NumeroLoja: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroOperacoesNaoFiscais( NumeroOperacoes: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroReducoes( NumeroReducoes: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroSerie( NumeroSerie: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroSerieCriptografado(NumeroSerie: String): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroSerieDescriptografado(NumeroSerieCriptografado:String; NumeroSerieDescriptografado:String): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroSerieMemoriaMFD(NumeroSerieMFD : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_NumeroSubstituicoesProprietario( NumeroSubstituicoes: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_PercentualLivreMFD( cMemoriaLivre: string ): integer; StdCall; External 'Elgin.DLL';
function Elgin_ProgramaAliquota( Aliquota: String; ICMS_ISS: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ProgramaArredondamento: Integer; StdCall; External 'Elgin.DLL';
function Elgin_ProgramaBaudRate(BaudRate:PChar):Integer;stdcall; External 'Elgin.DLL';
function Elgin_ProgramaCaracterAutenticacao( Parametros: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ProgramaFormaPagamentoMFD(FormaPagto, OperacaoTef: String ) : Integer; StdCall; External 'Elgin.DLL';
function Elgin_ProgramaHorarioVerao: Integer; StdCall; External 'Elgin.DLL';
function Elgin_ProgramaIdAplicativoMFD( NomeAplicativo: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ProgramaMoedaPlural( MoedaPlural: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ProgramaMoedaSingular( MoedaSingular: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ProgramaOperador(NomeOperador:String): Integer;stdcall; External 'Elgin.DLL';
function Elgin_ProgramaTruncamento: Integer; StdCall; External 'Elgin.DLL';
function Elgin_RecebimentoNaoFiscal( IndiceTotalizador: String; Valor: String; FormaPagamento: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ReducacaoZ( Data: String; Hora: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ReducoesRestantesMFD(Reducoes : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_RegistrosTipo60: Integer; StdCall; External 'Elgin.DLL';
function Elgin_ReimpressaoNaoFiscalVinculadoMFD() : Integer; StdCall; External 'Elgin.DLL';
function Elgin_RelatorioGerencial( Texto: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_RelatorioSintegraMFD( iRelatorios: Integer;cArquivo: String; cMes: String; cAno: String; cRazaoSocial: String; cEndereco: String;cNumero: String; cComplemento: String; cBairro: String; cCidade: String; cCEP: String; cTelefone: String; cFax: String; cContato: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_RelatorioTipo60Analitico: Integer; StdCall; External 'Elgin.DLL';
function Elgin_RelatorioTipo60AnaliticoMFD: Integer; StdCall; External 'Elgin.DLL';
function Elgin_RelatorioTipo60Mestre: Integer; StdCall; External 'Elgin.DLL';
function Elgin_ResetaImpressora: Integer; StdCall; External 'Elgin.DLL';
function Elgin_RetornoAliquotas( Aliquotas: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_RetornoImpressora( var i:integer; ErrorMessage:string):integer;StdCall; External 'Elgin.DLL';
function Elgin_Sangria( Valor: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_SegundaViaNaoFiscalVinculadoMFD(): Integer; StdCall; External 'Elgin.DLL';
function Elgin_SimboloMoeda( SimboloMoeda: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_StatusEstendidoMFD( Var iStatus: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_SubTotal( SubTotal: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_SubTotalComprovanteNaoFiscalMFD( cSubTotal: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_Suprimento( Valor: String; FormaPagamento: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_TamanhoTotalMFD( cTamanhoMFD: string ): integer; StdCall; External 'Elgin.DLL';
function Elgin_TempoOperacionalMFD(TempoOperacional : string): Integer; StdCall; External 'Elgin.DLL';
```

```
function Elgin_TerminaFechamentoCupom( Mensagem: String): Integer; StdCall; External 'Elgin.DLL';
function Elgin_TerminaFechamentoCupomCodigoBarrasMFD( cMensagem: string;cTipoCodigo:
string;cCodigo: string;iAltura: integer;iLargura: integer;iPosicaoCaracteres: integer;iFonte: integer;iMargem:
integer;iCorrecaoErros: integer;iColunas: integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_TotalDiaTroco( TotalDiaTroco: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_TotalDocTroco( TotalDocTroco: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_TotalIcmsCupom(ICMS: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_TotalLivreMFD( cMemoriaLivre: string ): integer; StdCall; External 'Elgin.DLL';
function Elgin_UltimoItemVendido( NumeroItem: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_UsaComprovanteNaoFiscalVinculado( Texto: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_UsaRelatorioGerencialMFD(Texto : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ValorFormaPagamento( FormaPagamento: String; Valor: String ): Integer; StdCall; External
