

# MANUAL DA INTERFACE EPSON

Ver. 4.9.0

## INTERFACE DE ALTO NÍVEL PARA IMPRESSORAS FISCAIS EPSON



PROGRAMA  
EPSON  
PARCERIA  
DE SOFTWARE

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



**Epson Stars**

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

0800 7768 6668

## Índice

<b>1</b>	<b>CONVENÇÕES .....</b>	<b>7</b>
	CONVENÇÃO DE SÍMBOLOS.....	7
	TIPOS DE DADOS SUPOSTADOS .....	7
<b>2</b>	<b>INTRODUÇÃO .....</b>	<b>7</b>
<b>3</b>	<b>RETORNOS DAS FUNÇÕES.....</b>	<b>8</b>
<b>4</b>	<b>FUNÇÕES DA INTERFACE.....</b>	<b>8</b>
4.1	GRUPO DA PORTA SERIAL.....	9
4.1.1	EPSON_Serial_Abrir_Porta .....	9
4.1.2	EPSON_Serial_Abrir_Fechar_Porta_CMD.....	11
4.1.3	EPSON_Serial_Abrir_PortaAD .....	13
4.1.4	EPSON_Serial_Abrir_PortaEX.....	15
4.1.5	EPSON_Serial_Fechar_Porta.....	16
4.1.6	EPSON_Serial_Obter_Estado_Com.....	17
4.1.7	EPSON_Serial_Config_Simplificada.....	18
4.2	GRUPO DO CUPOM FISCAL.....	19
4.2.1	EPSON_Fiscal_Abrir_Cupom.....	19
4.2.2	EPSON_Fiscal_Vender_Item .....	21
4.2.3	EPSON_Fiscal_Vender_Item_AD.....	23
4.2.4	EPSON_Fiscal_Obter_SubTotal.....	25
4.2.5	EPSON_Fiscal_Pagamento .....	27
4.2.6	EPSON_Fiscal_Desconto_Acrescimo_Item.....	28
4.2.7	EPSON_Fiscal_Desconto_Acrescimo_ItemEX.....	29
4.2.8	EPSON_Fiscal_Desconto_Acrescimo_Subtotal.....	31
4.2.9	EPSON_Fiscal_Cancelar_Cupom.....	32
4.2.10	EPSON_Fiscal_Cancelar_CupomEX .....	33
4.2.11	EPSON_Fiscal_Cancelar_Item .....	34
4.2.12	EPSON_Fiscal_Cancelar_Ultimo_Item.....	35
4.2.13	EPSON_Fiscal_Cancelar_Desconto_Acrescimo_Item.....	36
4.2.14	EPSON_Fiscal_Cancelar_Desconto_Acrescimo_ItemEX.....	37
4.2.15	EPSON_Fiscal_Cancelar_Acrescimo_Desconto_Subtotal.....	38
4.2.16	EPSON_Fiscal_Cancelar_Item_Parcial.....	39
4.2.17	EPSON_Fiscal_Imprimir_Mensagem.....	40
4.2.18	EPSON_Fiscal_Imprimir_MensagemEX .....	42
4.2.19	EPSON_Fiscal_Configurar_Codigo_Barras_Mensagem.....	44
4.2.20	EPSON_Fiscal_Fechar_CupomEX.....	48
4.2.21	EPSON_Fiscal_Fechar_Cupom .....	49
4.2.22	EPSON_Fiscal_Dados_Consumidor .....	50
4.2.23	EPSON_ESC_ECF_Fiscal_Abrir_Cupom.....	51
4.2.24	EPSON_ESC_ECF_Fiscal_Abrir_CupomEX .....	53
4.2.25	EPSON_ESC_ECF_Fiscal_Vender_Item.....	54
4.2.26	EPSON_ESC_ECF_Fiscal_Vender_ItemEX.....	56
4.2.27	EPSON_ESC_ECF_Fiscal_Cancelar_Item .....	58
4.2.28	EPSON_ESC_ECF_Fiscal_Cancelar_ItemEX.....	60
4.2.29	EPSON_ESC_ECF_Fiscal_Pagamento.....	61
4.2.30	EPSON_ESC_ECF_Fiscal_PagamentoEX .....	63
4.2.31	EPSON_ESC_ECF_Fiscal_Fechar_Cupom .....	65
4.2.32	EPSON_ESC_ECF_Fiscal_Fechar_CupomEX.....	67
4.2.33	EPSON_ESC_ECF_Fiscal_Cancelar_Cupom.....	68
4.2.34	EPSON_ESC_ECF_Desconto_Acrescimo_Item .....	69
4.2.35	EPSON_ESC_ECF_Desconto_Acrescimo_ItemEX.....	71
4.2.36	EPSON_ESC_ECF_Cancela_Desconto_Acrescimo_Item.....	73
4.2.37	EPSON_ESC_ECF_Cancela_Desconto_Acrescimo_ItemEX.....	75
4.2.38	EPSON_ESC_ECF_Desconto_Acrescimo_Subtotal .....	76

4.2.39	EPSON_ESC_ECF_Desconto_Acrescimo_SubtotalEX.....	78
4.2.40	EPSON_ESC_ECF_Cancela_Desconto_Acrescimo_Subtotal.....	79
4.2.41	EPSON_ESC_ECF_Cancela_Desconto_Acrescimo_SubtotalEX.....	81
4.2.42	EPSON_ESC_ECF_Cancela_Cupom_Fiscal_Nao_Fiscal.....	82
4.2.43	EPSON_ESC_ECF_Cancela_Item_Parcial.....	83
4.2.44	EPSON_ESC_ECF_Cancela_Item_ParcialEX.....	85
4.2.45	EPSON_ESC_ECF_Fiscal_Abrir_Cupom_CFE.....	86
4.2.46	EPSON_ESC_ECF_Fiscal_Vender_Item_Detalhado .....	87
GRUPO DO COMPROVANTE NÃO-FISCAL .....		89
4.2.47	EPSON_NaoFiscal_Abrir_Comprovante.....	89
4.2.48	EPSON_NaoFiscal_Vender_Item.....	91
4.2.49	EPSON_NaoFiscal_Desconto_Acrescimo_Item .....	92
4.2.50	EPSON_NaoFiscal_Desconto_Acrescimo_ItemEX.....	93
4.2.51	EPSON_NaoFiscal_Desconto_Acrescimo_Subtotal .....	95
4.2.52	EPSON_NaoFiscal_Pagamento .....	96
4.2.53	EPSON_NaoFiscal_Cancelar_Item.....	97
4.2.54	EPSON_NaoFiscal_Cancelar_Ultimo_Item .....	98
4.2.55	EPSON_NaoFiscal_Cancelar_Desconto_Acrescimo_Item .....	99
4.2.56	EPSON_NaoFiscal_Cancelar_Desconto_Acrescimo_ItemEX.....	100
4.2.57	EPSON_NaoFiscal_Cancelar_Desconto_Acrescimo_Subtotal .....	101
4.2.58	EPSON_NaoFiscal_Cancelar_Comprovante.....	102
4.2.59	EPSON_NaoFiscal_Cancelar_ComprovanteEX.....	103
4.2.60	EPSON_NaoFiscal_Fechar_Comprovante .....	104
4.2.61	EPSON_NaoFiscal_Abrir_CCD.....	105
4.2.62	EPSON_NaoFiscal_Abrir_Relatorio_Gerencial.....	106
4.2.63	EPSON_NaoFiscal_Imprimir_LinhaEX .....	107
4.2.64	EPSON_NaoFiscal_Imprimir_Linha .....	108
4.2.65	EPSON_NaoFiscal_Imprimir_15Linhas.....	110
4.2.66	EPSON_NaoFiscal_Fechar_CCD.....	112
4.2.67	EPSON_NaoFiscal_Fechar_Relatorio_Gerencial .....	113
4.2.68	EPSON_NaoFiscal_Cancelar_CCD.....	114
4.2.69	EPSON_NaoFiscal_Cancelar_Pagamento .....	116
4.2.70	EPSON_NaoFiscal_Nova_Parcela_CCD.....	117
4.2.71	EPSON_NaoFiscal_Nova_Via_CCD.....	118
4.2.72	EPSON_NaoFiscal_Reimprimir_CCD .....	119
4.2.73	EPSON_NaoFiscal_Sangria.....	120
4.2.74	EPSON_NaoFiscal_Fundo_Troco.....	121
4.2.75	EPSON_NaoFiscal_Imprimir_Codigo_Barras .....	122
4.2.76	EPSON_NaoFiscal_Obter_SubTotal .....	126
4.2.77	EPSON_ESC_ECF_NaoFiscal_Abrir_CCD.....	128
4.2.78	EPSON_ESC_ECF_NaoFiscal_Abrir_CCDEX .....	131
4.2.79	EPSON_ESC_ECF_NaoFiscal_Imprimir_Linha .....	133
4.2.80	EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RG.....	134
4.2.81	EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RGEX.....	136
4.2.82	EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_Gerencial.....	137
4.2.83	EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_GerencialEX .....	139
4.2.84	EPSON_ESC_ECF_NaoFiscal_Cancelar_CCD.....	140
4.2.85	EPSON_ESC_ECF_NaoFiscal_Cancelar_CCDEX .....	142
4.2.86	EPSON_ESC_ECF_NaoFiscal_Nova_Via_CCD.....	143
4.2.87	EPSON_ESC_ECF_NaoFiscal_Reimprimir_CCD .....	144
4.2.88	EPSON_ESC_ECF_NaoFiscal_Abrir_Comprovante.....	145
4.2.89	EPSON_ESC_ECF_NaoFiscal_Abrir_ComprovanteEX.....	147
4.2.90	EPSON_ESC_ECF_NaoFiscal_Vender_Item .....	148
4.2.91	EPSON_ESC_ECF_NaoFiscal_Vender_ItemEX .....	150
4.2.92	EPSON_ESC_ECF_NaoFiscal_Fechar_Comprovante .....	151
4.2.93	EPSON_ESC_ECF_NaoFiscal_Fechar_ComprovanteEX .....	153
4.2.94	EPSON_ESC_ECF_NaoFiscal_Cancelar_Pagamento .....	154
4.2.95	EPSON_ESC_ECF_NaoFiscal_Cancelar_PagamentoEX.....	157
4.2.96	EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_Troco.....	158
4.2.97	EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_TrocoEX.....	160
4.2.98	EPSON_ESC_ECF_NaoFiscal_Imprime_QRCode.....	161

<b>4.3</b>	<b>GRUPO DE RELATÓRIOS FISCAIS .....</b>	<b>162</b>
4.3.1	EPSON_RelatorioFiscal_LeituraX.....	162
4.3.2	EPSON_RelatorioFiscal_RZ.....	163
4.3.3	EPSON_RelatorioFiscal_RZEX .....	165
4.3.4	EPSON_RelatorioFiscal_Leitura_MF.....	166
4.3.5	EPSON_RelatorioFiscal_Salvar_LeituraX.....	168
4.3.6	EPSON_RelatorioFiscal_Abrir_Jornada .....	169
4.3.7	EPSON_ESC_ECF_RelatorioFiscal_LeituraX.....	170
4.3.8	EPSON_ESC_ECF_RelatorioFiscal_LeituraXEX .....	171
4.3.9	EPSON_ESC_ECF_RelatorioFiscal_RZ.....	173
4.3.10	EPSON_ESC_ECF_RelatorioFiscal_RZEX.....	175
4.3.11	EPSON_ESC_ECF_RelatorioFiscal_Leitura_MF .....	176
4.3.12	EPSON_ESC_ECF_RelatorioFiscal_Leitura_MFEX .....	178
4.3.13	EPSON_ESC_ECF_RelatorioFiscal_Imprime_RTD_RZ .....	180
4.3.14	EPSON_ESC_ECF_RelatorioFiscal_Reimprimir_MFD.....	181
<b>4.4</b>	<b>GRUPO OBTER INFORMAÇÕES DA IMPRESSORA. ....</b>	<b>182</b>
4.4.1	EPSON_Obter_Dados_Usuario .....	182
4.4.2	EPSON_Obter_Tabela_Aliquotas .....	184
4.4.3	EPSON_Obter_Tabela_Aliquotas_Cupom .....	186
4.4.4	EPSON_Obter_Tabela_Pagamentos .....	188
4.4.5	EPSON_Obter_Tabela_NaoFiscais .....	190
4.4.6	EPSON_Obter_Tabela_Relatorios_Gerenciais .....	192
4.4.7	EPSON_Obter_Total_Cancelado.....	194
4.4.8	EPSON_Obter_Total_Aliquotas.....	196
4.4.9	EPSON_Obter_Total_Bruto.....	198
4.4.10	EPSON_Obter_Total_Descontos.....	199
4.4.11	EPSON_Obter_Total_Acrescimos .....	201
4.4.12	EPSON_Obter_Total_Troco.....	203
4.4.13	EPSON_Obter_Venda_Liquida_ICMS .....	204
4.4.14	EPSON_Obter_Venda_Liquida_ISSQN.....	205
4.4.15	EPSON_Obter_Total_ICMS.....	207
4.4.16	EPSON_Obter_Total_ISSQN .....	208
4.4.17	EPSON_Obter_Dados_Impressora .....	209
4.4.18	EPSON_Obter_Cliche_Usuario .....	211
4.4.19	EPSON_Obter_Cliche_UsuarioEX.....	213
4.4.20	EPSON_Obter_Data_Hora_Jornada.....	215
4.4.21	EPSON_Obter_Numero_ECF_Loja .....	217
4.4.22	EPSON_Obter_Hora_Relogio.....	219
4.4.23	EPSON_Obter_Simbolo_Moeda.....	221
4.4.24	EPSON_Obter_Casas_Decimais .....	222
4.4.25	EPSON_Obter_Desconto_Iss .....	224
4.4.26	EPSON_Obter_Contadores .....	226
4.4.27	EPSON_Obter_Estado_ImpressoraEX.....	228
4.4.28	EPSON_Obter_Estado_Impressora.....	234
4.4.29	EPSON_Obter_GT.....	240
4.4.30	EPSON_Obter_Linhas_Impressas.....	241
4.4.31	EPSON_Obter_Linhas_Impressas_RG.....	242
4.4.32	EPSON_Obter_Linhas_Impressas_CCD .....	244
4.4.33	EPSON_Obter_Dados_Jornada .....	245
4.4.34	EPSON_Obter_Caracteres_Linha.....	247
4.4.35	EPSON_Obter_Operador.....	248
4.4.36	EPSON_Obter_Numero_Ultimo_Item .....	249
4.4.37	EPSON_Obter_Informacao_Item .....	250
4.4.38	EPSON_Obter_Estado_Cupom .....	252
4.4.39	EPSON_Obter_Informacao_Ultimo_Documento.....	254
4.4.40	EPSON_Obter_Estado_Corte_Papel.....	256
4.4.41	EPSON_Obter_Linhas_Impressas_Vendas.....	257
4.4.42	EPSON_Obter_Linhas_Impressas_Pagamentos .....	258
4.4.43	EPSON_Obter_Total_Itens_Vendidos .....	260
4.4.44	EPSON_Obter_Estado_Memoria_Fiscal.....	261