'Elgin.DLL';
function Elgin_ValorFormaPagamentoMFD( FormaPagamento: String; Valor: String ): Integer; StdCall; External
'Elgin.DLL';
function Elgin_ValorPagoUltimoCupom( ValorCupom: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_ValorTotalizadorNaoFiscal( Totalizador: String; Valor: String ): Integer; StdCall; External
'Elgin.DLL';
function Elgin_ValorTotalizadorNaoFiscalMFD( Totalizador: String; Valor: String ): Integer; StdCall; External
'Elgin.DLL';
function Elgin_VendaBruta( VendaBruta: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VendaLiquida( VendaLiquida: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VendeItem(Codigo: String; Descricao: String; Aliquota: String; TipoQuantidade: String;
Quantidade: String; CasasDecimais: Integer; ValorUnitario: String; TipoDesconto: String; Desconto: String):
Integer; StdCall; External 'Elgin.DLL';
function Elgin_VendeItemDepartamento(Codigo: String; Descricao: String; Aliquota: String; ValorUnitario:
String; Quantidade: String; Acrescimento: String; Desconto: String; IndiceDepartamento: String; UnidadeMedida:
String): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaAliquotasICMS( Flag: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaAliquotasIss( Flag: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaDepartamentos( Departamentos: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaEstadoGaveta( Var EstadoGaveta: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaEstadoGavetaStr( EstadoGaveta: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaEstadoImpressora( Var ACK: Integer; Var ST1: Integer; Var ST2: Integer ): Integer;
StdCall; External 'Elgin.DLL';
function Elgin_VerificaEstadoImpressoraMFD( Var ACK: Integer; Var ST1: Integer; Var ST2: Integer; Var
ST3: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaEstadoImpressoraStr(ACK: String; ST1: String; ST2: String ): Integer; StdCall; External
'Elgin.DLL';
function Elgin_VerificaFormasPagamento( Formas: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaFormasPagamentoMFD(FormasPagamento : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_VerificaImpressoraLigada: Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaIndiceAliquotasICMS( Flag: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaIndiceAliquotasIss( Flag: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaModoOperacao( Modo: string ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaRecebimentoNaoFiscal( Recebimentos: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaRecebimentoNaoFiscalMFD(Recebimentos : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_VerificaReducaoZAutomatica(var Flag:integer): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaRelatorioGerencialMFD(Relatorios : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaSensorPoucoPapelMFD( Flag: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaStatusCheque( Var StatusCheque: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaTipoImpressora( Var TipoImpressora: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaTipoImpressoraStr(TipoImpressora: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaTotalizadoresNaoFiscais( Totalizadores: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaTotalizadoresNaoFiscaisMFD(Totalizadores : string): Integer; StdCall; External
'Elgin.DLL';
function Elgin_VerificaTotalizadoresParciais( Totalizadores: String ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaTotalizadoresParciaisMFD(Totalizadores : string): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VerificaTruncamento( Flag: string ): Integer; StdCall; External 'Elgin.DLL';
```

```

function Elgin_VerificaZPendente( var Flag: Integer ): Integer; StdCall; External 'Elgin.DLL';
function Elgin_VersaoFirmware( VersaoFirmware: String ): Integer; StdCall; External 'Elgin.DLL';
function Wind_AcionaGaveta:integer; stdcall; External 'Elgin.DLL';
function Wind_AcionaGuilhotina(Modo:integer):integer; stdcall; External 'Elgin.DLL';
function Wind_AcionaGuilhotinaParcial(Modo:integer):integer; stdcall; External 'Elgin.DLL';
function Wind_AjustaLarguraPapel(LarguraPapel : integer) : Integer; stdcall; External 'Elgin.DLL';
function Wind_ConfiguraCodigoBarras(Altura, Largura, PosicaoCaracteres, Fonte, Margem : Integer) : Integer;
stdcall; External 'Elgin.DLL';
function Wind_EnviaBuffer(Buffer:string):integer; stdcall; External 'Elgin.DLL';
function Wind_EnviaBufferFormatado(Buffer:string; TipoLetra:integer; Italico:integer;Sublinhado:integer;
Expandido:integer; Enfatizado:integer):integer; stdcall;External 'Elgin.DLL';
function Wind_EnviaComando (Buffer:string; TamanhoBuffer:integer):integer; stdcall; External 'Elgin.DLL';
function Wind_ImprimeBitmap(NomeArquivo:string; TamanhoBuffer:integer):integer; stdcall; External
'Elgin.DLL';
function Wind_ImprimeBmpEspecial (NomeArquivo:string; EscalaX, EscalaY, Angulo:integer):integer; stdcall;
External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasCODABAR(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasCODE128(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasCODE39(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasCODE93(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasEAN13(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasEAN8(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasISBN(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasITF(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasMSI(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasPDF417(NivelCorrecaoErros , Altura , Largura , Colunas: Integer;
Codigo: string) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasPLESSEY(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasUPCA(Codigo : String) :Integer; stdcall; External 'Elgin.DLL';
Function Wind_ImprimeCodigoBarrasUPCE(Codigo : String) : Integer; stdcall; External 'Elgin.DLL';
function Wind_VerificaEstadoGaveta:integer; stdcall; External 'Elgin.DLL';
Function Wind_VerificaFimPapel: Integer; StdCall; External 'Elgin.DLL';
Function Wind_VerificaPoucoPapel: Integer; StdCall; External 'Elgin.DLL';
Function RFD_ConvertedaMFD(CRZ:String): Integer; StdCall; External 'Elgin.DLL';
Function RFD_ConvertedaMFDData(DataInicial: String; DataFinal: String): Integer; StdCall; External
'Elgin.DLL';

```

## 12 Declaração das Funções da Elgin.dll em VB

```

'=====
'*****
'
'          DECLARAÇÃO DAS FUNÇÕES DA Elgin.DLL
'          Última atualização: 23/11/2009
'          Elgin.dll v.1.0.0.0
'
'*****
'===== }
Declare Function Elgin_AberturaDoDia Lib "Elgin.dll" (ByVal ValorCompra As String, ByVal
FormaPagamento As String) As Integer
Declare Function Elgin_AbreComprovanteNaoFiscalVinculado Lib "Elgin.dll" (ByVal FormaPagamento As
String, ByVal Valor As String, ByVal NumeroCupom As String) As Integer
Declare Function Elgin_AbreComprovanteNaoFiscalVinculadoMFD Lib "Elgin.dll" (ByVal FormaPagamento
As String, ByVal Valor As String, ByVal NumeroCupom As String, ByVal CGC As String, ByVal Nome As
String, ByVal Endereco As String) As Integer
Declare Function Elgin_AbreCupom Lib "Elgin.dll" (ByVal CGC_CPF As String) As Integer
Declare Function Elgin_AbreCupomMFD Lib "Elgin.dll" (ByVal CGC As String, ByVal Nome As String,
ByVal Endereco As String) As Integer
Declare Function Elgin_AbrePortaSerial Lib "Elgin.dll" () As Integer
Declare Function Elgin_AbreRecebimentoNaoFiscalMFD Lib "Elgin.dll" (ByVal CGC As String, ByVal Nome
As String, ByVal Endereco As String) As Integer

```

```
Declare Function Elgin_AbreRelatorioGerencial Lib "Elgin.dll" () As Integer
Declare Function Elgin_AbreRelatorioGerencialMFD Lib "Elgin.dll" (ByVal Indice As String) As Integer
Declare Function Elgin_AcionaGaveta Lib "Elgin.dll" () As Integer
Declare Function Elgin_AcionaGuilhotinaMFD Lib "Elgin.dll" (ByVal TipoCorte As Integer) As Integer
Declare Function Elgin_AcrescimoDescontoItemMFD Lib "Elgin.dll" (ByVal Item As String, ByVal
AcrescimoDesconto As String, ByVal TipoAcrescimoDesconto As String, ByVal ValorAcrescimoDesconto As
String) As Integer
Declare Function Elgin_AcrescimoDescontoSubtotalMFD Lib "Elgin.dll" (ByVal cFlag As String, ByVal cTipo
As String, ByVal cValor As String) As Integer
Declare Function Elgin_AcrescimoDescontoSubtotalRecebimentoMFD Lib "Elgin.dll" (ByVal cFlag As String,
ByVal cTipo As String, ByVal cValor As String) As Integer
Declare Function Elgin_AcrescimoItemNaoFiscalMFD Lib "Elgin.dll" (ByVal strNroItem As String, ByVal
strAcrescDesc As String, ByVal strTipoAcrescDesc As String, ByVal strValor As String) As Integer
Declare Function Elgin_Acrescimos Lib "Elgin.dll" (ByVal ValorAcrescimos As String) As Integer
Declare Function Elgin_AlteraSimboloMoeda Lib "Elgin.dll" (ByVal SimboloMoeda As String) As Integer
Declare Function Elgin_AtivaDesativaVendaUmaLinhaMFD Lib "Elgin.dll" (ByVal iFlag As Integer) As Integer
Declare Function Elgin_AumentaDescricaoItem Lib "Elgin.dll" (ByVal Descricao As String) As Integer
Declare Function Elgin_Autenticacao Lib "Elgin.dll" () As Integer
Declare Function Elgin_AutenticacaoMFD Lib "Elgin.dll" (ByVal Linhas As String, ByVal Texto As String) As
Integer
Declare Function Elgin_CancelaAcrescimoDescontoItemMFD Lib "Elgin.dll" (ByVal cFlag As String, ByVal
cItem As String) As Integer
Declare Function Elgin_CancelaAcrescimoDescontoSubtotalMFD Lib "Elgin.dll" (ByVal cFlag As String) As
Integer
Declare Function Elgin_CancelaAcrescimoDescontoSubtotalRecebimentoMFD Lib "Elgin.dll" (ByVal cFlag As
String) As Integer
Declare Function Elgin_CancelaAcrescimoNaoFiscalMFD Lib "Elgin.dll" (ByVal strNumeroItem As String,
ByVal strAcrescDesc As String) As Integer
Declare Function Elgin_CancelaCupom Lib "Elgin.dll" () As Integer
Declare Function Elgin_CancelaCupomMFD Lib "Elgin.dll" (ByVal CGC As String, ByVal Nome As String,
ByVal Endereco As String) As Integer
Declare Function Elgin_CancelaImpressaoCheque Lib "Elgin.dll" () As Integer
Declare Function Elgin_CancelaItemAnterior Lib "Elgin.dll" () As Integer
Declare Function Elgin_CancelaItemGenerico Lib "Elgin.dll" (ByVal NumeroItem As String) As Integer