4.4.45	EPSON_Obter_Estado_MFD .....	263
4.4.46	EPSON_Obter_Total_Leituras_X_Impressas .....	265
4.4.47	EPSON_Obter_Estado_Horario_Verao .....	266
4.4.48	EPSON_Obter_Venda_Bruta .....	268
4.4.49	EPSON_Obter_Mensagem_Erro .....	270
4.4.50	EPSON_Obter_Dados_MF_MFD .....	272
4.4.51	EPSON_Obter_Dados_Arquivos_MF_MFD .....	275
4.4.52	EPSON_Obter_Dados_MF_MFD_Progresso .....	278
4.4.53	EPSON_Obter_Versao_DLL .....	281
4.4.54	EPSON_Obter_Total_JornadaEX .....	282
4.4.55	EPSON_Obter_Dados_Ultima_RZ .....	284
4.4.56	EPSON_Obter_AtoCOTEPE_SeparadoEX .....	286
4.4.57	EPSON_Obter_Arquivos_Binarios .....	287
4.4.58	EPSON_Obter_Arquivo_Binario_MF .....	289
4.4.59	EPSON_Obter_Arquivo_Binario_MFD .....	290
4.4.60	EPSON_Obter_Versao_SWBasicoEX .....	291
4.4.61	EPSON_Obter_Codigo_Nacional_ECF .....	293
4.4.62	EPSON_Obter_Numero_Usuario .....	295
4.4.63	EPSON_Obter_Arredonda_Trunca_Fabricacao .....	296
4.4.64	EPSON_Obter_Log_Comandos .....	298
4.4.65	EPSON_Obter_Estado_ReducacaoZ_Automatica .....	300
4.4.66	EPSON_Obter_Valores_Cupom .....	301
4.4.67	EPSON_ESC_ECF_Obter_Dados .....	303
4.4.68	EPSON_ESC_ECF_Obter_Binario_SB .....	308
4.4.69	EPSON_ESC_ECF_Obter_Binario_MF .....	309
4.4.70	EPSON_ESC_ECF_Obter_Binario_MFD .....	310
4.4.71	EPSON_ESC_ECF_Obter_Binario_TDM .....	311
4.4.72	EPSON_ESC_ECF_Obter_Versao_Comandos .....	312
4.4.73	EPSON_ESC_ECF_Obter_XML_Cupom_Fiscal .....	314
4.4.74	EPSON_ESC_ECF_Obter_Tabela_NaoFiscais .....	315
4.4.75	EPSON_ESC_ECF_Obter_Chave_Busca_CFE .....	317
4.5	GRUPO DE CONFIGURAÇÃO DA IMPRESSORA .....	318
4.5.1	EPSON_Config_Aliquota .....	318
4.5.2	EPSON_Config_Relatorio_Gerencial .....	320
4.5.3	EPSON_Config_Forma_Pagamento .....	321
4.5.4	EPSON_Config_Forma_PagamentoEX .....	322
4.5.5	EPSON_Config_Totalizador_NaoFiscal .....	323
4.5.6	EPSON_Config_Horario_Verao .....	324
4.5.7	EPSON_Config_Espaco_Entre_Documentos .....	325
4.5.8	EPSON_Config_Espaco_Entre_Linhas .....	326
4.5.9	EPSON_Config_Habilita_Logotipo .....	327
4.5.10	EPSON_Config_Logotipo .....	328
4.5.11	EPSON_Config_Operador .....	330
4.5.12	EPSON_Config_OperadorEX .....	331
4.5.13	EPSON_Config_Corte_Papel .....	332
4.5.14	EPSON_Config_Serial_Impressora .....	333
4.5.15	EPSON_Config_Dados_Sintegra .....	334
4.5.16	EPSON_Config_Dados_SPED .....	336
4.5.17	EPSON_Config_Habilita_CAT52_Auto .....	337
4.5.18	EPSON_Config_Habilita_RJSSER16_Auto .....	339
4.5.19	EPSON_Config_Habilita_Mensagem_Cupom_Mania .....	341
4.5.20	EPSON_Config_Habilita_Mensagem_Minas_Legal .....	343
4.5.21	EPSON_Config_Habilita_PAFECF_Auto .....	345
4.5.22	EPSON_Config_Dados_PAFECF .....	347
4.5.23	EPSON_Config_Mensagem_Aplicacao .....	349
4.5.24	EPSON_Config_Mensagem_Aplicacao_Auto .....	350
4.5.25	EPSON_Config_Habilita_Sintegra_Auto .....	351
4.5.26	EPSON_Config_Habilita_EAD .....	353
4.5.27	EPSON_Config_ReducacaoZ_Automatica .....	354
4.5.28	EPSON_ESC_ECF_Config_Mensagem_Aplicacao .....	355
4.5.29	EPSON_ESC_ECF_Config_Horario_Verao .....	356



4.5.30	EPSON_ESC_ECF_Config_Aliquota .....	357
4.5.31	EPSON_ESC_ECF_Config_Forma_Pagamento.....	358
4.5.32	EPSON_ESC_ECF_Config_Totalizador_NaoFiscal.....	359
4.5.33	EPSON_ESC_ECF_Config_Relatorio_Gerencial.....	360
4.5.34	EPSON_ESC_ECF_Config_Loja .....	361
4.5.35	EPSON_ESC_ECF_Config_Dados_Consumidor_Rodape.....	362
4.5.36	EPSON_ESC_ECF_Config_Operador .....	363
4.6	GRUPO DO CHEQUE .....	364
4.6.1	EPSON_Cheque_Configurar_Moeda.....	364
4.6.2	EPSON_Cheque_Configurar_Parametros1.....	365
4.6.3	EPSON_Cheque_Configurar_Parametros2.....	367
4.6.4	EPSON_Cheque_Imprimir.....	369
4.6.5	EPSON_Cheque_ImprimirEX .....	370
4.6.6	EPSON_Cheque_Preparar_Endosso.....	371
4.6.7	EPSON_Cheque_Endosso_Estacao .....	372
4.6.8	EPSON_Cheque_Imprimir_Endosso .....	374
4.6.9	EPSON_Cheque_Ejetar_Endosso .....	375
4.6.10	EPSON_Cheque_Cancelar_Impressao.....	376
4.6.11	EPSON_Cheque_Endosso_EstacaoEX.....	377
4.6.12	EPSON_Cheque_Endosso_EstacaoCFG .....	379
4.6.13	EPSON_Cheque_Ler_MICR.....	380
4.7	GRUPO DA IMPRESSORA .....	381
4.7.1	EPSON_Impressora_Abrir_Gaveta.....	381
4.7.2	EPSON_Impressora_Cortar_Papel.....	382
4.7.3	EPSON_Impressora_Avançar_Papel .....	383
4.7.4	EPSON_ESC_ECF_Impressora_Abrir_Gaveta.....	384
4.7.5	EPSON_ESC_ECF_Impressora_Cortar_Papel.....	385
4.8	GRUPO DE AUTENTICAÇÃO .....	386
4.8.1	EPSON_Autenticar_Imprimir .....	386
4.8.2	EPSON_Autenticar_Reimprimir.....	387
4.9	GRUPO DO DISPLAY .....	388
4.9.1	EPSON_Display_Enviar_Texto.....	388
4.9.2	EPSON_Display_Cursor.....	389
4.9.3	EPSON_Display_Apagar_Texto.....	391
4.9.4	EPSON_Display_Configurar .....	392
4.9.5	EPSON_Display_Inicializar.....	394
4.10	GRUPO DE FUNÇÕES DE SISTEMA.....	395
4.10.1	EPSON_Sys_Informar_Handle_Janela.....	395
4.10.2	EPSON_Sys_Atualizar_Janela .....	396
4.10.3	EPSON_Sys_Aguardar_Arquivo.....	397
4.10.4	EPSON_Sys_Bloquear_Entradas .....	398
4.10.5	EPSON_Sys_Log.....	399
5	INTERFACE OCX.....	401
6	SINTEGRA (CONVÊNIO ICMS 57/95).....	403
7	ATO/COTEPE 17/04 (CAT 52).....	404
8	ATO/COTEPE 06/08 (PAF-ECF).....	405
8.1	COMANDOS DE AUXILIO O DESENVOLVIMENTO DO PROGRAMA APLICATIVO FISCAL (PAF). .....	405
8.2	GERAÇÃO DO ARQUIVO ELETRÔNICO DE REGISTROS EFETUADOS PELO PAF-ECF.....	405

## 1 Convenções

### Convenção de Símbolos

Símbolo

Significado...



Este símbolo indica que o texto que vem logo em seguida é uma referência a outros tópicos deste documento.



Este símbolo indica que em seguida encontra-se uma dica de como utilizar a interface.

Tabela 1 – Convenção de Símbolos

### Tipos de Dados Suportados

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

Tabela 2 – Tipos de Dados

## 2 Introdução

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

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

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

### 3 Retornos das Funções

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

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

Tabela 3 – Retornos das Funções

### 4 Funções da Interface

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

```
function Nome_Função(...)
```

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

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

Tabela 4 – Tipos de Dados

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

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

Tabela 5 – Tipos de Ponteiros

Por convenção, todas as tabelas que detalham as posições em buffers retornados pela InterfaceEpson utilizam a posição "0" como sendo a posição inicial do mesmo (notação utilizada por linguagens como C/C++, Delphi e Java). Caso a linguagem de programação utilizada utilize por convenção a posição "1" como sendo a posição inicial de um buffer, todas as posições das tabelas devem ser acrescidas de uma unidade.



## 4.1 Grupo da Porta Serial

As funções deste grupo são utilizadas para controlar a porta de comunicação serial.

### 4.1.1 EPSON\_Serial\_Abrir\_Porta

Esta função abre a porta de comunicação com a impressora fiscal. A execução bem sucedida desta função (ou uma de suas similares) é necessária para o funcionamento de todos os demais comandos da interface.

- Dica – Existe uma chave no registro do Windows chamada **PortAuto** que faz com que a interface, caso não consiga abrir a porta com os parâmetros informados, busque automaticamente por uma impressora conectada ao PDV. A localização desta chave no registro é: HKEY\_LOCAL\_MACHINE \ SOFTWARE \ EPSON \ InterfaceEpson.

#### Sintaxe:

```
function EPSON_Serial_Abrir_Porta(dwVelocidade:Integer; wPorta:ShortInt):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

#### Entradas:

Variável	Tipo	Tam.	Descrição
dwVelocidade	INTEGER	-	Velocidade da porta serial 115200 – 57600 – 38400 – 19200 – 9600
wPorta	SHORTINT	-	Número da porta 0 – porta USB 1..8 – portas seriais

#### Saídas:

Nenhum.

#### Retornos:

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

#### Requisitos:

A porta devera estar fechada.

---

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

```
Retorno = EPSON_Serial_Abrir_Porta( 38400, 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Serial_Abrir_Porta( 38400, 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Serial_Abrir_Porta( 38400, 1 );
```

**4.1.2 EPSON\_Serial\_Abrir\_Fechar\_Porta\_CMD**

Esta função abre a porta de comunicação com a impressora fiscal de maneira automática sempre que um comando da interface for executado, e a fecha após a execução deste comando. A execução bem sucedida desta função (ou uma de suas similares) é necessária para o funcionamento de todos os demais comandos da interface.

- Dica – Existe uma chave no registro do Windows chamada **PortAuto** que faz com que a interface, caso não consiga abrir a porta com os parâmetros informados, busque automaticamente por uma impressora conectada ao PDV. A localização desta chave no registro é: HKEY\_LOCAL\_MACHINE \ SOFTWARE \ EPSON \ InterfaceEpson.

**Sintaxe:**

```
Function EPSON_Serial_Abrir_Fechar_Porta_CMD( dwVelocidade:Integer;wPorta:ShortInt ):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwVelocidade	INTEGER	-	Velocidade da porta serial <b>115200 – 57600 – 38400 – 19200 – 9600</b>
wPorta	SHORTINT	-	Número da porta serial <b>-1</b> – auto detecção da porta <b>0</b> – porta USB <b>1</b> até <b>8</b> – portas seriais

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A porta devera estar fechada.

O valor de 115200 bps está disponível apenas para ECF modelos TM-H6000 FBIII, TM-T81 FBIII e TM-T88 FBIII.

---

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

```
Retorno = EPSON_Serial_Abrir_Fechar_Porta_CMD ( 38400, 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Serial_Abrir_Fechar_Porta_CMD ( 38400, 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Serial_Abrir_Fechar_Porta_CMD ( 38400, 1 );
```

### 4.1.3 EPSON\_Serial\_Abrir\_PortaAD

Esta função detecta e abre automaticamente a porta de comunicação com a impressora fiscal, retornando nas variáveis passadas como parâmetros para a função, os valores da porta de comunicação localizada. A execução bem sucedida desta função (ou uma de suas similares) é necessária para o funcionamento de todos os demais comandos da interface.

**Sintaxe:**

```
function EPSON_Serial_Abrir_PortaAD(pszVelocidade:Pchar;pszPorta:Pchar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szVelocidade	PCHAR	6	Velocidade da porta
szPorta	PCHAR	2	Número da porta

**Retornos:**

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

**Requisitos:**

A porta serial fechada.

---

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

```
char szVelocidade[7], szPorta[3];  
Retorno = EPSON_Serial_Abrir_PortaAD (szVelocidade , szPorta );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szVelocidade As String  
szVelocidade = Space(7)  
Dim szPorta As String  
szPorta = Space(3)  
Retorno = EPSON_Serial_Abrir_PortaAD ( szVelocidade ,  szPorta )
```

---

**Exemplo em Delphi 7:**

```
szVelocidade: array[0..7] of Char;  
szPorta: array[0..3] of Char;  
Retorno := EPSON_Serial_Abrir_PortaAD (szVelocidade , szPorta );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szVelocidade := AnsiString(StringOfChar(' ', 6));  
szPorta := AnsiString(StringOfChar(' ', 1));  
iRetorno := EPSON_Serial_Abrir_PortaAD(PAnsiChar(szVelocidade),  
PAnsiChar(szPorta));
```



#### 4.1.4 EPSON\_Serial\_Abrir\_PortaEX

Esta função detecta e abre automaticamente a porta de comunicação com a impressora fiscal. A execução bem sucedida desta função (ou uma de suas similares) é necessária para o funcionamento de todos os demais comandos da interface.

---

**Sintaxe:**

```
function EPSON_Serial_Abrir_PortaEX():Integer; StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

**Requisitos:**

Porta devera estar fechada.

---

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

```
Retorno = EPSON_Serial_Abrir_PortaEX ( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Serial_Abrir_PortaEX ( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Serial_Abrir_PortaEX ( );
```

#### 4.1.5 EPSON\_Serial\_Fechar\_Porta

Esta função fecha a porta de comunicação serial.

---

**Sintaxe:**

```
function EPSON_Serial_Fechar_Porta():Integer; StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

---

**Requisitos:**

A porta serial deve estar aberta.

---

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

```
Retorno = EPSON_Serial_Fechar_Porta( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Serial_Fechar_Porta( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Serial_Fechar_Porta( );
```

#### 4.1.6 EPSON\_Serial\_Obter\_Estado\_Com

Verifica o estado atual da porta serial.

---

**Sintaxe:**

```
function EPSON_Serial_Obter_Estado_Com():Integer; StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

SERIAL\_SUCESSO

Porta aberta com sucesso.

FUNC\_ERRO

Erro durante a execução.

---

Obs. Se o retorno for diferente de SERIAL\_SUCESSO, a porta serial não está pronta para comunicação.

---

**Requisitos:**

Nenhum requisito é necessário.

---

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

```
Retorno = EPSON_Serial_Obter_Estado_Com( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Serial_Obter_Estado_Com( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Serial_Obter_Estado_Com( );
```

#### 4.1.7 EPSON\_Serial\_Config\_Simplificada

Configura porta serial para que não considere os controles de fluxo de hardware.

##### Sintaxe:

```
function EPSON_Serial_Config_Simplificada(dwTipo:Integer):Integer; StdCall; External 'InterfaceEpson.dll';
```

##### Entradas:

Variável	Tipo	Tam.	Descrição
dwTipo	INTEGER		<b>0</b> – Configuração padrão (DSR/DTR ativos) - <b>padrão</b> <b>1</b> – Desabilita controles (DSR/DTR desabilitados)

##### Saídas:

Nenhum.

##### Retornos:

SERIAL_SUCESSO	Parâmetro configurado com sucesso.
FUNC_ERRO	Erro durante a execução.

##### Requisitos:

Deve ser usada **antes** do comando de abertura de porta serial.

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



```
Retorno = EPSON_Serial_Config_Simplificada( 1 );
```

##### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Serial_Config_Simplificada( 1 );
```

##### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



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

## 4.2 Grupo do Cupom Fiscal

Estas funções são usadas para impressão de cupons fiscais:

### 4.2.1 EPSON\_Fiscal\_Abrir\_Cupom

Esta função abre um Cupom Fiscal.

#### Sintaxe:

```
function EPSON_Fiscal_Abrir_Cupom( pszCPFCNPJ:PChar; pszRazaoSocialNomeDoCliente:PChar;  
    pszEnderecoLinha1:PChar; pszEnderecoLinha2:PChar;  
    dwPosicaoDadosImpressão:Integer):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

#### Entradas:

Variável	Tipo	Tam.	Descrição
pszCPFCNPJ	PCHAR	20 (O,RT)	Número do CPF / CNPJ do cliente.
pszRazaoSocialNomeDoCliente	PCHAR	30 (O, RT)	Razão Social / Nome do cliente.
pszEnderecoLinha1	PCHAR	40 (O, RT)	Endereço – Linha 01.
pszEnderecoLinha2	PCHAR	39 (O, RT)	Endereço – Linha 02.
dwPosicaoDadosImpressão	INTEGER		Posição onde os dados do cliente serão impressos: <b>0</b> - Não altera. <b>1</b> - Desabilita. <b>2</b> - No cabeçalho. <b>3</b> - No rodapé.