Declare Function Elgin_CancelaItemNaoFiscalMFD Lib "Elgin.dll" (ByVal strNroItem As String) As Integer
Declare Function Elgin_Cancelamentos Lib "Elgin.dll" (ByVal ValorCancelamentos As String) As Integer
Declare Function Elgin_CancelaRecebimentoNaoFiscalMFD Lib "Elgin.dll" (ByVal CGC As String, ByVal
Nome As String, ByVal Endereco As String) As Integer
Declare Function Elgin_CGC_IE Lib "Elgin.dll" (ByVal CGC As String, ByVal IE As String) As Integer
Declare Function Elgin_ClicheProprietario Lib "Elgin.dll" (ByVal Cliche As String) As Integer
Declare Function Elgin_CNPJ_IE Lib "Elgin.dll" (ByVal CNPJ As String, ByVal IE As String) As Integer
Declare Function Elgin_CNPJMFD Lib "Elgin.dll" (ByVal CNPJ As String) As Integer
Declare Function Elgin_CodigoBarrasCODABARMFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasCODE128MFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasCODE39MFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasCODE93MFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasEAN13MFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasEAN8MFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasISBNMFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasITFMFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasMSIMFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasPLESSEYMFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasUPCAMFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_CodigoBarrasUPCEMFD Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Elgin_ComprovantesNaoFiscaisNaoEmitidosMFD Lib "Elgin.dll" (ByVal Comprovantes As
String) As Integer
Declare Function Elgin_ConfiguraCodigoBarrasMFD Lib "Elgin.dll" (ByVal Altura As Integer, ByVal Largura
As Integer, ByVal pos As Integer, ByVal Fonte As Integer, ByVal Margem As Integer) As Integer
Declare Function Elgin_ContadorComprovantesCreditoMFD Lib "Elgin.dll" (ByVal Comprovantes As String)
As Integer
Declare Function Elgin_ContadorCupomFiscalMFD Lib "Elgin.dll" (ByVal CuponsEmitidos As String) As
```

```
Integer
Declare Function Elgin_ContadoresTotalizadoresNaoFiscais Lib "Elgin.dll" (ByVal Contadores As String) As Integer
Declare Function Elgin_ContadoresTotalizadoresNaoFiscaisMFD Lib "Elgin.dll" (ByVal Contadores As String) As Integer
Declare Function Elgin_ContadorFitaDetalheMFD Lib "Elgin.dll" (ByVal ContadorFita As String) As Integer
Declare Function Elgin_ContadorOperacoesNaoFiscaisCanceladasMFD Lib "Elgin.dll" (ByVal OperacoesCanceladas As String) As Integer
Declare Function Elgin_ContadorRelatoriosGerenciaisMFD Lib "Elgin.dll" (ByVal Relatorios As String) As Integer
Declare Function Elgin_CupomAdicionalMFD Lib "Elgin.dll" () As Integer
Declare Function Elgin_DadosSintegra Lib "Elgin.dll" (ByVal DataInicial As String, ByVal DataFinal As String) As Integer
Declare Function Elgin_DadosUltimaReducao Lib "Elgin.dll" (ByVal DadosReducao As String) As Integer
Declare Function Elgin_DadosUltimaReducaoMFD Lib "Elgin.dll" (ByVal DadosReducao As String) As Integer
Declare Function Elgin_DataHoraImpressora Lib "Elgin.dll" (ByVal Data As String, ByVal Hora As String) As Integer
Declare Function Elgin_DataHoraReducao Lib "Elgin.dll" (ByVal Data As String, ByVal Hora As String) As Integer
Declare Function Elgin_DataHoraSoftwareBasico Lib "Elgin.dll" (ByVal DataSW As String, ByVal HoraSW As String) As Integer
Declare Function Elgin_DataHoraUltimoDocumentoMFD Lib "Elgin.dll" (ByVal cDataHora As String) As Integer
Declare Function Elgin_DataMovimento Lib "Elgin.dll" (ByVal Data As String) As Integer
Declare Function Elgin_DataMovimentoUltimaReducaoMFD Lib "Elgin.dll" (ByVal cDataMovimento As String) As Integer
Declare Function Elgin_Descontos Lib "Elgin.dll" (ByVal ValorDescontos As String) As Integer
Declare Function Elgin_DownloadMF Lib "Elgin.dll" (ByVal Arquivo As String) As Integer
Declare Function Elgin_DownloadMFD Lib "Elgin.dll" (ByVal Arquivo As String, ByVal TipoDownload As String, ByVal ParametroInicial As String, ByVal ParametroFinal As String, ByVal UsuarioECF As String) As Integer
Declare Function Elgin_EfetuaFormaPagamento Lib "Elgin.dll" (ByVal FormaPagamento As String, ByVal ValorFormaPagamento As String) As Integer
Declare Function Elgin_EfetuaFormaPagamentoDescricaoForma Lib "Elgin.dll" (ByVal FormaPagamento As String, ByVal ValorFormaPagamento As String, ByVal DescricaoFormaPagto As String) As Integer
Declare Function Elgin_EfetuaFormaPagamentoMFD Lib "Elgin.dll" (ByVal FormaPagamento As String, ByVal ValorFormaPagamento As String, ByVal Parcelas As String, ByVal DescricaoFormaPagto As String) As Integer
Declare Function Elgin_EfetuaRecebimentoNaoFiscalMFD Lib "Elgin.dll" (ByVal IndiceTotalizador As String, ByVal ValorRecebimento As String) As Integer
Declare Function Elgin_EspacoEntreLinhas Lib "Elgin.dll" (ByVal Dots As Integer) As Integer
Declare Function Elgin_EstornoFormasPagamento Lib "Elgin.dll" (ByVal FormaOrigem As String, ByVal FormaDestino As String, ByVal Valor As String) As Integer
Declare Function Elgin_EstornoNaoFiscalVinculadoMFD Lib "Elgin.dll" (ByVal CGC As String, ByVal Nome As String, ByVal Endereco As String) As Integer
Declare Function Elgin_ExecutaComando Lib "Elgin.dll" (ByVal Comando As String, ByVal parametros As String) As Integer
Declare Function Elgin_ExecutaLeitura Lib "Elgin.dll" (ByVal Comando As String, ByVal parametros As String, ByVal retorno As String) As Integer
Declare Function Elgin_FechaComprovanteNaoFiscalVinculado Lib "Elgin.dll" () As Integer
Declare Function Elgin_FechaCupom Lib "Elgin.dll" (ByVal FormaPagamento As String, ByVal acrescimoDesconto As String, ByVal TipoAcrescimoDesconto As String, ByVal ValorAcrescimoDesconto As String, ByVal ValorPago As String, ByVal Mensagem As String) As Integer
Declare Function Elgin_FechaCupomResumido Lib "Elgin.dll" (ByVal FormaPagamento As String, ByVal Mensagem As String) As Integer
Declare Function Elgin_FechamentoDoDia Lib "Elgin.dll" () As Integer
Declare Function Elgin_FechaPortaSerial Lib "Elgin.dll" () As Integer
Declare Function Elgin_FechaRecebimentoNaoFiscalMFD Lib "Elgin.dll" (ByVal Mensagem As String) As Integer
Declare Function Elgin_FechaRelatorioGerencial Lib "Elgin.dll" () As Integer
Declare Function Elgin_FlagsFiscais Lib "Elgin.dll" (ByRef Flag As Integer) As Integer
Declare Function Elgin_FlagsFiscaisStr Lib "Elgin.dll" (ByVal FlagFiscal As String) As Integer
```

Declare Function Elgin\_FormatoDadosMFD Lib "Elgin.dll" (ByVal ArquivoOrigem As String, ByVal ArquivoDestino As String, ByVal TipoFormato As String, ByVal TipoDownload As String, ByVal ParametroInicial As String, ByVal ParametroFinal As String, ByVal UsuarioECF As String) As Integer

Declare Function Elgin\_GrandeTotal Lib "Elgin.dll" (ByVal GrandeTotal As String) As Integer

Declare Function Elgin\_GrandeTotalUltimaReducaoMFD Lib "Elgin.dll" (ByVal cGT As String) As Integer

Declare Function Elgin\_HabilitaDesabilitaRetornoEstendidoMFD Lib "Elgin.dll" (ByVal FlagRetorno As String) As Integer

Declare Function Elgin\_IdentificaConsumidor Lib "Elgin.dll" (ByVal CNPJ\_CPF As String, ByVal Nome As String, ByVal Endereco As String) As Integer

Declare Function Elgin\_ImprimeCheque Lib "Elgin.dll" (ByVal Banco As String, ByVal Valor As String, ByVal Favorecido As String, ByVal Cidade As String, ByVal Data As String, ByVal Mensagem As String) As Integer

Declare Function Elgin\_ImprimeChequeMFD Lib "Elgin.dll" (ByVal NumeroBanco As String, ByVal Valor As String, ByVal Favorecido As String, ByVal Cidade As String, ByVal Data As String, ByVal Mensagem As String, ByVal ImpressaoVerso As String, ByVal Linhas As String) As Integer

Declare Function Elgin\_ImprimeConfiguracoesImpressora Lib "Elgin.dll" () As Integer

Declare Function Elgin\_ImprimeCopiaCheque Lib "Elgin.dll" () As Integer

Declare Function Elgin\_ImprimeDepartamentos Lib "Elgin.dll" () As Integer

Declare Function Elgin\_IncluiCidadeFavorecido Lib "Elgin.dll" (ByVal Cidade As String, ByVal Favorecido As String) As Integer

Declare Function Elgin\_IniciaFechamentoCupom Lib "Elgin.dll" (ByVal AcrescimentoDesconto As String, ByVal TipoAcrescimentoDesconto As String, ByVal ValorAcrescimentoDesconto As String) As Integer

Declare Function Elgin\_IniciaFechamentoCupomMFD Lib "Elgin.dll" (ByVal AcrescimentoDesconto As String, ByVal TipoAcrescimentoDesconto As String, ByVal ValorAcrescimento As String, ByVal ValorDesconto As String) As Integer

Declare Function Elgin\_IniciaFechamentoRecebimentoNaoFiscalMFD Lib "Elgin.dll" (ByVal AcrescimentoDesconto As String, ByVal TipoAcrescimentoDesconto As String, ByVal ValorAcrescimento As String, ByVal ValorDesconto As String) As Integer

Declare Function Elgin\_InicioFimCOOsMFD Lib "Elgin.dll" (ByVal cCOOIni As String, ByVal cCOOFim As String) As Integer

Declare Function Elgin\_InicioFimGTsMFD Lib "Elgin.dll" (ByVal cGTIni, ByVal cGTFim As String) As Integer

Declare Function Elgin\_InscricaoEstadualMFD Lib "Elgin.dll" (ByVal InscricaoEstadual As String) As Integer

Declare Function Elgin\_InscricaoMunicipalMFD Lib "Elgin.dll" (ByVal InscricaoMunicipal As String) As Integer

Declare Function Elgin\_LeArquivoRetorno Lib "Elgin.dll" (ByVal sCupom As String) As Integer

Declare Function Elgin\_LeIndicadores Lib "Elgin.dll" (ByRef indicador As Integer) As Integer

Declare Function Elgin\_LeituraCheque Lib "Elgin.dll" (ByVal CodigocMC7 As String) As Integer

Declare Function Elgin\_LeituraMemoriaFiscalData Lib "Elgin.dll" (ByVal DataInicial As String, ByVal DataFinal As String, ByVal FlagLeitura As String) As Integer

Declare Function Elgin\_LeituraMemoriaFiscalReducao Lib "Elgin.dll" (ByVal ReducaoInicial As String, ByVal ReducaoFinal As String, ByVal FlagLeitura As String) As Integer

Declare Function Elgin\_LeituraMemoriaFiscalSerialData Lib "Elgin.dll" (ByVal DataInicial As String, ByVal DataFinal As String, ByVal FlagLeitura As String) As Integer

Declare Function Elgin\_LeituraMemoriaFiscalSerialReducao Lib "Elgin.dll" (ByVal ReducaoInicial As String, ByVal ReducaoFinal As String, ByVal FlagLeitura As String) As Integer