#### Saídas:

Nenhum.

#### Retornos:

SERIAL_SUCESSO	Parâmetro configurado com sucesso.
FUNC_ERRO	Erro durante a execução.

#### Requisitos:

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel. Caso os dados do usuário sejam nulos, os campos relativos aos mesmos serão omitidos no cupom fiscal.

---

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

```
Retorno = EPSON_Fiscal_Abrir_Cupom ( "", "", "", "", 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Abrir_Cupom ( "", "", "", "", 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Abrir_Cupom ( '', '', '', '', 1 );
```



**4.2.2 EPSON\_Fiscal\_Vender\_Item**

Esta função vende um item no Cupom Fiscal.

**Sintaxe:**

```
function EPSON_Fiscal_Vender_Item( pszCodigo:PChar; pszDescricao:PChar; pszQuantidade:PChar;  
dwCasasDecimaisQuantidade:Integer; pszUnidadeDeMedida:PChar;  
pszPrecoUnidade:PChar; dwCasasDecimaisPreco:Integer;  
pszAliquotas:PChar; dwLinhasImpressao:Integer):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	14 (A)	Código do item.
pszDescricao	PCHAR	233 (A)	Descrição do item.
pszQuantidade	PCHAR	7 (N)	Quantidade do item.
dwCasasDecimaisQuantidade	INTEGER	-	Número de casas decimais do valor informado no campo quantidade.
pszUnidadeDeMedida	PCHAR	3 (O, A)	Símbolo da unidade.
pszPrecoUnidade	PCHAR	8 (N)	Preço unitário.
dwCasasDecimaisPreco	INTEGER	-	Número de casas decimais do valor informado no campo preço.
pszAliquotas	PCHAR	2 (A) 4 (N)	Alíquota do item. ( <i>exemplos: "Ta", "Sb", "F", "NS"</i> ) ou Valor da alíquota em percentual com 2 casas decimais. ( <i>Exemplos: 1000, 1250, 0500</i> )
dwLinhasImpressao	INTEGER	-	1 - Impressão em uma linha. 2 - Impressão em duas ou mais linhas.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto e o ECF com papel.

---

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

```
Retorno = EPSON_Fiscal_Vender_Item ( "78912341234", "Produto XYZ", "1000",  
3, "Cx", "3500", 3, "Tb", 1 );  
Retorno = EPSON_Fiscal_Vender_Item ( "78912341234", "Produto XYZ", "1000",  
3, "Cx", "3500", 3, "1000", 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Vender_Item ( "78912341234", "Produto XYZ", "1000",  
3, "Cx", "3500", 3, "F", 1 )  
Retorno = EPSON_Fiscal_Vender_Item ( "78912341234", "Produto XYZ", "1000",  
3, "Cx", "3500", 3, "0500", 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Vender_Item ( '78912341234', 'Produto XYZ', '1000',  
3, 'Cx', '3500', 3, 'FS', 1 );  
Retorno := EPSON_Fiscal_Vender_Item ( '78912341234', 'Produto XYZ', '1000',  
3, 'Cx', '3500', 3, '1200', 1 );
```

**4.2.3 EPSON\_Fiscal\_Vender\_Item\_AD**

Esta função vende item no Cupom Fiscal, permitindo selecionar se o resultado da operação de multiplicação do preço pela quantidade deverá ser truncado ou arredondado. Este comando permite também salvar a informação se o item é de fabricação própria ou de terceiros podendo ser usada posteriormente na geração dos registros documento PAF.

**Sintaxe:**

```
function EPSON_Fiscal_Vender_Item_AD( pszCodigo:PChar; pszDescricao:PChar; pszQuantidade:PChar;  
dwCasasDecimaisQuantidade:Integer; pszUnidadeDeMedida:PChar;  
pszPrecoUnidade:PChar; dwCasasDecimaisPreco:Integer;  
pszAliquotas:PChar; dwLinhas:Integer; dwArredondaTrunca:Integer;  
dwFabricacaoPropria:Integer):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszCodigo	PCHAR	14 (A)	Código do item.
pszDescricao	PCHAR	233 (A)	Descrição do item.
pszQuantidade	PCHAR	7 (N)	Quantidade, em duas ou três casas decimais.
dwCasasDecimaisQuantidade	INTEGER	-	Número de casas decimais do valor informado no campo quantidade.
pszUnidadeDeMedida	PCHAR	(O, A)	Símbolo da unidade.
pszPrecoUnidade	PCHAR	8 (N)	Preço unitário, em duas ou três casas decimais.
dwCasasDecimaisPreco	INTEGER	-	Número de casas decimais do valor informado no campo preço.
pszAliquotas	PCHAR	2 (A) 4 (N)	Alíquota do item. (exemplos: "Ta", "Sb", "F", "NS") ou Valor da alíquota em percentual com 2 casas decimais. (Exemplos: 1000, 1250, 0500)
dwLinhas	INTEGER	-	1 - impressão em uma linha. 2 - impressão em duas ou mais linhas.
dwArredondaTrunca	INTEGER	-	1 - Trunca. 2 - Arredonda.
bFabricacaoPropria	INTEGER	-	1 - item de fabricado por terceiros. 2 - item de fabricação própria.

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto e o ECF com papel.  
Comando disponível apenas para ECF modelos TM-H6000 FBIII, TM-T81 FBIII e TM-T88 FBIII.

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

```
Retorno = EPSON_Fiscal_Vender_Item_AD ( "78912341234", "Produto XYZ",  
"1000", 3, "Cx", "3500", 3, "Tb", 1, 1, 1 );  
Retorno = EPSON_Fiscal_Vender_Item_AD ( "78912341234", "Produto XYZ",  
"1000", 3, "Cx", "3500", 3, "1000", 1, 1, 1 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Vender_Item_AD ( "78912341234", "Produto XYZ",  
"1000", 3, "Cx", "3500", 3, "F", 1, 1, 1 )  
Retorno = EPSON_Fiscal_Vender_Item_AD ( "78912341234", "Produto XYZ",  
"1000", 3, "Cx", "3500", 3, "0500", 1, 1, 1 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Vender_Item_AD ( '78912341234', 'Produto XYZ',  
'1000', 3, 'Cx', '3500', 3, 'FS', 1, 1, 1 );  
Retorno := EPSON_Fiscal_Vender_Item_AD ( '78912341234', 'Produto XYZ',  
'1000', 3, 'Cx', '3500', 3, '1200', 1, 1, 1 );
```

**4.2.4 EPSON\_Fiscal\_Obter\_SubTotal**

Esta função retorna o subtotal do cupom corrente ou o subtotal do último cupom fiscal.

**Sintaxe:**

```
function EPSON_Fiscal_Obter_SubTotal(pszSubTotalCupom:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszSubTotalCupom	PCHAR	13 (N)	Subtotal do cupom.

**Retornos:**

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

**Requisitos:**

Subtotal do cupom corrente: a Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto e o ECF com papel.

Subtotal do último cupom: a impressora deve estar com os documentos fechados e o último documento impresso deve ser um cupom fiscal.

---

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

```
char szSubtotalCupom[20];  
Retorno = EPSON_Fiscal_Obter_SubTotal ( szSubtotalCupom );
```

---

**Exemplo em C# :**

```
StringBuilder szSubtotalCupom = new StringBuilder(20,20);  
Retorno = EPSON_Fiscal_Obter_SubTotal ( szSubtotalCupom );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szSubTotalCupom As String  
szSubTotalCupom = Space(20)  
Retorno = EPSON_Fiscal_Obter_SubTotal ( szSubTotalCupom )
```

---

**Exemplo em Delphi 7:**

```
szSubTotalCupom: array[0..20] of Char;  
Retorno := EPSON_Fiscal_Obter_SubTotal ( szSubTotalCupom );
```

---

**Exemplo Delphi 2010 / Delphi XE:**

```
szSubTotalCupom := AnsiString(StringOfChar(' ', 20));  
iRetorno := EPSON_Fiscal_Obter_SubTotal(PAnsiChar(szSubTotalCupom));
```



#### 4.2.5 EPSON\_Fiscal\_Pagamento

Esta função processa o pagamento do Cupom Fiscal.

**Sintaxe:**

```
function EPSON_Fiscal_Pagamento( pszNumeroDescricaoPagamento:PChar; pszValorPagamento:PChar;  
dwCasasDecimaisPagamento:Integer; pszLinhaDescricao1:PChar;  
pszLinhaDescricao2:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroDescricaoPagamento	PCHAR	2 (N) / 15 (A)	Número do pagamento cadastrado na impressora ou descritivo da forma de pagamento.
pszValorPagamento	PCHAR	13 (N)	Valor fornecido.
dwCasasDecimaisPagamento	INTEGER	-	Número de casas decimais.
pszLinhaDescricao1	PCHAR	40 (O,RT)	Primeira linha de descrição (opcional).
pszLinhaDescricao2	PCHAR	40 (O,RT)	Segunda linha de descrição (opcional).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto e o ECF com papel.

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

```
Retorno = EPSON_Fiscal_Pagamento ( "1", "20000", 3, "", "" );  
Retorno = EPSON_Fiscal_Pagamento ( "Dinheiro", "20000", 3, "", "" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Pagamento ( "1", "20000", 3, "", "" )  
Retorno = EPSON_Fiscal_Pagamento ( "Dinheiro", "20000", 3, "", "" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Pagamento ( '1', '20000', 3, '', '' );  
Retorno := EPSON_Fiscal_Pagamento ( 'Dinheiro', '20000', 3, '', '' );
```

#### 4.2.6 EPSON\_Fiscal\_Desconto\_A acrescimo\_Item

Esta função efetua desconto ou acréscimo no último item vendido no cupom.

**Sintaxe:**

```
function EPSON_Fiscal_Desconto_A acrescimo_Item( pszValorDescontoAcrescimo:PChar;  
dwNumeroCasasDecimais:Integer;  
bTipoDescontoAcrescimo:Boolean;  
bTipoPercentagemMoeda:Boolean):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszValorDescontoAcrescimo	PCHAR	11 (N)	Valor do desconto ou acréscimo.
dwNumeroCasasDecimais	INTEGER	-	Número de casas decimais do valor.
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - Desconto. <b>False</b> - Acréscimo.
bTipoPercentagemMoeda	BOOLEAN	-	<b>True</b> - Valor percentual. <b>False</b> - Valor em moeda.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

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

```
Retorno = EPSON_Fiscal_Desconto_A acrescimo_Item ( "2500", 3, true, false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Desconto_A acrescimo_Item ( "2500", 3, true, false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Desconto_A acrescimo_Item ( '2500', 3, true, false);
```

**4.2.7 EPSON\_Fiscal\_Desconto\_A acrescimo\_ItemEX**

Esta função efetua desconto ou acréscimo em qualquer item do cupom.

**Sintaxe:**

```
function EPSON_Fiscal_Desconto_A acrescimo_ItemEX( pszNumeroltemCupom:PChar;  
                                                    pszValorDescontoAcrescimo:PChar;  
                                                    dwNumeroCasasDecimais:Integer;  
                                                    bTipoDescontoAcrescimo:Boolean;  
                                                    bTipoPercentagemMoeda:Boolean):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroltemCupom	PCHAR	3 (N)	Número do item no cupom.
pszValorDescontoAcrescimo	PCHAR	11 (N)	Valor do desconto ou acréscimo.
dwNumeroCasasDecimais	INTEGER	-	Número de casas decimais do valor.
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - Desconto. <b>False</b> - Acréscimo.
bTipoPercentagemMoeda	BOOLEAN	-	<b>True</b> - Valor percentual. <b>False</b> - Valor em moeda.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

Comando disponível apenas para ECF modelos TM-H6000 FBIII, TM-T81 FBIII e TM-T88 FBIII.

---

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

```
Retorno = EPSON_Fiscal_Desconto_Acrescimo_ItemEX ( "1", "2500", 3, true,  
false );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Desconto_Acrescimo_ItemEX ( "1", "2500", 3, true,  
false )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Desconto_Acrescimo_ItemEX ( '1', '2500', 3, true,  
false);
```

**4.2.8 EPSON\_Fiscal\_Desconto\_Acrescimo\_Subtotal**

Esta função efetua um desconto ou acréscimo no subtotal do cupom.

**Sintaxe:**

```
function EPSON_Fiscal_Desconto_Acrescimo_Subtotal( pszValorDescontoAcrescimo:PChar;  
dwNumeroCasasDecimais:Integer;  
bTipoDescontoAcrescimo:Boolean;  
bTipoPercentagemMoeda:Boolean):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszValorDescontoAcrescimo	PCHAR	11 (N)	Valor do desconto ou acréscimo.
dwNumeroCasasDecimais	INTEGER	-	Número de casas decimais do valor.
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> – Desconto. <b>False</b> - Acréscimo.
bTipoPercentagemMoeda	BOOLEAN	-	<b>True</b> - Valor percentual. <b>False</b> - Valor em moeda.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

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

```
Retorno = EPSON_Fiscal_Desconto_Acrescimo_Subtotal ( "2500", 3, true,  
false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Desconto_Acrescimo_Subtotal ( "2500", 3, true,  
false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Desconto_Acrescimo_Subtotal ( '2500', 3, true,  
false );
```

### 4.2.9 EPSON\_Fiscal\_Cancelar\_Cupom

Esta função cancela o último cupom fiscal.

---

**Sintaxe:**

```
function EPSON_Fiscal_Cancelar_Cupom():Integer; StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

---

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas e o ECF com papel.

---

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

```
Retorno = EPSON_Fiscal_Cancelar_Cupom ( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Cancelar_Cupom ( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Cancelar_Cupom ( );
```



**4.2.10 EPSON\_Fiscal\_Cancelar\_CupomEX**

Esta função cancela o último cupom fiscal. Caso existam Comprovantes de Crédito ou Débito relacionados a este cupom, os mesmos também serão cancelados.

**Sintaxe:**

```
function EPSON_Fiscal_Cancelar_CupomEX():Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas e o ECF com papel.

O último documento impresso deverá ser um Cupom Fiscal ou um Comprovante de Crédito ou Débito vinculado ao último Cupom Fiscal.

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

```
Retorno = EPSON_Fiscal_Cancelar_CupomEX( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Cancelar_CupomEX( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Cancelar_CupomEX( );
```

**4.2.11 EPSON\_Fiscal\_Cancelar\_Item**

Esta função cancela um item do cupom fiscal, caso este item seja informado com um valor nulo, a função cancelará o ultimo item inserido no cupom.

**Sintaxe:**

```
function EPSON_Fiscal_Cancelar_Item(pszNumerolItem:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumerolItem	PCHAR	3 (O,N)	Item a ser cancelado.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

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

```
Retorno = EPSON_Fiscal_Cancelar_Item ( "2" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Cancelar_Item ( "2" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Cancelar_Item ( '2' );
```

**4.2.12 EPSON\_Fiscal\_Cancelar\_Ultimo\_Item**

Esta função cancela o último item vendido no cupom fiscal atual.

**Sintaxe:**

```
function EPSON_Fiscal_Cancelar_Ultimo_Item():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

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

```
Retorno = EPSON_Fiscal_Cancelar_Ultimo_Item ( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Cancelar_Ultimo_Item ( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Cancelar_Ultimo_Item ( );
```

**4.2.13 EPSON\_Fiscal\_Cancelar\_Desconto\_Acrescimo\_Item**

Esta função cancela o desconto ou acréscimo aplicado sobre o último item vendido no cupom fiscal atual.

**Sintaxe:**

```
function EPSON_Fiscal_Cancelar_Desconto_Acrescimo_Item(bDescontoAcrescimoCancelar:Boolean);  
Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bDescontoAcrescimoCancelar	BOOLEAN	-	<b>True</b> - cancelar o desconto. <b>False</b> - cancelar o acréscimo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e o item referido deve possuir acréscimo ou desconto.

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

```
Retorno = EPSON_Fiscal_Cancelar_Desconto_Acrescimo_Item ( false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Cancelar_Desconto_Acrescimo_Item ( false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Cancelar_Desconto_Acrescimo_Item ( false );
```

**4.2.14 EPSON\_Fiscal\_Cancelar\_Desconto\_Acrescimo\_ItemEX**

Esta função cancela o desconto ou acréscimo aplicado sobre qualquer item vendido no cupom fiscal atual.

**Sintaxe:**

```
function EPSON_Fiscal_Cancelar_Desconto_Acrescimo_ItemEX( pszNumerolItem:PChar;  
                                                         bTipoDescontoAcrescimo:Boolean):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumerolItem	PCHAR	3 (N)	Número do item.
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - cancelar o desconto. <b>False</b> - cancelar o acréscimo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e o item referido deve possuir acréscimo ou desconto.

Comando disponível apenas para ECF modelos TM-H6000 FBIII, TM-T81 FBIII e TM-T88 FBIII.

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

```
Retorno = EPSON_Fiscal_Cancelar_Desconto_Acrescimo_ItemEX ( "2", false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Cancelar_Desconto_Acrescimo_ItemEX ( "2", false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Cancelar_Desconto_Acrescimo_ItemEX ( '2', false );
```

**4.2.15 EPSON\_Fiscal\_Cancelar\_A acrescimo\_Desconto\_Subtotal**

Esta função cancela o desconto ou acréscimo aplicado sobre o subtotal do cupom fiscal atual.

**Sintaxe:**

```
function EPSON_Fiscal_Cancelar_A acrescimo_Desconto_Subtotal( bTipoDescontoAcrescimo:Boolean );  
Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - cancelar o desconto. <b>False</b> - cancelar o acréscimo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel, e o cupom fiscal deve possuir acréscimo ou desconto em subtotal.

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

```
Retorno = EPSON_Fiscal_Cancelar_A acrescimo_Desconto_Subtotal ( false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Cancelar_A acrescimo_Desconto_Subtotal ( false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Cancelar_A acrescimo_Desconto_Subtotal ( false );
```

**4.2.16 EPSON\_Fiscal\_Cancelar\_Item\_Parcial**

Esta função cancela parcialmente um item.

**Sintaxe:**

```
function EPSON_Fiscal_Cancelar_Item_Parcial( pszNumerolItemCancelar:PChar;  
                                             pszQuantidadeCancelar:PChar;  
                                             dwNumeroCasasDecimais:Integer): Integer;StdCall; External  
                                             'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumerolItemCancelar	PCHAR	3	Item a ser cancelado.
pszQuantidadeCancelar	PCHAR	7 (N)	Quantidade a cancelar.
dwNumeroCasasDecimais	INTEGER	-	Número de casas decimais utilizada.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

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

```
Retorno = EPSON_Fiscal_Cancelar_Item_Parcial ( "2", "3000", 3 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Cancelar_Item_Parcial ( "2", "3000", 3 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Cancelar_Item_Parcial ( '2', '3000', 3 );
```

**4.2.17 EPSON\_Fiscal\_Imprimir\_Mensagem**

Esta função imprime oito linhas de texto depois da totalização do ticket. As linhas em branco serão omitidas. Para imprimir código de barras nas linhas de texto promocional veja detalhes na função EPSON\_Fiscal\_Configurar\_Codigo\_Barras\_Mensagem.

**Sintaxe:**

```
function EPSON_Fiscal_Imprimir_Mensagem( pszLinhaTexto1:PChar; pszLinhaTexto2:PChar;  
                                          pszLinhaTexto3:PChar; pszLinhaTexto4:PChar;  
                                          pszLinhaTexto5:PChar; pszLinhaTexto6:PChar;  
                                          pszLinhaTexto7:PChar; pszLinhaTexto8:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszLinhaTexto1 a pszLinhaTexto8	PCHAR	56 (O,RT)	Linhas de texto a imprimir.

**Tags de Formatação:**

É possível formatar a linha e palavras a serem impressas para utilizar fontes em negrito, sublinhado, largura dupla, altura dupla, e cor invertida utilizando os seguintes tags de formatação antecedendo as palavras a serem impressas.

- <AT> - Ativa todos os atributos de formatação
- <AN> - Ativa negrito
- <AS> - Ativa sublinhado
- <AL> - Ativa largura dupla
- <AA> - Ativa altura Dupla
- <AI> - Ativa cor invertida
- <DT> - Desativa todos
- <DN> - Desativa negrito
- <DS> - Desativa sublinhado
- <DL> - Desativa largura dupla
- <DA> - Desativa altura dupla
- <DI> - Desativa cor invertida

Por padrão, a impressora já permite utilizar os tags para formatar uma linha texto inteira.

Para que seja possível utilizar formatação somente por palavra, é necessário habilitar a função de atributo nos jumpers de configuração da impressora. Para isto, basta entrar em modo de Intervenção Técnica, desligar a impressora e então alterar seus estados nas microchaves de seleção (Dip-Switch).

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.



---

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal ou documento não fiscal aberto, o ECF com papel e o cupom com os devidos pagamentos já efetuados.

---

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

```
Retorno = EPSON_Fiscal_Imprimir_Mensagem ( "<AN>Linha1 negrito",  
"<AL>Linha2 Largura dupla", "3", "4", "5", "6", "7", "8" );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Imprimir_Mensagem ( "<AN>Linha1 negrito",  
"<AL>Linha2 Largura dupla", "3", "4", "5", "6", "7", "8" )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Imprimir_Mensagem ( "<AN>Linha1 negrito",  
"<AL>Linha2 Largura dupla", "3", "4", "5", "6", "7", "8" );
```

**4.2.18 EPSON\_Fiscal\_Imprimir\_MensagemEX**

Esta função imprime linhas promocionais quebrando o texto ao fim de 56 caracteres ou pelos terminadores de linha #13#10 (CR LF), limitado a 8 linhas.

**Sintaxe:**

```
function EPSON_Fiscal_Imprimir_MensagemEX(pszLinhasTexto:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszLinhasTexto	PCHAR	448 (O,RT)	Linhas de texto a imprimir.

**Tags de Formatação:**

É possível formatar a linha impressa para utilizar fontes em negrito, sublinhado, largura dupla, altura dupla, e cor invertida utilizando os seguintes tags de formatação antecedendo as palavras a serem impressas.

<AT> - Ativa todos os atributos de formatação  
<AN> - Ativa negrito  
<AS> - Ativa sublinhado  
<AL> - Ativa largura dupla  
<AA> - Ativa altura Dupla  
<AI> - Ativa cor invertida  
<DT> - Desativa todos  
<DN> - Desativa negrito  
<DS> - Desativa sublinhado  
<DL> - Desativa largura dupla  
<DA> - Desativa altura dupla  
<DI> - Desativa cor invertida

Por padrão, a impressora já permite utilizar os tags para formatar uma linha texto inteira.

Para que seja possível utilizar formatação somente por palavra, é necessário habilitar a função de atributo nos jumpers de configuração da impressora. Para isto, basta entrar em modo de Intervenção Técnica, desligar a impressora e então alterar seus estados nas microchaves de seleção (Dip-Switch).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal ou documento não fiscal aberto, o ECF com papel e o cupom com os devidos pagamentos já efetuados.

---

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

```
Retorno = EPSON_Fiscal_Imprimir_MensagemEX ( "<AT>LINHA 01\n<AN>LINHA  
02\n<AS>LINHA 03\n<AL>LINHA 04\n<AA>LINHA 05\n<AI>LINHA 06\nLINHA  
07\nLINHA 08" );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Imprimir_MensagemEX ("<AT>LINHA 01" + vbCr +  
"<AN>LINHA 02" + vbCr + "<AS>LINHA 03" + vbCr + "<AL>LINHA 04" + vbCr +  
"<AA>LINHA 05" + vbCr + "<AI>LINHA 06" + vbCr + "LINHA 07" + vbCr + "LINHA  
08" )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
iRetorno := EPSON_Fiscal_Imprimir_MensagemEX( '<AT>LINHA 01' + #13#10 +  
'<AN>LINHA 02' + #13#10 + '<AS>LINHA 03' + #13#10 + '<AL>LINHA 04' +  
#13#10 + '<AA>LINHA 05' + #13#10 + '<AI>LINHA 06' + #13#10 + 'LINHA 07' +  
#13#10 + 'LINHA 08');
```

**4.2.19 EPSON\_Fiscal\_Configurar\_Codigo\_Barras\_Mensagem**

Esta função configura as características do código de barras a ser impresso em uma das oito linhas de mensagem promocional existentes no encerramento de um cupom fiscal ou comprovante não fiscal. É possível configurar um código de barras para cada uma das oito linhas de texto promocional. Observe que com esta função configuram-se apenas as características do código de barras, sendo que o valor do mesmo (o número do código) deve ser fornecido logo em seguida pela função EPSON\_Fiscal\_Imprimir\_Mensagem, na respectiva linha.

Obs: Este comando também pode ser utilizado para imprimir código de barras em comprovante não fiscal.

**Sintaxe:**

```
function EPSON_Fiscal_Configurar_Codigo_Barras_Mensagem(dwLinhaMensagem:Integer;  
dwTipoCodigoDeBarras:Integer;  
dwAlturaImpressao:Integer;  
dwLarguraCodigoDeBarras:Integer;  
dwPosicaoImpressao:Integer;  
dwLegenda:Integer):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwLinhaMensagem	INTEGER	-	Linha da mensagem promocional onde se deseja imprimir o código de barras.
dwTipoCodigoDeBarras	INTEGER	-	Tipo de código de barras: <i>Veja tabela abaixo.</i>
dwAlturaImpressao	INTEGER	-	Altura de impressão do código de barras, entre 1 e 255.
dwLarguraCodigoDeBarras	INTEGER	-	Largura do código de barras, entre 2 e 6.
dwPosicaoImpressao	INTEGER	-	Posição de impressão do texto do código de barras: <b>0</b> - Sem impressão de texto. <b>1</b> - impressão acima do código. <b>2</b> - impressão abaixo do código. <b>3</b> - impressão acima e abaixo do código.
dwLegenda	INTEGER	-	Tipo de caractere usado no texto do código: <b>0</b> - Letra tipo A (12x24). <b>1</b> - Letra tipo B (9X17).

Valor (dwTipo)	Padrão	Quantidade de caracteres	Tipo de dados
0	UPC-A	11 à 12	0 – 9 (apenas números)
1	UPC-E	11 à 12	0 – 9 (apenas números)
2	JAN13 (EAN13)	12 à 13	0 – 9 (apenas números)
3	JAN8 (EAN8)	7 à 8	0 – 9 (apenas números)
4	CODE39	Maior ou igual a 1	0 à 9, A à Z e os caracteres: [espaço] \$ % * + - . , /
6	CODABAR (NW7)	Maior ou igual a 1	0 à 9, A à D e os caracteres: \$ + - . / :
65	UPC-A	11 à 12	0 à 9
66	UPC-E	11 à 12	0 à 9
67	JAN13 (EAN13)	12 à 13	0 à 9
68	JAN8 (EAN8)	7 à 8	0 à 9
69	CODE39	1 à 255	0 à 9, A à Z e os caracteres: [espaço] \$ % * + - . /
70	ITF	2 à 255 (par)	0 à 9
71	CODABAR (NW7)	1 à 255	0 à 9, A à D e os caracteres: \$ + - . / :
72	CODE93	1 à 255	NULL à [CTRL]
73	CODE128	2 à 255	NULL à [CTRL]

**Observação:**

**UPC-E:** Deve ser preenchido com 0, obrigatoriamente seu primeiro caractere.

**CODE128:** Os dois primeiros bytes dos dados, selecionam as tabelas abaixo A, B e C, conforme os códigos:

{ **A**(0x7B 0x41) = Seleciona tabela A

{ **B**(0x7B 0x42) = Seleciona tabela B

{ **C**(0x7B 0x43) = Seleciona tabela C

d		Character		
Hex	Decimal	CODE A	CODE B	CODE C
00	0	NUL		00
01	1	SOH		01
02	2	STX		02
03	3	ETX		03
04	4	EDT		04
05	5	ENQ		05
06	6	ACK		06
07	7	BEL		07
08	8	BS		08
09	9	HT		09
0A	10	LF		10
0B	11	VT		11
0C	12	FF		12
0D	13	CR		13
0E	14	SO		14
0F	15	SI		15
10	16	DLE		16
11	17	DC1		17
12	18	DC2		18
13	19	DC3		19
14	20	DC4		20
15	21	NAK		21
16	22	SYN		22
17	23	ETB		23
18	24	CAN		24
19	25	EM		25
1A	26	SUB		26
1B	27	ESC		27
1C	28	FS		28
1D	29	GS		29
1E	30	RS		30
1F	31	US		31
20	32	SP	SP	32
21	33	!	!	33
22	34	"	"	34
23	35	#	#	35
24	36	\$	\$	36

d		Character		
Hex	Decimal	CODE A	CODE B	CODE C
25	37	%	%	37
26	38	&	&	38
27	39	'	'	39
28	40	(	(	40
29	41	)	)	41
2A	42	*	*	42
2B	43	+	+	43
2C	44	,	,	44
2D	45	-	-	45
2E	46	.	.	46
2F	47	/	/	47
30	48	0	0	48
31	49	1	1	49
32	50	2	2	50
33	51	3	3	51
34	52	4	4	52
35	53	5	5	53
36	54	6	6	54
37	55	7	7	55
38	56	8	8	56
39	57	9	9	57
3A	58	:	:	58
3B	59	;	;	59
3C	60	<	<	60
3D	61	=	=	61
3E	62	>	>	62
3F	63	?	?	63
40	64	@	@	64
41	65	A	A	65
42	66	B	B	66
43	67	C	C	67
44	68	D	D	68
45	69	E	E	69
46	70	F	F	70
47	71	G	G	71
48	72	H	H	72
49	73	I	I	73

d		Character		
Hex	Decimal	CODE A	CODE B	CODE C
4A	74	J	J	74
4B	75	K	K	75
4C	76	L	L	76
4D	77	M	M	77
4E	78	N	N	78
4F	79	O	O	79
50	80	P	P	80
51	81	Q	Q	81
52	82	R	R	82
53	83	S	S	83
54	84	T	T	84
55	85	U	U	85
56	86	V	V	86
57	87	W	W	87
58	88	X	X	88
59	89	Y	Y	89
5A	90	Z	Z	90
5B	91	[	[	91
5C	92	\	\	92
5D	93	]	]	93
5E	94	^	^	94
5F	95	_	_	95
60	96		'	96
61	97		a	97
62	98		b	98
63	99		c	99
64	100		d	
65	101		e	
66	102		f	
67	103		g	
68	104		h	
69	105		i	
6A	106		j	
6B	107		k	
6C	108		l	
6D	109		m	
6E	110		n	

d		Character		
Hex	Decimal	CODE A	CODE B	CODE C
6F	111		o	
70	112		p	
71	113		q	
72	114		r	
73	115		s	
74	116		t	
75	117		u	
76	118		v	
77	119		w	
78	120		x	
79	121		y	
7A	122		z	
7B, 7B	123, 123		{	
7C	124			
7D	125		}	
7E	126		~	
7F	127		DEL	

d		Special Character		
Hex	Decimal	CODE A	CODE B	CODE C
7B, 31	123, 49	FNC1	FNC1	FNC1
7B, 32	123, 50	FNC2	FNC2	
7B, 33	123, 51	FNC3	FNC3	
7B, 34	123, 52	FNC4	FNC4	
7B, 41	123, 65		CODE A	CODE A
7B, 42	123, 66	CODE B		CODE B
7B, 43	123, 67	CODE C	CODE C	
7B, 53	123, 83	SHIFT	SHIFT	

### Exemplo:

Configurando código de barras tipo: 2; padrão: JAN13 (EAN13); altura 100, largura 4, posição 2 sem impressão de texto.