Declare Function Elgin\_LeituraX Lib "Elgin.dll" () As Integer

Declare Function Elgin\_LeituraXSerial Lib "Elgin.dll" () As Integer

Declare Function Elgin\_LeCodigoNacionalIdentificacaoECF Lib "Elgin.dll" (ByVal CNI As String) As Integer

Declare Function Elgin\_LeNomeRelatorioGerencial Lib "Elgin.dll" (ByVal Codigoc As String, ByVal NomeRelatorio As String) As Integer

Declare Function Elgin\_LeParametrosPAF Lib "Elgin.dll" (ByVal CNPJ As String, ByVal Data As String, ByVal Hora As String, ByVal NumeroSerie As String, ByVal NumeroECF As String, ByVal GrandeTotal As String) As Integer

Declare Function Elgin\_LinhasEntreCupons Lib "Elgin.dll" (ByVal Linhas As Integer) As Integer

Declare Function Elgin\_MapasResumo Lib "Elgin.dll" () As Integer

Declare Function Elgin\_MapasResumoMFD Lib "Elgin.dll" () As Integer

Declare Function Elgin\_MarcaModeloTipoImpressoraMFD Lib "Elgin.dll" (ByVal Marca As String, ByVal Modelo As String, ByVal Tipo As String) As Integer

Declare Function Elgin\_MinutosEmitindoDocumentosFiscaisMFD Lib "Elgin.dll" (ByVal Minutos As String) As Integer

Declare Function Elgin\_MinutosImprimindo Lib "Elgin.dll" (ByVal Minutos As String) As Integer

```
Declare Function Elgin_MinutosLigada Lib "Elgin.dll" (ByVal Minutos As String) As Integer
Declare Function Elgin_ModeloImpressora Lib "Elgin.dll" (ByVal ModeloImpressora As String) As Integer
Declare Function Elgin_NomeiaDepartamento Lib "Elgin.dll" (ByVal Indice As Integer, ByVal Departamento
As String) As Integer
Declare Function Elgin_NomeiaRelatorioGerencialMFD Lib "Elgin.dll" (ByVal Indice As String, ByVal
Descricao As String) As Integer
Declare Function Elgin_NomeiaTotalizadorNaoSujeitoIcms Lib "Elgin.dll" (ByVal Indice As Integer, ByVal
Totalizador As String) As Integer
Declare Function Elgin_NumeroCaixa Lib "Elgin.dll" (ByVal NumeroCaixa As String) As Integer
Declare Function Elgin_NumeroCupom Lib "Elgin.dll" (ByVal NumeroCupom As String) As Integer
Declare Function Elgin_NumeroCuponsCancelados Lib "Elgin.dll" (ByVal NumeroCancelamentos As String) As
Integer
Declare Function Elgin_NumeroIntervencoes Lib "Elgin.dll" (ByVal NumeroIntervencoes As String) As Integer
Declare Function Elgin_NumeroLoja Lib "Elgin.dll" (ByVal NumeroLoja As String) As Integer
Declare Function Elgin_NumeroOperacoesNaoFiscais Lib "Elgin.dll" (ByVal NumeroOperacoes As String) As
Integer
Declare Function Elgin_NumeroReducoes Lib "Elgin.dll" (ByVal NumeroReducoes As String) As Integer
Declare Function Elgin_NumeroSerie Lib "Elgin.dll" (ByVal NumeroSerie As String) As Integer
Declare Function Elgin_NumeroSerieCriptografado Lib "Elgin.dll" (ByVal NumeroSerie As String) As Integer
Declare Function Elgin_NumeroSerieDescriptografado Lib "Elgin.dll" (ByVal NumeroSerieCriptografado As
String, ByVal NumeroSerieDescriptografado As String) As Integer
Declare Function Elgin_NumeroSerieMemoriaMFD Lib "Elgin.dll" (ByVal NumeroSerieMFD As String) As
Integer
Declare Function Elgin_NumeroSubstituicoesProprietario Lib "Elgin.dll" (ByVal NumeroSubstituicoes As
String) As Integer
Declare Function Elgin_PercentualLivreMFD Lib "Elgin.dll" (ByVal cMemoriaLivre As String) As Integer
Declare Function Elgin_ProgramaAliquota Lib "Elgin.dll" (ByVal Aliquota As String, ByVal ICMS_ISS As
Integer) As Integer
Declare Function Elgin_ProgramaArredondamento Lib "Elgin.dll" () As Integer
Declare Function Elgin_ProgramaBaudRate Lib "Elgin.dll" (ByVal BaudRate As String) As Integer
Declare Function Elgin_ProgramaCaracterAutenticacao Lib "Elgin.dll" (ByVal parametros As String) As Integer
Declare Function Elgin_ProgramaFormaPagamentoMFD Lib "Elgin.dll" (ByVal FormaPagto As String, ByVal
OperacaoTef As String) As Integer
Declare Function Elgin_ProgramaHorarioVerao Lib "Elgin.dll" () As Integer
Declare Function Elgin_ProgramaIdAplicativoMFD Lib "Elgin.dll" (ByVal NomeAplicativo As String) As
Integer
Declare Function Elgin_ProgramaMoedaPlural Lib "Elgin.dll" (ByVal MoedaPlural As String) As Integer
Declare Function Elgin_ProgramaMoedaSingular Lib "Elgin.dll" (ByVal MoedaSingular As String) As Integer
Declare Function Elgin_ProgramaOperador Lib "Elgin.dll" (ByVal NomeOperador As String) As Integer
Declare Function Elgin_ProgramaTruncamento Lib "Elgin.dll" () As Integer
Declare Function Elgin_RecebimentoNaoFiscal Lib "Elgin.dll" (ByVal IndiceTotalizador As String, ByVal
Valor As String, ByVal FormaPagamento As String) As Integer
Declare Function Elgin_ReducacaoZ Lib "Elgin.dll" (ByVal Data As String, ByVal Hora As String) As Integer
Declare Function Elgin_ReducoesRestantesMFD Lib "Elgin.dll" (ByVal Reducoes As String) As Integer
Declare Function Elgin_RegistrosTipo60 Lib "Elgin.dll" () As Integer
Declare Function Elgin_ReimpressaoNaoFiscalVinculadoMFD Lib "Elgin.dll" () As Integer
Declare Function Elgin_RelatorioGerencial Lib "Elgin.dll" (ByVal Texto As String) As Integer
Declare Function Elgin_RelatorioSintegraMFD Lib "Elgin.dll" (ByVal iRelatorios As Integer, ByVal cArquivo
As String, ByVal cMes As String, ByVal cAno As String, ByVal cRazaoSocial As String, ByVal cEndereco As