### Configura o código de barras:

Retorno = EPSON\_Fiscal\_Configurar\_Codigo\_Barras\_Mensagem ( 2, 2, 100, 4, 2, 1 );

### Imprime o código de barras:

Retorno = EPSON\_Fiscal\_Imprimir\_Mensagem ( "Mensagem promocional1",  
"0000000000017", "Mensagem promocional2 ", "7890000000056283", "Mensagem  
promocional3", "7890000000056283", "Mensagem promocional4", "7890000000056284" );

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal ou comprovante não fiscal aberto, o ECF com papel e o cupom com os devidos pagamentos já efetuados.

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

```
Retorno = EPSON_Fiscal_Configurar_Codigo_Barras_Mensagem ( 2, 2, 100, 4,  
2, 0 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Configurar_Codigo_Barras_Mensagem ( 2, 2, 100, 4,  
2, 0 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Configurar_Codigo_Barras_Mensagem ( 2, 2, 100, 4,  
2, 0 );
```

**4.2.20 EPSON\_Fiscal\_Fechar\_CupomEX**

Esta função finaliza rapidamente um cupom pagando em dinheiro. O valor total do cupom será retornado na variável passada como parâmetro para a função.

**Sintaxe:**

```
function EPSON_Fiscal_Fechar_CupomEX(pszTotalCupom:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalCupom	PCHAR	13 (N)	Total do cupom.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, ao menos um item vendido no cupom.

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

```
char szTotalCupom[20];  
Retorno = EPSON_Fiscal_Fechar_CupomEX ( szTotalCupom );
```

**Exemplo em C# :**

```
StringBuilder szTotalCupom = new StringBuilder(20,20);  
Retorno = EPSON_Fiscal_Fechar_CupomEX ( szTotalCupom );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalCupom As String  
szTotalCupom = Space(20)  
Retorno = EPSON_Fiscal_Fechar_CupomEX ( szTotalCupom )
```

**Exemplo em Delphi 7:**

```
szTotalCupom: array[0..20] of Char;  
Retorno := EPSON_Fiscal_Fechar_CupomEX ( szTotalCupom );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalCupom := AnsiString(StringOfChar(' ', 20));  
iRetorno := EPSON_Fiscal_Fechar_CupomEX(PAnsiChar(szTotalCupom));
```



**4.2.21 EPSON\_Fiscal\_Fechar\_Cupom**

Esta função fecha o cupom fiscal e corta o papel.

**Sintaxe:**

```
function EPSON_Fiscal_Fechar_Cupom(bCortarPapel:Boolean; blmprimirCupomAdicional:Boolean):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bCortarPapel	BOOLEAN	-	<b>True</b> - Cortar o papel depois do rodapé. <b>False</b> - Não cortar o papel depois do rodapé.
blmprimirCupomAdicional	BOOLEAN	-	<b>True</b> - Imprimir Cupom Adicional. <b>False</b> - Não imprimir Cupom Adicional.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e o cupom com os devidos pagamentos já efetuados.

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

```
Retorno = EPSON_Fiscal_Fechar_Cupom ( false, false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Fechar_Cupom ( false, false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Fechar_Cupom ( false, false );
```

**4.2.22 EPSON\_Fiscal\_Dados\_Consumidor**

Esta função configura os dados do cliente que serão impressos no cupom fiscal atual ou seguinte.

**Sintaxe:**

```
function EPSON_Fiscal_Dados_Consumidor(pszCPFCNPJ:PChar; pszRazaoSocialNomeCliente:PChar;  
    pszEnderecoLinha1:PChar; pszEnderecoLinha2:PChar;  
    dwPosicaoDadosImpressao:Integer):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszCPFCNPJ	PCHAR	20 (O,RT)	Número CPF ou CNPJ do cliente.
pszRazaoSocialNomeCliente	PCHAR	30 (O, RT)	Razão Social / Nome do cliente.
pszEnderecoLinha1	PCHAR	40 (O, RT)	Endereço – Linha 01.
pszEnderecoLinha2	PCHAR	39 (O, RT)	Endereço – Linha 02.
dwPosicaoDadosImpressao	INTEGER		Posição onde os dados do cliente serão impressos: 1 - Desabilitado. 2 - No cabeçalho. 3 - No rodapé.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

O ECF deve estar ligado e com papel.

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

```
Retorno = EPSON_Fiscal_Dados_Consumidor ( "12345678901234", "Empresa XYZ",  
    "", "", 3 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Fiscal_Dados_Consumidor ( "12345678901234", "Empresa XYZ",  
    "", "", 3 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Fiscal_Dados_Consumidor ( '12345678901234', 'Empresa  
XYZ', '', '', 3 );
```

**4.2.23 EPSON\_ESC\_ECF\_Fiscal\_Abrir\_Cupom**

Esta função efetua a abertura de um cupom fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Abrir_Cupom(szCPFCNPJ:Pchar, szRazaoSocialNomeDoCliente:Pchar,  
szEndereco:Pchar; szCOO:Pchar; szDataHora:Pchar;  
szVendaBruta:Pchar; szNumeroSerie:Pchar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCPFCNPJ	PCHAR	14 (O,RT)	Número CPF ou CNPJ do cliente.
szRazaoSocialNomeCliente	PCHAR	30 (O, RT)	Razão Social / Nome do cliente.
szEndereco	PCHAR	79 (O, RT)	Endereço do cliente.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do cupom aberto.
szDataHora	PCHAR	16 (N)	Data e hora de abertura do cupom.
szVendaBruta	PCHAR	15 (N)	Valor da Venda Bruta.
szNumeroSerie	PCHAR	21 (A)	Numero de Série do ECF.

**Retornos:**

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

**Requisitos:**

O ECF deve estar ligado, com papel e com a jornada fiscal aberta ou sem redução Z pendente.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szCOO[10], szDataHora[16], szVendaBruta[15], szNumeroSerie[21];  
Retorno = EPSON_ESC_ECF_Fiscal_Abrir_Cupom( "12345678901230", "Empresa  
XYZ", "", szCOO, szDataHora, szVendaBruta, szNumeroSerie);
```

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);  
StringBuilder szDataHora = new StringBuilder(16, 16);  
StringBuilder szVendaBruta = new StringBuilder(15, 15);  
StringBuilder szNumeroSerie = new StringBuilder(21, 21);  
Retorno = EPSON_ESC_ECF_Fiscal_Abrir_Cupom( "12345678901230", "Empresa  
XYZ", "", szCOO, szDataHora, szVendaBruta, szNumeroSerie);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String
Dim szDataHora As String
Dim szVendaBruta As String
Dim szNumeroSerie As String
szCOO = Space(10)
szDataHora = Space(16)
szVendaBruta = Space(15)
szNumeroSerie = Space(21)
Retorno = EPSON_ESC_ECF_Fiscal_Abrir_Cupom( "12345678901230", "Empresa
XYZ", "", szCOO, szDataHora, szVendaBruta, szNumeroSerie)
```

---

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;
szDataHora: array[0..16] of Char;
szVendaBruta: array[0..15] of Char;
szNumeroSerie: array[0..21] of Char;
Retorno := EPSON_ESC_ECF_Fiscal_Abrir_Cupom( '12345678901230', 'Empresa
XYZ', '', szCOO, szDataHora, szVendaBruta, szNumeroSerie);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));
szDataHora := AnsiString(StringOfChar(' ', 16));
szVendaBruta := AnsiString(StringOfChar(' ', 15));
szNumeroSerie := AnsiString(StringOfChar(' ', 21));
Retorno := EPSON_ESC_ECF_Fiscal_Abrir_Cupom( '12345678901230', 'Empresa
XYZ', '', PAnsiChar(szCOO), PAnsiChar(szDataHora),
PAnsiChar(szVendaBruta), PAnsiChar(szNumeroSerie));
```

**4.2.24 EPSON\_ESC\_ECF\_Fiscal\_Abrir\_CupomEX**

Esta função efetua a abertura de um cupom fiscal, sem o retorno de parâmetros, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Abrir_CupomEX(szCPFCNPJ:Pchar, szRazaoSocialNomeDoCliente:Pchar,  
szEndereco:Pchar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCPFCNPJ	PCHAR	14 (O,RT)	Número CPF ou CNPJ do cliente.
szRazaoSocialNomeDoCliente	PCHAR	30 (O, RT)	Razão Social / Nome do cliente.
szEndereco	PCHAR	79 (O, RT)	Endereço.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

O ECF deve estar ligado, com papel e com a jornada fiscal aberta ou sem redução Z pendente.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Fiscal_Abrir_CupomEX("12345678901230", "Empresa  
XYZ", "");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Fiscal_Abrir_CupomEX("12345678901230", "Empresa  
XYZ", "")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Fiscal_Abrir_CupomEX('12345678901230', 'Empresa  
XYZ', '');
```

**4.2.25 EPSON\_ESC\_ECF\_Fiscal\_Vender\_Item**

Esta função vende um item no Cupom Fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Vender_Item( szCodigo:Pchar; szDescricao:Pchar; szAliquota:Pchar;  
szUnidadeDeMedida:Pchar; dwQuantidade:Pchar;  
dwCasasDecimaisQuantidade:Pchar; dwPrecoUnidade:Pchar;  
dwCasasDecimaisPreco:Pchar; szArredondaTrunca:Pchar;  
szNumerolItem:Pchar; szValorItem:Pchar;  
szValorSubtotal:Pchar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCodigo	PCHAR	14 (A)	Código do item.
szDescricao	PCHAR	233 (A)	Descrição do item.
szAliquota	PCHAR	4(A)	Alíquota do item. Para alíquotas tributadas, utilizar 'T' ou 'S' + índice na tabela cadastrada no ECF (1 a 30). Para alíquotas não tributadas, utilizar I, F, N, IS, FS, NS + índice (1 a 3).
szUnidadeDeMedida	PCHAR	3 (A)	Símbolo da unidade.
szQuantidade	PCHAR	7 (N)	Quantidade do item.
szCasasDecimaisQuantidade	PCHAR	1 (N)	Número de casas decimais do valor informado no campo quantidade.
szPrecoUnidade	PCHAR	8 (N)	Preço unitário.
szCasasDecimaisPreco	PCHAR	1 (N)	Número de casas decimais do valor informado no campo preço.
szArredondaTrunca	PCHAR	1 (A)	Indicador do tipo de cálculo. 'A' = Arredondado, 'T' = Truncado.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szNumerolItem	PCHAR	4 (N)	Número do Item que foi registrado.
szValorItem	PCHAR	9 (N)	Valor do item registrado.
szValorSubtotal	PCHAR	14 (N)	Valor do subtotal do cupom.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szNumeroItem[4], szValorItem[9], szValorSubtotal[14];  
Retorno = EPSON_ESC_ECF_Fiscal_Vender_Item( "78912341234", "Produto XYZ",  
"T1", "UNI", "100", "2", "100", "2", "T", szNumeroItem, szValorItem,  
szValorSubtotal);
```

**Exemplo em C#:**

```
StringBuilder szNumeroItem = new StringBuilder(4, 4);  
StringBuilder szValorItem = new StringBuilder(9, 9);  
StringBuilder szValorSubtotal = new StringBuilder(14, 14);  
Retorno = EPSON_ESC_ECF_Fiscal_Vender_Item( "78912341234", "Produto XYZ",  
"T1", "UNI", "100", "2", "100", "2", "T", szNumeroItem, szValorItem,  
szValorSubtotal);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szNumeroItem As String  
Dim szValorItem As String  
Dim szValorSubtotal As String  
szNumeroItem = Space(4)  
szValorItem = Space(9)  
szValorSubtotal = Space(14)  
Retorno = EPSON_ESC_ECF_Fiscal_Vender_Item("78912341234", "Produto XYZ",  
"T1", "UNI", "100", "2", "100", "2", "T", szNumeroItem, szValorItem,  
szValorSubtotal)
```

**Exemplo em Delphi 7:**

```
szNumeroItem: array[0..4] of Char;  
szValorItem: array[0..9] of Char;  
szValorSubtotal: array[0..14] of Char;  
Retorno := EPSON_ESC_ECF_Fiscal_Vender_Item('78912341234', 'Produto XYZ',  
'T1', 'UNI', '100', '2', '100', '2', 'T', szNumeroItem, szValorItem,  
szValorSubtotal);
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szNumeroItem := AnsiString(StringOfChar(' ', 4));  
szValorItem := AnsiString(StringOfChar(' ', 9));  
szValorSubtotal := AnsiString(StringOfChar(' ', 14));  
Retorno := EPSON_ESC_ECF_Fiscal_Vender_Item('78912341234', 'Produto XYZ',  
'T1', 'UNI', '100', '2', '100', '2', 'T', PAnsiChar(szNumeroItem),  
PAnsiChar(szValorItem), PAnsiChar(szValorSubtotal));
```

**4.2.26 EPSON\_ESC\_ECF\_Fiscal\_Vender\_ItemEX**

Esta função vende um item no Cupom Fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Vender_ItemEX( szCodigo:Pchar; szDescricao:Pchar; szAliquota:Pchar;  
szUnidadeDeMedida:Pchar; dwQuantidade:Pchar;  
dwCasasDecimaisQuantidade:Pchar; dwPrecoUnidade:Pchar;  
dwCasasDecimaisPreco:Pchar; szArredondaTrunca:Pchar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCodigo	PCHAR	14 (A)	Código do item.
szDescricao	PCHAR	233 (A)	Descrição do item.
szAliquota	PCHAR	4(A)	Alíquota do item. Para alíquotas tributadas, utilizar 'T' ou 'S' + índice na tabela cadastrada no ECF (1 a 30). Para alíquotas não tributadas, utilizar I, F, N, IS, FS, NS + índice (1 a 3).
szUnidadeDeMedida	PCHAR	3 (A)	Símbolo da unidade
szQuantidade	PCHAR	7 (N)	Quantidade do item.
szCasasDecimaisQuantidade	PCHAR	1 (N)	Número de casas decimais do valor informado no campo quantidade.
szPrecoUnidade	PCHAR	8 (N)	Preço unitário.
szCasasDecimaisPreco	PCHAR	1 (N)	Número de casas decimais do valor informado no campo preço.
szArredondaTrunca	PCHAR	1 (A)	Indicador do tipo de cálculo. 'A' = Arredondado, 'T' = Truncado.

**Saídas:**

Nenhuma.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.



---

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

```
Retorno = EPSON_ESC_ECF_Fiscal_Vender_ItemEX( "78912341234", "Produto  
XYZ", "T1", "UNI", "100", "2", "100", "2", "T");
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Fiscal_Vender_ItemEX("78912341234", "Produto XYZ",  
"T1", "UNI", "100", "2", "100", "2", "T")
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Fiscal_Vender_ItemEX('78912341234', 'Produto  
XYZ', 'T1', 'UNI', '100', '2', '100', '2', 'T');
```

**4.2.27 EPSON\_ESC\_ECF\_Fiscal\_Cancelar\_Item**

Esta função cancela um item do cupom fiscal ou não fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Cancelar_Item(szNumerolItem:PChar; szValorSubtotal:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szNumerolItem	PCHAR	3 (N)	Número do item que será cancelado.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szValorSubtotal	PCHAR	14 (N)	Valor do subtotal do cupom.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szValorSubtotal[14];  
Retorno = EPSON_ESC_ECF_Fiscal_Cancelar_Item( "1", szValorSubtotal);
```

**Exemplo em C#:**


```
StringBuilder szValorSubtotal = new StringBuilder(14, 14);  
Retorno = EPSON_ESC_ECF_Fiscal_Cancelar_Item( "1", szValorSubtotal);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szValorSubtotal As String  
szValorSubtotal = Space(14)  
Retorno = EPSON_ESC_ECF_Fiscal_Cancelar_Item("1", szValorSubtotal)
```

**Exemplo em Delphi 7:**

```
szValorSubtotal: array[0..14] of Char;  
Retorno := EPSON_ESC_ECF_Fiscal_Cancelar_Item('1', szValorSubtotal);
```

 EXCEED YOUR VISION	Manual da Interface Epson		
	Ver.: 4.9.0	Data: 10/02/2017	Pg.: 59/406

---

### Exemplo em Delphi 2010 / Delphi XE:



```
szValorSubtotal := AnsiString(StringOfChar(' ', 14));
Retorno := EPSON_ESC_ECF_Fiscal_Cancelar_Item('1',
PAnsiChar(szValorSubtotal));
```

**4.2.28 EPSON\_ESC\_ECF\_Fiscal\_Cancelar\_ItemEX**

Esta função cancela um item do cupom fiscal ou não fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Cancelar_ItemEX(szNumerolItem:PChar):Integer; StdCall; External  
      'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szNumerolItem	PCHAR	3 (N)	Número do item que será cancelado.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

Comando disponível apenas para ECF modelo TM-T800F.

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

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

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Fiscal_Cancelar_ItemEX("1")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

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

**4.2.29 EPSON\_ESC\_ECF\_Fiscal\_Pagamento**

Esta função processa o pagamento do Cupom Fiscal ou não fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Pagamento(szIndice:PChar; szValor:PChar; szParcelas:PChar;  
szAdicional:PChar; szCodigo:PChar;  
szValorFaltante:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice do pagamento cadastrado no ECF.
szValor	PCHAR	13 (N)	Valor do pagamento.
szParcelas	PCHAR	2 (N)	Número de parcelas do pagamento.
szAdicional	PCHAR	84 (O,A)	Informações adicionais do pagamento.
szCodigo	PCHAR	2 (O,N)	Código do meio de pagamento: 1 – Dinheiro. 2 – Cheque. 3 - Cartão de Crédito. 4 - Cartão de Débito. 5 - Cartão Refeição/Alimentação. 6 - Vale Refeição/Alimentação (em papel). 7 – Outros.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szValorFaltante	PCHAR	14 (N)	Valor restante para o pagamento total do cupom.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szValorFaltante[14];  
Retorno = EPSON_ESC_ECF_Fiscal_Pagamento( "1", "1000", "1", "", "1",  
szValorFaltante);
```

**Exemplo em C#:**

```
StringBuilder szValorFaltante = new StringBuilder(14, 14);  
Retorno = EPSON_ESC_ECF_Fiscal_Pagamento( "1", "1000", "1", "", "1",  
szValorFaltante);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szValorFaltante As String  
szValorFaltante = Space(14)  
Retorno = EPSON_ESC_ECF_Fiscal_Pagamento("1", "1000", "1", "", "1",  
szValorFaltante)
```

**Exemplo em Delphi 7:**

```
szValorFaltante: array[0..14] of Char;  
Retorno := EPSON_ESC_ECF_Fiscal_Pagamento('1', '1000', '1', '', '1',  
szValorFaltante);
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szValorFaltante := AnsiString(StringOfChar(' ', 14));  
Retorno := EPSON_ESC_ECF_Fiscal_Pagamento ('1', '1000', '1', '', '1',  
PAnsiChar( szValorFaltante));
```

**4.2.30 EPSON\_ESC\_ECF\_Fiscal\_PagamentoEX**

Esta função processa o pagamento do Cupom Fiscal ou não fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_PagamentoEX(szIndice:PChar; szValor:PChar; szParcelas:PChar;  
szAdicional:PChar; szCodigo:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice do pagamento cadastrado no ECF.
szValor	PCHAR	13 (N)	Valor do pagamento.
szParcelas	PCHAR	2 (N)	Número de parcelas do pagamento.
szAdicional	PCHAR	84 (O,A)	Informações adicionais do pagamento.
szCodigo	PCHAR	2 (O,N)	Código do meio de pagamento: 1 – Dinheiro. 2 – Cheque. 3 - Cartão de Crédito. 4 - Cartão de Débito. 5 - Cartão Refeição/Alimentação. 6 - Vale Refeição/Alimentação (em papel). 7 – Outros.

**Saídas:**

Nenhuma.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Fiscal_PagamentoEX( "1", "1000", "1", "", "1");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Fiscal_PagamentoEX("1", "1000", "1", "", "1")
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Fiscal_PagamentoEX('1', '1000', '1', '', '1');
```



**4.2.31 EPSON\_ESC\_ECF\_Fiscal\_Fechar\_Cupom**

Esta função fecha o cupom fiscal e corta o papel, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Fechar_Cupom(szAdicional:PChar; szGuilhotina:PChar;  
szInformacoes:PChar; szCOO:PChar; szDataHora:PChar;  
szVendaBruta:PChar; szIndices:PChar; szCodigos:PChar;  
szValores:PChar; szParcelas:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szAdicional	PCHAR	1 (N)	Imprime cupom adicional. <b>0</b> – Não Imprime, <b>1</b> - Imprime
szGuilhotina	PCHAR	1 (N)	Aciona Guilhotina. <b>0</b> – Não Aciona, <b>1</b> - Aciona
szInformacoes	PCHAR	457(O,A)	Informações adicionais do cupom (8 linhas com até 57 caracteres cada).

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do cupom que foi encerrado.
szDataHora	PCHAR	16 (A)	Data e Hora de fechamento do cupom.
szVendaBruta	PCHAR	15 (N)	Venda Bruta.
szIndices	PCHAR	60 (N)	Sequencial do meio de pagamento no cupom (até 20 pagamentos).
szCodigos	PCHAR	60 (N)	Número do meio de pagamento (até 20 pagamentos).
szValores	PCHAR	280 (N)	Valor recebido no meio de pagamento (até 20 pagamentos).
szParcelas	PCHAR	60 (N)	Número de parcelas utilizadas no meio de pagamento (até 20 pagamentos).

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e o cupom com os devidos pagamentos já efetuados.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szCOO[10], szDataHora[16], szVendaBruta[15], szIndices[60],  
szCodigos[60], szValores[280], szParcelas[60];  
Retorno = EPSON_ESC_ECF_Fiscal_Fechar_Cupom( "0", "1", "Obrigado e Volte  
Sempre!", szCOO, szDataHora, szVendaBruta, szIndices, szCodigos,  
szValores, szParcelas);
```

---

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);
StringBuilder szDataHora = new StringBuilder(16, 16);
StringBuilder szVendaBruta = new StringBuilder(15, 15);
StringBuilder szIndices = new StringBuilder(60, 60);
StringBuilder szCodigos = new StringBuilder(60, 60);
StringBuilder szValores = new StringBuilder(280, 280);
StringBuilder szParcelas = new StringBuilder(60, 60);
Retorno = EPSON_ESC_ECF_Fiscal_Fechar_Cupom("0", "1", "Obrigado e Volte
Sempre!", szCOO, szDataHora, szVendaBruta, szIndices, szCodigos,
szValores, szParcelas);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String
Dim szDataHora As String
Dim szVendaBruta As String
Dim szIndices As String
Dim szCodigos As String
Dim szValores As String
Dim szParcelas As String
szCOO = Space(10)
szDataHora = Space(16)
szVendaBruta = Space(15)
szIndices = Space(60)
szCodigos = Space(60)
szValores = Space(280)
szParcelas = Space(60)
Retorno = EPSON_ESC_ECF_Fiscal_Fechar_Cupom("0", "1", "Obrigado e Volte
Sempre!", szCOO, szDataHora, szVendaBruta, szIndices, szCodigos,
szValores, szParcelas)
```

---

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;
szDataHora: array[0..16] of Char;
szVendaBruta: array[0..15] of Char;
szIndices: array[0..60] of Char;
szCodigos: array[0..60] of Char;
szValores: array[0..280] of Char;
szParcelas: array[0..60] of Char;
Retorno := EPSON_ESC_ECF_Fiscal_Fechar_Cupom('0', '1', 'Obrigado e Volte
Sempre!', szCOO, szDataHora, szVendaBruta, szIndices, szCodigos,
szValores, szParcelas);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));
szDataHora := AnsiString(StringOfChar(' ', 16));
szVendaBruta := AnsiString(StringOfChar(' ', 15));
szIndices := AnsiString(StringOfChar(' ', 60));
szCodigos := AnsiString(StringOfChar(' ', 60));
szValores := AnsiString(StringOfChar(' ', 280));
szParcelas := AnsiString(StringOfChar(' ', 60));
Retorno := EPSON_ESC_ECF_Fiscal_Fechar_Cupom('0', '1', 'Obrigado e Volte
Sempre!', PAnsiChar(szCOO), PAnsiChar(szDataHora),
PAnsiChar(szVendaBruta), PAnsiChar(szIndices), PAnsiChar(szCodigos),
PAnsiChar(szValores), PAnsiChar(szParcelas));
```

**4.2.32 EPSON\_ESC\_ECF\_Fiscal\_Fechar\_CupomEX**

Esta função fecha o cupom fiscal e corta o papel, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Fechar_CupomEX(szAdicional:PChar; szGuilhotina:PChar;  
szInformacoes:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szAdicional	PCHAR	1 (N)	Imprime cupom adicional. <b>0</b> – Não Imprime, <b>1</b> - Imprime
szGuilhotina	PCHAR	1 (N)	Aciona Guilhotina. <b>0</b> – Não Aciona, <b>1</b> - Aciona
szInformacoes	PCHAR	457(O,A)	Informações adicionais do cupom (8 linhas com até 57 caracteres cada).

**Saídas:**

Nenuma.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e o cupom com os devidos pagamentos já efetuados.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Fiscal_Fechar_CupomEX("0", "1", "Obrigado e Volte  
Sempre!");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Fiscal_Fechar_CupomEX("0", "1", "Obrigado e Volte  
Sempre!")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Fiscal_Fechar_CupomEX('0', '1', 'Obrigado e Volte  
Sempre!');
```

**4.2.33 EPSON\_ESC\_ECF\_Fiscal\_Cancelar\_Cupom**

Esta função cancela um cupom fiscal ou não fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Cancelar_Cupom(szCOO:PChar):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	9 (N)	COO do cupom fiscal ou comprovante não fiscal que será cancelado.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Fiscal_Cancelar_Cupom("100");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Fiscal_Cancelar_Cupom("100")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Fiscal_Cancelar_Cupom('100');
```

**4.2.34 EPSON\_ESC\_ECF\_Desconto\_A acrescimo\_Item**

Esta função efetua desconto ou acréscimo em qualquer item do cupom fiscal ou comprovante não fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Desconto_A acrescimo_Item(szOperacao:PChar; szTipo:PChar; szValor:PChar;  
szItem:PChar; szValorItem:PChar; szValorSubtotal:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szOperacao	PCHAR	1 (N)	Operação que será executada: <b>0</b> - Desconto. <b>1</b> – Acréscimo.
szTipo	PCHAR	1 (N)	Tipo da operação que será realizada: <b>0</b> – Em percentual (%). <b>1</b> – Em valor absoluto.
szValor	PCHAR	13 (N)	Valor da operação.
szItem	PCHAR	3 (O, N)	Número do item (Se nulo, será considerado o último item registrado).

**Saídas:**

Variável	Tipo	Tam.	Descrição
szValorItem	PCHAR	14 (N)	Valor atualizado do item.
szValorSubtotal	PCHAR	14 (N)	Valor atualizado do sub total.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szValorItem[14], szValorSubtotal[14];  
Retorno = EPSON_ESC_ECF_Desconto_A acrescimo_Item("0", "0", "1000", "1",  
szValorItem, szValorSubtotal);
```

**Exemplo em C#:**

```
StringBuilder szValorItem = new StringBuilder(14, 14);  
StringBuilder szValorSubtotal = new StringBuilder(14, 14);  
Retorno = EPSON_ESC_ECF_Desconto_Acrescimo_Item("0", "0", "1000", "1",  
szValorItem, szValorSubtotal);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szValorItem As String  
Dim szValorSubtotal As String  
szValorItem = Space(14)  
szValorSubtotal = Space(14)  
Retorno = EPSON_ESC_ECF_Desconto_Acrescimo_Item( "0", "0", "1000", "1",  
szValorItem, szValorSubtotal)
```

**Exemplo em Delphi 7:**

```
szValorItem: array[0..14] of Char;  
szValorSubtotal: array[0..14] of Char;  
Retorno := EPSON_ESC_ECF_Desconto_Acrescimo_Item('0', '0', '1000', '1',  
szValorItem, szValorSubtotal);
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szValorItem := AnsiString(StringOfChar(' ', 14));  
szValorSubtotal := AnsiString(StringOfChar(' ', 14));  
Retorno:= EPSON_ESC_ECF_Desconto_Acrescimo_Item('0', '0', '1000', '1',  
PAnsiChar(szValorItem), PAnsiChar(szValorSubtotal)) ;
```

**4.2.35 EPSON\_ESC\_ECF\_Desconto\_A acrescimo\_ItemEX**

Esta função efetua desconto ou acréscimo em qualquer item do cupom fiscal ou comprovante não fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Desconto_A acrescimo_ItemEX(szOperacao:PChar; szTipo:PChar; szValor:PChar;  
szItem:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szOperacao	PCHAR	1 (N)	Operação que será executada: <b>0</b> - Desconto. <b>1</b> – Acréscimo.
szTipo	PCHAR	1 (N)	Tipo da operação que será realizada: <b>0</b> – Em percentual (%). <b>1</b> – Em valor absoluto.
szValor	PCHAR	13 (N)	Valor da operação.
szItem	PCHAR	3 (O, N)	Número do item (Se nulo, será considerado o último item registrado).

**Saídas:**

Nenuma.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.


Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Desconto_A acrescimo_ItemEX("0", "0", "1000", "1");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Desconto_A acrescimo_ItemEX("0", "0", "1000", "1")
```

 EXCEED YOUR VISION	Manual da Interface Epson		
	Ver.: 4.9.0	Data: 10/02/2017	Pg.: 72/406

---

#### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



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



**4.2.36 EPSON\_ESC\_ECF\_Cancela\_Desconto\_A acrescimo\_Item**

Esta função cancela o desconto ou acréscimo aplicado sobre qualquer item vendido no cupom fiscal ou comprovante não fiscal atual, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_Item(szOperacao:PChar; szItem:PChar;  
szValorItem:PChar; szValorSubtotal:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szOperacao	PCHAR	1 (N)	Operação que será executada: <b>0</b> - Desconto. <b>1</b> – Acréscimo.
szItem	PCHAR	3 (N)	Número do item.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szValorItem	PCHAR	14 (N)	Valor atualizado do item.
szValorSubtotal	PCHAR	14 (N)	Valor atualizado do sub total.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e o item referido deve possuir acréscimo ou desconto.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szValorItem[14], szValorSubtotal[14];  
Retorno = EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_Item("0", "1",  
szValorItem, szValorSubtotal);
```

**Exemplo em C#:**

```
StringBuilder szValorItem = new StringBuilder(14, 14);  
StringBuilder szValorSubtotal = new StringBuilder(14, 14);  
Retorno = EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_Item("0", "1",  
szValorItem, szValorSubtotal);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szValorItem As String
Dim szValorSubtotal As String
szValorItem = Space(14)
szValorSubtotal = Space(14)
Retorno = EPSON_ESC_ECF_Cancela_Desconto_Acrescimo_Item("0", "1",
szValorItem, szValorSubtotal)
```

---

**Exemplo em Delphi 7:**

```
szValorItem: array[0..14] of Char;
szValorSubtotal: array[0..14] of Char;
Retorno := EPSON_ESC_ECF_Cancela_Desconto_Acrescimo_Item('0', '1',
szValorItem, szValorSubtotal);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szValorItem := AnsiString(StringOfChar(' ', 14));
szValorSubtotal := AnsiString(StringOfChar(' ', 14));
Retorno := EPSON_ESC_ECF_Cancela_Desconto_Acrescimo_Item('0', '1',
PAnsiChar(szValorItem), PAnsiChar(szValorSubtotal)) ;
```

**4.2.37 EPSON\_ESC\_ECF\_Cancela\_Desconto\_A acrescimo\_ItemEX**

Esta função cancela o desconto ou acréscimo aplicado sobre qualquer item vendido no cupom fiscal ou comprovante não fiscal atual, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_ItemEX(szOperacao:PChar; szItem:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szOperacao	PCHAR	1 (N)	Operação que será executada: <b>0</b> - Desconto. <b>1</b> – Acréscimo.
szItem	PCHAR	3 (N)	Número do item.

**Saídas:**

Nenhuma.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e o item referido deve possuir acréscimo ou desconto.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_ItemEX("0", "1");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_ItemEX("0", "1")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_ItemEX('0', '1');
```