String, ByVal cNumero As String, ByVal cComplemento As String, ByVal cBairro As String, ByVal cCidade As
String, ByVal cCEP As String, ByVal cTelefone As String, ByVal cFax As String, ByVal cContato As String)
As Integer
Declare Function Elgin_RelatorioTipo60Analitico Lib "Elgin.dll" () As Integer
Declare Function Elgin_RelatorioTipo60AnaliticoMFD Lib "Elgin.dll" () As Integer
Declare Function Elgin_RelatorioTipo60Mestre Lib "Elgin.dll" () As Integer
Declare Function Elgin_ResetaImpressora Lib "Elgin.dll" () As Integer
Declare Function Elgin_RetornoAliquotas Lib "Elgin.dll" (ByVal Aliquotas As String) As Integer
Declare Function Elgin_RetornoImpressora Lib "Elgin.dll" (ByRef i As Integer, ByVal ErrorMessage As String) As
Integer
Declare Function Elgin_Sangria Lib "Elgin.dll" (ByVal Valor As String) As Integer
Declare Function Elgin_SegundaViaNaoFiscalVinculadoMFD Lib "Elgin.dll" () As Integer
```

```
Declare Function Elgin_SimboloMoeda Lib "Elgin.dll" (ByVal SimboloMoeda As String) As Integer
Declare Function Elgin_StatusEstendidoMFD Lib "Elgin.dll" (ByRef iStatus As Integer) As Integer
Declare Function Elgin_SubTotal Lib "Elgin.dll" (ByVal SubTotal As String) As Integer
Declare Function Elgin_SubTotalComprovanteNaoFiscalMFD Lib "Elgin.dll" (ByVal cSubTotal As String) As Integer
Declare Function Elgin_Suprimto Lib "Elgin.dll" (ByVal Valor As String, ByVal FormaPagamento As String) As Integer
Declare Function Elgin_TamanhoTotalMFD Lib "Elgin.dll" (ByVal cTamanhoMFD As String) As Integer
Declare Function Elgin_TempoOperacionalMFD Lib "Elgin.dll" (ByVal TempoOperacional As String) As Integer
Declare Function Elgin_TerminaFechamentoCupom Lib "Elgin.dll" (ByVal Mensagem As String) As Integer
Declare Function Elgin_TerminaFechamentoCupomCodigoBarrasMFD Lib "Elgin.dll" (ByVal cMensagem As String, ByVal cTipoCodigo As String, ByVal cCodigo As String, ByVal iAltura As Integer, ByVal iLargura As Integer, ByVal iPosicaoCaracteres As Integer, ByVal iFonte As Integer, ByVal iMargem As Integer, ByVal iCorrecaoErros As Integer, ByVal iColunas As Integer) As Integer
Declare Function Elgin_TotalDiaTroco Lib "Elgin.dll" (ByVal TotalDiaTroco As String) As Integer
Declare Function Elgin_TotalDocTroco Lib "Elgin.dll" (ByVal TotalDocTroco As String) As Integer
Declare Function Elgin_TotalIcmsCupom Lib "Elgin.dll" (ByVal TotalICMS As String) As Integer
Declare Function Elgin_TotalLivreMFD Lib "Elgin.dll" (ByVal cMemoriaLivre As String) As Integer
Declare Function Elgin_UltimoItemVendido Lib "Elgin.dll" (ByVal NumeroItem As String) As Integer
Declare Function Elgin_UsaComprovanteNaoFiscalVinculado Lib "Elgin.dll" (ByVal Texto As String) As Integer
Declare Function Elgin_UsaRelatorioGerencialMFD Lib "Elgin.dll" (ByVal Texto As String) As Integer
Declare Function Elgin_ValorFormaPagamento Lib "Elgin.dll" (ByVal FormaPagamento As String, ByVal Valor As String) As Integer
Declare Function Elgin_ValorFormaPagamentoMFD Lib "Elgin.dll" (ByVal FormaPagamento As String, ByVal Valor As String) As Integer
Declare Function Elgin_ValorPagoUltimoCupom Lib "Elgin.dll" (ByVal ValorCupom As String) As Integer
Declare Function Elgin_ValorTotalizadorNaoFiscal Lib "Elgin.dll" (ByVal Totalizador As String, ByVal Valor As String) As Integer
Declare Function Elgin_ValorTotalizadorNaoFiscalMFD Lib "Elgin.dll" (ByVal Totalizador As String, ByVal Valor As String) As Integer
Declare Function Elgin_VendaBruta Lib "Elgin.dll" (ByVal VendaBruta As String) As Integer
Declare Function Elgin_VendaLiquida Lib "Elgin.dll" (ByVal VendaLiquida As String) As Integer
Declare Function Elgin_VendeItem Lib "Elgin.dll" (ByVal Codigo As String, ByVal Descricao As String, ByVal Aliquota As String, ByVal TipoQuantidade As String, ByVal Quantidade As String, ByVal CasasDecimais As Integer, ByVal ValorUnitario As String, ByVal TipoDesconto As String, ByVal Desconto As String) As Integer
Declare Function Elgin_VendeItemDepartamento Lib "Elgin.dll" (ByVal Codigo As String, ByVal Descricao As String, ByVal Aliquota As String, ByVal ValorUnitario As String, ByVal Quantidade As String, ByVal acrescimo As String, ByVal Desconto As String, ByVal IndiceDepartamento As String, ByVal UnidadeMedida As String) As Integer
Declare Function Elgin_VerificaAliquotasICMS Lib "Elgin.dll" (ByVal Flag As String) As Integer
Declare Function Elgin_VerificaAliquotasIss Lib "Elgin.dll" (ByVal Flag As String) As Integer
Declare Function Elgin_VerificaDepartamentos Lib "Elgin.dll" (ByVal Departamentos As String) As Integer
Declare Function Elgin_VerificaEstadoGaveta Lib "Elgin.dll" (ByRef EstadoGaveta As Integer) As Integer
Declare Function Elgin_VerificaEstadoGavetaStr Lib "Elgin.dll" (ByVal EstadoGaveta As String) As Integer
Declare Function Elgin_VerificaEstadoImpressora Lib "Elgin.dll" (ByRef ACK As Integer, ByRef ST1 As Integer, ByRef ST2 As Integer) As Integer
Declare Function Elgin_VerificaEstadoImpressoraMFD Lib "Elgin.dll" (ByRef ACK As Integer, ByRef ST1 As Integer, ByRef ST2 As Integer, ByRef ST3 As Integer) As Integer
Declare Function Elgin_VerificaEstadoImpressoraStr Lib "Elgin.dll" (ByVal ACK As String, ByVal ST1 As String, ByVal ST2 As String) As Integer
Declare Function Elgin_VerificaFormasPagamento Lib "Elgin.dll" (ByVal Formas As String) As Integer
Declare Function Elgin_VerificaFormasPagamentoMFD Lib "Elgin.dll" (ByVal FormasPagamento As String) As Integer
Declare Function Elgin_VerificaImpressoraLigada Lib "Elgin.dll" () As Integer
Declare Function Elgin_VerificaIndiceAliquotasICMS Lib "Elgin.dll" (ByVal Flag As String) As Integer
Declare Function Elgin_VerificaIndiceAliquotasIss Lib "Elgin.dll" (ByVal Flag As String) As Integer
Declare Function Elgin_VerificaModoOperacao Lib "Elgin.dll" (ByVal Modo As String) As Integer
Declare Function Elgin_VerificaRecebimentoNaoFiscal Lib "Elgin.dll" (ByVal Recebimentos As String) As Integer
```

```
Declare Function Elgin_VerificaRecebimentoNaoFiscalMFD Lib "Elgin.dll" (ByVal Recebimentos As String)
As Integer
Declare Function Elgin_VerificaReducaoZAutomatica Lib "Elgin.dll" (ByRef Flag As Integer) As Integer
Declare Function Elgin_VerificaRelatorioGerencialMFD Lib "Elgin.dll" (ByVal Relatorios As String) As Integer
Declare Function Elgin_VerificaSensorPoucoPapelMFD Lib "Elgin.dll" (ByVal Flag As String) As Integer
Declare Function Elgin_VerificaStatusCheque Lib "Elgin.dll" (ByRef StatusCheque As Integer) As Integer
Declare Function Elgin_VerificaTipoImpressora Lib "Elgin.dll" (ByRef TipoImpressora As Integer) As Integer
Declare Function Elgin_VerificaTipoImpressoraStr Lib "Elgin.dll" (ByVal TipoImpressora As String) As Integer
Declare Function Elgin_VerificaTotalizadoresNaoFiscais Lib "Elgin.dll" (ByVal Totalizadores As String) As
Integer
Declare Function Elgin_VerificaTotalizadoresNaoFiscaisMFD Lib "Elgin.dll" (ByVal Totalizadores As String)
As Integer
Declare Function Elgin_VerificaTotalizadoresParciais Lib "Elgin.dll" (ByVal Totalizadores As String) As
Integer
Declare Function Elgin_VerificaTotalizadoresParciaisMFD Lib "Elgin.dll" (ByVal Totalizadores As String) As
Integer
Declare Function Elgin_VerificaTruncamento Lib "Elgin.dll" (ByVal Flag As String) As Integer
Declare Function Elgin_VerificaZPendente Lib "Elgin.dll" (ByRef Flag As Integer) As Integer
Declare Function Elgin_VersaoFirmware Lib "Elgin.dll" (ByVal VersaoFirmware As String) As Integer
Declare Function RFD_ConvertedaMFD Lib "Elgin.dll" (ByVal CRZ As String) As Integer
Declare Function RFD_ConvertedaMFDData Lib "Elgin.dll" (ByVal DataInicial As String, ByVal DataFinal As
String) As Integer
Declare Function Wind_AcionaGaveta Lib "Elgin.dll" () As Integer
Declare Function Wind_AcionaGuilhotina Lib "Elgin.dll" (ByVal Modo As Integer) As Integer
Declare Function Wind_AcionaGuilhotinaParcial Lib "Elgin.dll" (ByVal Modo As Integer) As Integer
Declare Function Wind_AjustaLarguraPapel Lib "Elgin.dll" (ByVal LarguraPapel As Integer) As Integer
Declare Function Wind_ConfiguraCodigoBarras Lib "Elgin.dll" (ByVal Altura As Integer, ByVal Largura As
Integer, ByVal PosicaoCaracteres As Integer, ByVal Fonte As Integer, ByVal Margem As Integer) As Integer
Declare Function Wind_EnviaBuffer Lib "Elgin.dll" (ByVal Buffer As String) As Integer
Declare Function Wind_EnviaBufferFormatado Lib "Elgin.dll" (ByVal Buffer As String, ByVal TipoLetra As
Integer, ByVal Italico As Integer, ByVal Sublinhado As Integer, ByVal Expandido As Integer, ByVal Enfatizado
As Integer) As Integer
Declare Function Wind_EnviaComando Lib "Elgin.dll" (ByVal Buffer As String, ByVal TamanhoBuffer As
Integer) As Integer
Declare Function Wind_ImprimeBitmap Lib "Elgin.dll" (ByVal NomeArquivo As String, ByVal Modo As
Integer) As Integer
Declare Function Wind_ImprimeBmpEspecial Lib "Elgin.dll" (ByVal NomeArquivo As String, ByVal EscalaX
As Integer, ByVal EscalaY As Integer, ByVal Angulo As Integer) As Integer
Declare Function Wind_ImprimeCodigoBarrasCODABAR Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasCODE128 Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasCODE39 Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasCODE93 Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasEAN13 Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasEAN8 Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasISBN Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasITF Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasMSI Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasPDF417 Lib "Elgin.dll" (ByVal NivelCorrecaoErros As Integer,
ByVal Altura As Integer, ByVal Largura As Integer, ByVal Colunas As Integer, ByVal Codigo As String) As
Integer
Declare Function Wind_ImprimeCodigoBarrasPLESSEY Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasUPCA Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_ImprimeCodigoBarrasUPCE Lib "Elgin.dll" (ByVal Codigo As String) As Integer
Declare Function Wind_VerificaEstadoGaveta Lib "Elgin.dll" () As Integer
Declare Function Wind_VerificaFimPapel Lib "Elgin.dll" () As Integer
Declare Function Wind_VerificaPoucoPapel Lib "Elgin.dll" () As Integer
```