**4.2.38 EPSON\_ESC\_ECF\_Desconto\_A acrescimo\_Subtotal**

Esta função efetua um desconto ou acréscimo no subtotal do cupom fiscal ou comprovante não fiscal atual, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Desconto_A acrescimo_Subtotal(szOperacao:PChar; szTipo:PChar; szValor:PChar;  
szValorSubtotal:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szOperacao	PCHAR	1 (N)	Operação que será executada: <b>0</b> - Desconto. <b>1</b> – Acréscimo.
szTipo	PCHAR	1 (N)	Tipo da operação que será realizada: <b>0</b> – Em percentual (%). <b>1</b> – Em valor absoluto.
szValor	PCHAR	13 (N)	Valor da operação.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szValorSubtotal	PCHAR	14 (N)	Valor atualizado do sub total.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szValorSubtotal[14];  
Retorno = EPSON_ESC_ECF_Desconto_A acrescimo_Subtotal("1","1","1000",  
szValorSubtotal);
```

**Exemplo em C#:**

```
StringBuilder szValorSubtotal = new StringBuilder(14, 14);  
Retorno = EPSON_ESC_ECF_Desconto_A acrescimo_Subtotal("1","1","1000",  
szValorSubtotal);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szValorSubtotal As String  
szValorSubtotal = Space(14)  
Retorno = EPSON_ESC_ECF_Desconto_Acrescimo_Subtotal("1", "1", "1000",  
szValorSubtotal)
```

---

**Exemplo em Delphi 7:**

```
szValorSubtotal: array[0..14] of Char;  
Retorno := EPSON_ESC_ECF_Desconto_Acrescimo_Subtotal('1','1','1000',  
szValorSubtotal);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szValorSubtotal := AnsiString(StringOfChar(' ', 14));  
Retorno := EPSON_ESC_ECF_Desconto_Acrescimo_Subtotal('1', '1', '1000',  
PAnsiChar(szValorSubtotal)) ;
```

**4.2.39 EPSON\_ESC\_ECF\_Desconto\_A acrescimo\_SubtotalEX**

Esta função efetua um desconto ou acréscimo no subtotal do cupom fiscal ou comprovante não fiscal atual, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Desconto_A acrescimo_SubtotalEX(szOperacao:PChar; szTipo:PChar;  
szValor:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szOperacao	PCHAR	1 (N)	Operação que será executada: 0 - Desconto. 1 – Acréscimo.
szTipo	PCHAR	1 (N)	Tipo da operação que será realizada: 0 – Em percentual (%). 1 – Em valor absoluto.
szValor	PCHAR	13 (N)	Valor da operação.

**Saídas:**

Nenhuma.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Desconto_A acrescimo_SubtotalEX("1", "1", "1000");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Desconto_A acrescimo_SubtotalEX("1", "1", "1000")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Desconto_A acrescimo_SubtotalEX('1', '1', '1000');
```

**4.2.40 EPSON\_ESC\_ECF\_Cancela\_Desconto\_A acrescimo\_Subtotal**

Esta função cancela o desconto ou acréscimo aplicado sobre o subtotal do cupom fiscal ou comprovante não fiscal atual, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_Subtotal(szOperacao:PChar;  
szValorSubtotal:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szOperacao	PCHAR	1 (N)	Operação que será executada: 0 - Desconto. 1 – Acréscimo.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szValorSubtotal	PCHAR	14 (N)	Valor atualizado do sub total.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documento fiscal aberto, o ECF com papel, e o documento fiscal deve possuir acréscimo ou desconto em subtotal.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szValorSubtotal[14];  
Retorno = EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_Subtotal("1",  
szValorSubtotal);
```

**Exemplo em C#:**

```
StringBuilder szValorSubtotal = new StringBuilder(14, 14);  
Retorno = EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_Subtotal("1",  
szValorSubtotal);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szValorSubtotal As String  
szValorSubtotal = Space(14)  
Retorno = EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_Subtotal("1",  
szValorSubtotal)
```

---

**Exemplo em Delphi 7:**

```
szValorSubtotal: array[0..14] of Char;  
Retorno := EPSON_ESC_ECF_Cancela_Desconto_Acrescimo_Subtotal('1',  
szValorSubtotal);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szValorSubtotal := AnsiString(StringOfChar(' ', 14));  
Retorno := EPSON_ESC_ECF_Cancela_Desconto_Acrescimo_Subtotal('1',  
PAnsiChar(szValorSubtotal)) ;
```



**4.2.41 EPSON\_ESC\_ECF\_Cancela\_Desconto\_A acrescimo\_SubtotalEX**

Esta função cancela o desconto ou acréscimo aplicado sobre o subtotal do cupom fiscal ou comprovante não fiscal atual, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_SubtotalEX(szOperacao:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szOperacao	PCHAR	1 (N)	Operação que será executada: 0 - Desconto. 1 - Acréscimo.

**Saídas:**

Nenhuma.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documento fiscal aberto, o ECF com papel, e o documento fiscal deve possuir acréscimo ou desconto em subtotal.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_SubtotalEX("1");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_SubtotalEX("1")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Cancela_Desconto_A acrescimo_SubtotalEX('1');
```

**4.2.42 EPSON\_ESC\_ECF\_Cancela\_Cupom\_Fiscal\_Nao\_Fiscal**

Esta função cancela o cupom fiscal ou comprovante não fiscal em andamento, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Cancela_Cupom_Fiscal_Nao_Fiscal():Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documento fiscal aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Cancela_Cupom_Fiscal_Nao_Fiscal();
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Cancela_Cupom_Fiscal_Nao_Fiscal()
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Cancela_Cupom_Fiscal_Nao_Fiscal();
```

**4.2.43 EPSON\_ESC\_ECF\_Cancela\_Item\_Parcial**

Esta função cancela parcialmente um item, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Cancela_Item_Parcial(szNumero:PChar; szQuantidade:PChar;  
                                             szValorLiquido:PChar; szValorSubtotal:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szNumero	PCHAR	3 (N)	Número do item que será parcialmente cancelado.
szQuantidade	PCHAR	7 (N)	Quantidade que será cancelada.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szValorItem	PCHAR	14 (N)	Valor atualizado do item.
szValorSubtotal	PCHAR	14 (N)	Valor atualizado do sub total.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szValorItem[14], szValorSubtotal[14];  
Retorno = EPSON_ESC_ECF_Cancela_Item_Parcial("1", "2000", szValorItem,  
szValorSubtotal);
```

**Exemplo em C#:**

```
StringBuilder szValorItem = new StringBuilder(14, 14);  
StringBuilder szValorSubtotal = new StringBuilder(14, 14);  
Retorno = EPSON_ESC_ECF_Cancela_Item_Parcial("1", "2000", szValorItem,  
szValorSubtotal);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szValorItem As String  
Dim szValorSubtotal As String  
szValorItem = Space(14)  
szValorSubtotal = Space(14)  
Retorno = EPSON_ESC_ECF_Cancela_Item_Parcial("1", "2000", szValorItem,  
szValorSubtotal)
```

---

**Exemplo em Delphi 7:**

```
szValorItem: array[0..14] of Char;  
szValorSubtotal: array[0..14] of Char;  
Retorno := EPSON_ESC_ECF_Cancela_Item_Parcial('1', '2000', szValorItem,  
szValorSubtotal);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szValorItem := AnsiString(StringOfChar(' ', 14));  
szValorSubtotal := AnsiString(StringOfChar(' ', 14));  
Retorno := EPSON_ESC_ECF_Cancela_Item_Parcial('1', '2000',  
PAnsiChar(szValorItem), PAnsiChar(szValorSubtotal)) ;
```

**4.2.44 EPSON\_ESC\_ECF\_Cancela\_Item\_ParcialEX**

Esta função cancela parcialmente um item, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Cancela_Item_ParcialEX(szNumero:PChar; szQuantidade:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szNumero	PCHAR	3 (N)	Número do item que será parcialmente cancelado.
szQuantidade	PCHAR	7 (N)	Quantidade que será cancelada.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e ao menos um item registrado.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Cancela_Item_ParcialEX("1", "2000");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Cancela_Item_ParcialEX("1", "2000")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Cancela_Item_ParcialEX('1', '2000');
```

**4.2.45 EPSON\_ESC\_ECF\_Fiscal\_Abrir\_Cupom\_CFE**

Esta função efetua a abertura de um cupom fiscal eletrônico, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Abrir_Cupom_CFE(szCRT:PChar; szRegTribISSQN:PChar; szOrig:PChar;  
szCST:PChar; szCSOSN:PChar; szMUNFG:PChar;  
szListServ:PChar; szServTribMun:PChar; szNatOp:PChar;  
szIndIncFisc:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCRT	PCHAR	1 (N)	Ver campo C14 do CF-e-ECF.
szRegTribISSQN	PCHAR	1 (N)	Ver campo C15 do CF-e-ECF.
szOrig	PCHAR	1 (N)	Ver campos N06 do CF-e-ECF.
szCST	PCHAR	2 (N)	Ver campo N07 do CF-e-ECF Esse campo será utilizado exclusivamente com os valores 00, 20 ou 90 quando a venda for via totalizador parcial com alíquota. Para II, FF e NN o ECF utilizará respectivamente os valores 40, 60 e 41.
szCSOSN	PCHAR	3 (N)	Ver campos N10 do CF-e-ECF
szMUNFG	PCHAR	7 (N)	Ver campo U06 do CF-e-ECF
szListServ	PCHAR	5 (A)	Ver campo U07 do CF-e-ECF
szServTribMun	PCHAR	20 (A)	Ver campo U08 do CF-e-ECF
szNatOp	PCHAR	2 (N)	Ver campo U09 do CF-e-ECF
szIndIncFisc	PCHAR	1 (N)	Ver campo U10 do CF-e-ECF

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

O ECF deve estar ligado, com papel e com a jornada fiscal aberta ou sem redução Z pendente.  
Comando disponível apenas para ECF modelo TM-T800F.

**4.2.46 EPSON\_ESC\_ECF\_Fiscal\_Vender\_Item\_Detalhado**

Esta função vende um item no Cupom Fiscal Eletrônico, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Fiscal_Vender_Item_Detalhado(szCodigo:PChar; szDescricao:PChar;  
szAliquota:PChar; szUnidadeDeMedida:PChar;  
szQuantidade:PChar; szCasasDecimaisQuantidade:PChar;  
szPrecoUnidade:PChar; szCasasDecimaisPreco:PChar;  
szArredondaTrunca:PChar; szEAN:PChar; szNCM:PChar;  
szCFOP:PChar; szInfAdProd:PChar; szOrig:PChar; szCST:PChar;  
szCSOSN:PChar; szMUNFG:PChar; szListServ:PChar;  
szServTribMun:PChar; szNatOp:PChar; szIndIncFisc:PChar;  
szNumerolItem:PChar; szValorItem:PChar;  
szValorSubtotal:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCodigo	PCHAR	14 (A)	Código do item.
szDescricao	PCHAR	233 (A)	Descrição do item.
szAliquota	PCHAR	4(A)	Alíquota do item. Para alíquotas tributadas, utilizar 'T' ou 'S' + índice na tabela cadastrada no ECF (1 a 30). Para alíquotas não tributadas, utilizar I, F, N, IS, FS, NS + índice (1 a 3).
szUnidadeDeMedida	PCHAR	3 (A)	Símbolo da unidade.
szQuantidade	PCHAR	7 (N)	Quantidade do item.
szCasasDecimaisQuantidade	PCHAR	1 (N)	Número de casas decimais do valor informado no campo quantidade.
szPrecoUnidade	PCHAR	8 (N)	Preço unitário.
szCasasDecimaisPreco	PCHAR	1 (N)	Número de casas decimais do valor informado no campo preço.
szArredondaTrunca	PCHAR	1 (A)	Indicador do tipo de cálculo. 'A' = Arredondado, 'T' = Truncado.
szEAN	PCHAR	14 (O,N)	Ver campo I03 do CF-e-ECF
szNCM	PCHAR	8 (O,A)	Ver campo I05 do CF-e-ECF
szCFOP	PCHAR	4 (O,N)	Ver campo I06 do CF-e-ECF
szInfAdProd	PCHAR	500 (O,A)	Ver campo V01 do CF-e-ECF
szOrig	PCHAR	1 (O,N)	Ver campos N06 do CF-e-ECF
szCST	PCHAR	2 (O,N)	Ver campos N07 do CF-e-ECF
szCSOSN	PCHAR	3 (O,N)	Ver campos N10 do CF-e-ECF
szMUNFG	PCHAR	7 (O,N)	Ver campo U06 do CF-e-ECF
szListServ	PCHAR	5 (O,A)	Ver campo U07 do CF-e-ECF
szServTribMun	PCHAR	20 (O,A)	Ver campo U08 do CF-e-ECF
szNatOp	PCHAR	2 (O,N)	Ver campo U09 do CF-e-ECF
szIndIncFisc	PCHAR	1 (O,N)	Ver campo U10 do CF-e-ECF

**Saídas:**

Variável	Tipo	Tam.	Descrição
szNumerolItem	PCHAR	4 (N)	Número do Item que foi registrado.
szValorItem	PCHAR	9 (N)	Valor do item registrado.
szValorSubtotal	PCHAR	14 (N)	Valor do subtotal do cupom.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.



**Grupo do Comprovante Não-Fiscal**

Estas funções são usadas para imprimir um Cupom Não-Fiscal:

**4.2.47 EPSON\_NaoFiscal\_Abrir\_Comprovante**

Abre um Comprovante Não-Fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Abrir_Comprovante( pszCPFCNPJ:PChar; pszRazaoSocialNomeCliente:PChar;  
                                             pszEnderecoLinha1:PChar; pszEnderecoLinha2:PChar;  
                                             dwPosicaoDadosImpressao:Integer):Integer; StdCall; External  
                                             'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszCPFCNPJ	PCHAR	20 (O,RT)	Número CNPJ / CPF do cliente.
pszRazaoSocialNomeCliente	PCHAR	30 (O,RT)	Razão Social / Nome do cliente.
pszEnderecoLinha1	PCHAR	40 (O,RT)	Endereço – Linha 01.
pszEnderecoLinha2	PCHAR	39 (O,RT)	Endereço – Linha 02.
dwPosicaoDadosImpressão	INTEGER	-	Posição onde os dados serão impressos: 0 - Não Altera. 1 - Desabilitado. 2 - No cabeçalho.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel. Caso os dados do consumidor sejam nulos, os campos referentes a estes serão omitidos do cupom não fiscal.

---

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

```
Retorno = EPSON_NaoFiscal_Abrir_Comprovante ( "", "", "", "", 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Abrir_Comprovante ( "", "", "", "", 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Abrir_Comprovante ( '', '', '', '', 1 );
```

**4.2.48 EPSON\_NaoFiscal\_Vender\_Item**

Esta função registra no Comprovante um item não-fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Vender_Item( pszNumeroOperacao:PChar; pszValor:PChar;  
dwCasasDecimaisValor:Integer):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroOperacao	PCHAR	2 (N)	Número da operação não-fiscal.
pszValor	PCHAR	11 (N)	Valor da operação não-fiscal.
dwCasasDecimaisValor	INTEGER	-	Quantidade de casas decimais usadas.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel. Caso o número da operação seja igual a 1 ou 2, será efetuada uma sangria ou recebimento respectivamente.

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

```
Retorno = EPSON_NaoFiscal_Vender_Item ( "1", "15000", 3 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Vender_Item ( "1", "15000", 3 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Vender_Item ( '1', '15000', 3 );
```

**4.2.49 EPSON\_NaoFiscal\_Desconto\_A acrescimo\_Item**

Aplica desconto ou acréscimo em um item não-fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Desconto_A acrescimo_Item( pszValorDescontoAcrescimo:PChar;  
dwNumeroCasasDecimais:Integer;  
bTipoDescontoAcrescimo:Boolean;  
bTipoPercentagemMoeda:Boolean):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszValorDescontoAcrescimo	PCHAR	11 (N)	Valor do desconto ou acréscimo.
dwNumeroCasasDecimais	INTEGER	-	Número de casas decimais usado.
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - Desconto. <b>False</b> - Acréscimo.
bTipoPercentagemMoeda	BOOLEAN	-	<b>True</b> - Valor percentual. <b>False</b> - Valor em moeda.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto e ao menos um item vendido no cupom.

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

```
Retorno = EPSON_NaoFiscal_Desconto_A acrescimo_Item ( "9000", 3, true, false  
);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Desconto_A acrescimo_Item ( "9000", 3, true, false  
)
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Desconto_A acrescimo_Item ( '9000', 3, true,  
false );
```

**4.2.50 EPSON\_NaoFiscal\_Desconto\_A acrescimo\_ItemEX**

Esta função efetua desconto ou acréscimo em qualquer item do cupom não fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Desconto_A acrescimo_ItemEX( pszNumeroltem:PChar;  
                                                       pszValorDescontoAcrescimo:PChar;  
                                                       dwNumeroCasasDecimais:Integer;  
                                                       bTipoDescontoAcrescimo:Boolean;  
                                                       bTipoPercentagemMoeda:Boolean):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroltem	PCHAR	3 (N)	Número do item no cupom não fiscal.
pszValorDescontoAcrescimo	PCHAR	11 (N)	Valor do desconto ou acréscimo.
dwNumeroCasasDecimais	INTEGER	-	Número de casas decimais do valor.
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - Desconto. <b>False</b> - Acréscimo.
bTipoPercentagemMoeda	BOOLEAN	-	<b>True</b> - Valor percentual. <b>False</b> - Valor em moeda.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel e ao menos um item registrado.

Comando disponível apenas para ECF modelos TM-H6000 FBIII, TM-T81 FBIII e TM-T88 FBIII.

---

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

```
Retorno = EPSON_NaoFiscal_Desconto_Acrescimo_ItemEX ( "1", "2500", 3,  
true, false );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Desconto_Acrescimo_ItemEX ( "1", "2500", 3,  
true, false )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Desconto_Acrescimo_ItemEX ( '1', '2500', 3,  
true, false);
```

**4.2.51 EPSON\_NaoFiscal\_Desconto\_A acrescimo\_Subtotal**

Esta função aplica um desconto ou acréscimo no subtotal do Comprovante Não-Fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Desconto_A acrescimo_Subtotal( pszValorDescontoAcrescimo:PChar;  
dwNumeroCasasDecimais:Integer;  
bTipoDescontoAcrescimo:Boolean;  
bTipoPercentagemMoeda:Boolean):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszValorDescontoAcrescimo	PCHAR	11 (N)	Valor do desconto ou acréscimo.
dwNumeroCasasDecimais	INTEGER	-	Número de casas decimais.
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - Desconto. <b>False</b> - Acréscimo.
bTipoPercentagemMoeda	BOOLEAN	-	<b>True</b> - Valor percentual. <b>False</b> - Valor em moeda.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto e ao menos um item vendido no cupom.

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

```
Retorno = EPSON_NaoFiscal_Desconto_A acrescimo_Subtotal ( "9000", 3, true,  
false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Desconto_A acrescimo_Subtotal ( "9000", 3, true,  
false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Desconto_A acrescimo_Subtotal ( '9000', 3, true,  
false );
```

**4.2.52 EPSON\_NaoFiscal\_Pagamento**

Esta função processa o pagamento do Comprovante Não-Fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Pagamento( pszNumeroDescricaoPagamento:PChar; pszValorPagamento:PChar;  
dwCasasDecimaisPagamento:Integer; pszDescricaoLinha1:PChar;  
pszDescricaoLinha2:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroDescricaoPagamento	PCHAR	2 / 15	Número da Forma de Pagamento cadastrada na Impressora Fiscal ou descritivo da forma de pagamento.
pszValorPagamento	PCHAR	13	Valor do pagamento.
dwCasasDecimaisPagamento	INTEGER	-	Número de casas decimais utilizado.
pszDescricaoLinha1	PCHAR	40	Primeira linha de descrição (opcional).
pszDescricaoLinha2	PCHAR	40	Segunda linha de descrição (opcional).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto e ao menos um item vendido no cupom.

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

```
Retorno = EPSON_NaoFiscal_Pagamento ( "2", "5000", 3, "", "" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Pagamento ( "2", "5000", 3, "", "" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Pagamento ( '2', '5000', 3, '', '' );
```



**4.2.53 EPSON\_NaoFiscal\_Cancelar\_Item**

Esta função cancela um item no Comprovante Não-Fiscal, caso este item seja informado com um valor nulo, a função cancelará o ultimo item inserido no cupom.

**Sintaxe:**

```
function EPSON_NaoFiscal_Cancelar_Item(pszNumerolItem:PChar):Integer;StdCall; External InterfaceEpson.dll;
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumerolItem	PCHAR	3 (O,N)	Item a cancelar.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto e ao menos um item vendido no cupom.

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

```
Retorno = EPSON_NaoFiscal_Cancelar_Item ( "2" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Cancelar_Item ( "2" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Cancelar_Item ( '2' );
```

**4.2.54 EPSON\_NaoFiscal\_Cancelar\_Ultimo\_Item**

Esta função cancela o último item do Comprovante Não-Fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Cancelar_Ultimo_Item():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto e ao menos um item vendido no cupom.

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

```
Retorno = EPSON_NaoFiscal_Cancelar_Ultimo_Item ( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Cancelar_Ultimo_Item ( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Cancelar_Ultimo_Item ( );
```

**4.2.55 EPSON\_NaoFiscal\_Cancelar\_Desconto\_A acrescimo\_Item**

Esta função cancela o desconto ou acréscimo de um item não fiscal.

**Sintaxe:**

```
functionEPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_Item(bTipoDescontoAcrescimo:Boolean):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - Cancelar o desconto. <b>False</b> - Cancelar o acréscimo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel e o item referido deve possuir acréscimo ou desconto.

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

```
Retorno = EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_Item ( false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_Item ( false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_Item ( false );
```

**4.2.56 EPSON\_NaoFiscal\_Cancelar\_Desconto\_A acrescimo\_ItemEX**

Esta função cancela o desconto ou acréscimo aplicado sobre qualquer item no cupom não fiscal atual.

**Sintaxe:**

```
function EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_ItemEX( pszNumerolItem:PChar;  
                                                                bTipoDescontoAcrescimo:Boolean);  
Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumerolItem	PCHAR	3 (N)	Número do item.
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - cancelar o desconto. <b>False</b> - cancelar o acréscimo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel e o item referido deve possuir acréscimo ou desconto.

Comando disponível apenas para ECF modelos TM-H6000 FBIII, TM-T81 FBIII e TM-T88 FBIII.

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

```
Retorno = EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_ItemEX("2", false);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_ItemEX("2", false)
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_ItemEX('2', false);
```

**4.2.57 EPSON\_NaoFiscal\_Cancelar\_Desconto\_A acrescimo\_Subtotal**

Esta função cancela o desconto ou acréscimo aplicado sobre o subtotal do Comprovante Não Fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_Subtotal( bTipoDescontoAcrescimo:Boolean):  
Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bTipoDescontoAcrescimo	BOOLEAN	-	<b>True</b> - Cancelar o desconto. <b>False</b> - Cancelar o acréscimo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel e o cupom deve possuir acréscimo ou desconto.

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

```
Retorno = EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_Subtotal ( false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_Subtotal ( false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Cancelar_Desconto_A acrescimo_Subtotal ( false );
```

**4.2.58 EPSON\_NaoFiscal\_Cancelar\_Comprovante**

Esta função cancela o último Comprovante Não-Fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Cancelar_Comprovante():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto e o ECF com papel.

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

```
Retorno = EPSON_NaoFiscal_Cancelar_Comprovante ( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Cancelar_Comprovante ( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Cancelar_Comprovante ( );
```

**4.2.59 EPSON\_NaoFiscal\_Cancelar\_ComprovanteEX**

Esta função cancela o último Comprovante Não-Fiscal. Caso existam Comprovantes de Crédito ou Débito relacionados a este comprovante, os mesmos serão cancelados.

**Sintaxe:**

```
function EPSON_NaoFiscal_Cancelar_ComprovanteEX():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, comprovante não fiscal aberto e o ECF com papel.

O último documento impresso deverá ser um comprovante não fiscal ou um Comprovante de Crédito ou Débito vinculado ao último comprovante não fiscal.

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

```
Retorno = EPSON_NaoFiscal_Cancelar_ComprovanteEX( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Cancelar_ComprovanteEX( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Cancelar_ComprovanteEX( );
```

**4.2.60 EPSON\_NaoFiscal\_Fechar\_Comprovante**

Esta função fecha o Comprovante Não-Fiscal.

**Sintaxe:**

```
function EPSON_NaoFiscal_Fechar_Comprovante(bCortarPapel:Boolean):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bCortarPapel	BOOLEAN	-	<b>True</b> - Cortar o papel depois do rodapé. <b>False</b> - Não cortar o papel depois do rodapé.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel e caso o cupom não seja referente à sangria ou recebimento, deve possuir os devidos pagamentos.

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

```
Retorno = EPSON_NaoFiscal_Fechar_Comprovante( true );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Fechar_Comprovante( true )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Fechar_Comprovante( true );
```



**4.2.61 EPSON\_NaoFiscal\_Abrir\_CCD**

Esta função abre um Comprovante de Crédito ou Débito (TEF).

**Sintaxe:**

```
function EPSON_NaoFiscal_Abrir_CCD(pszNumeroDescricaoPagamento:PChar; pszValorPagamento:PChar;  
dwCasasDecimaisPagamento:Integer;pszParcelas:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroDescricaoPagamento	PCHAR	2 (N) / 15 (A)	Número da forma de pagamento ou descritivo da forma da forma de pagamento.
pszValorPagamento	PCHAR	13 (N)	Valor do pagamento.
dwCasasDecimaisPagamento	INTEGER	-	Número de casas decimais utilizada.
pszParcelas	PCHAR	6 (N)	Número de parcelas.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados, o ECF com papel, o valor e o número da forma de pagamento deve coincidir com o valor e a forma de pagamento no último cupom.

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

```
Retorno = EPSON_NaoFiscal_Abrir_CCD ( "3", "250000", 3, "1" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Abrir_CCD ( "3", "250000", 3, "1" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Abrir_CCD ( '3', '250000', 3, '1' );
```

**4.2.62 EPSON\_NaoFiscal\_Abrir\_Relatorio\_Gerencial**

Esta função abre um Relatório Gerencial.

**Sintaxe:**

```
function EPSON_NaoFiscal_Abrir_Relatorio_Gerencial(pszNumeroRelatorio:PChar):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroRelatorio	PCHAR	2	Número do Relatório Gerencial.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Documentos fechados e o ECF com papel.

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

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

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

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

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

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

**4.2.63 EPSON\_NaoFiscal\_Imprimir\_LinhaEX**

Esta função imprime linhas texto no Relatório Gerencial aberto ou no Comprovante de Crédito ou Débito aberto. Esse comando efetua a quebra automática de linha ao final de 56 caracteres da linha e também permite a quebra de linha utilizando-se os caracteres terminadores de linha 0x0D 0x0A.

**Sintaxe:**

```
function EPSON_NaoFiscal_Imprimir_LinhaEX( pszLinhaTexto:PChar ):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszLinhaTexto	PCHAR	-	Linha a ser impressa.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Um documento não fiscal aberto e o ECF com papel.

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

```
Retorno = EPSON_NaoFiscal_Imprimir_LinhaEX("Linha: 001\n Linha:  
002\nLinha: 003");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
LinhaTexto = "Linha: 001" & vbCrLf & "Linha: 002" & vbCrLf & "Linha: 003"  
Retorno = EPSON_NaoFiscal_Imprimir_LinhaEX(LinhaTexto);
```

**Exemplo em Delphi :**

```
LinhaTexto:= 'Linha: 001' +#13+#10 + 'Linha: 002 '+#13+#10 'Linha: 003';  
Retorno := EPSON_NaoFiscal_Imprimir_LinhaEX(pchar(LinhaTexto));
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
LinhaTexto:='Linha: 001' +#13+#10 +' Linha: 002 '+#13+#10+'Linha: 003';  
Retorno := EPSON_NaoFiscal_Imprimir_LinhaEX(PAnsiChar(LinhaTexto));
```

**4.2.64 EPSON\_NaoFiscal\_Imprimir\_Linha**

Esta função imprime uma linha texto no Relatório Gerencial aberto ou no Comprovante de Crédito ou Débito aberto.

**Sintaxe:**

```
function EPSON_NaoFiscal_Imprimir_Linha( pszImpressaoLinhaTexto:PChar ):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszImpressaoLinhaTexto	PCHAR	56 RT	Linha a ser impressa.

**Tags de Formatação:**

É possível formatar a linha e palavras a serem impressas para utilizar fontes em negrito, sublinhado, largura dupla, altura dupla, e cor invertida utilizando os seguintes tags de formatação antecedendo as palavras a serem impressas.

<AT> - Ativas todos atributos de formatação  
<AN> - Ativa negrito  
<AS> - Ativa sublinhado  
<AL> - Ativa largura dupla  
<AA> - Ativa altura Dupla  
<AI> - Ativa cor invertida  
<DT> - Desativa todos  
<DN> - Desativa negrito  
<DS> - Desativa sublinhado  
<DL> - Desativa largura dupla  
<DA> - Desativa altura dupla  
<DI> - Desativa cor invertida

Por padrão, a impressora já permite utilizar os tags para formatar uma linha texto inteira.

Para que seja possível utilizar formatação somente por palavra separadamente, é necessário habilitar a função de atributo nos jumpers de configuração da impressora. Para isto, basta entrar em modo de Intervenção Técnica, desligar a impressora e então alterar seus estados nas microchaves de seleção (Dip-Switch).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Um documento não fiscal aberto e o ECF com papel.

---

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

```
Retorno = EPSON_NaoFiscal_Imprimir_Linha ( "<AN>Linha Texto em negrito" );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Imprimir_Linha ( "<AN>Linha Texto em negrito " )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Imprimir_Linha ( '<AN>Linha Texto em negrito'  
);
```

**4.2.65 EPSON\_NaoFiscal\_Imprimir\_15Linhas**

Esta função imprime 15 linhas de texto no Relatório Gerencial aberto ou no Comprovante de Crédito ou Débito aberto.

**Sintaxe:**

```
function EPSON_NaoFiscal_Imprimir_15Linhas( pszLinhaTexto00:PChar; pszLinhaTexto01:PChar;  
      pszLinhaTexto02:PChar; pszLinhaTexto03:PChar;  
      pszLinhaTexto04:PChar; pszLinhaTexto05:  
      PChar;pszLinhaTexto06:PChar;pszLinhaTexto07:  
      PChar;pszLinhaTexto08:PChar;pszLinhaTexto09:  
      PChar;pszLinhaTexto10:PChar;pszLinhaTexto11:  
      PChar;pszLinhaTexto12:PChar;pszLinhaTexto13:  
      PChar;pszLinhaTexto14:PChar):Integer;StdCall;External  
      'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszLinhaTexto00	PCHAR	56 RT	Linha 1 a ser impressa.
pszLinhaTexto01	PCHAR	56 RT	Linha 2 a ser impressa.
pszLinhaTexto02	PCHAR	56 RT	Linha 3 a ser impressa.
pszLinhaTexto03	PCHAR	56 RT	Linha 4 a ser impressa.
pszLinhaTexto04	PCHAR	56 RT	Linha 5 a ser impressa.
pszLinhaTexto05	PCHAR	56 RT	Linha 6 a ser impressa.
pszLinhaTexto06	PCHAR	56 RT	Linha 7 a ser impressa.
pszLinhaTexto07	PCHAR	56 RT	Linha 8 a ser impressa.
pszLinhaTexto08	PCHAR	56 RT	Linha 9 a ser impressa.
pszLinhaTexto09	PCHAR	56 RT	Linha 10 a ser impressa.
pszLinhaTexto10	PCHAR	56 RT	Linha 11 a ser impressa.
pszLinhaTexto11	PCHAR	56 RT	Linha 12 a ser impressa.
pszLinhaTexto12	PCHAR	56 RT	Linha 13 a ser impressa.
pszLinhaTexto13	PCHAR	56 RT	Linha 14 a ser impressa.
pszLinhaTexto14	PCHAR	56 RT	Linha 15 a ser impressa.

**Tags de Formatação:**

É possível formatar a linha e palavras a serem impressas para utilizar fontes em negrito, sublinhado, largura dupla, altura dupla, e cor invertida utilizando os seguintes tags de formatação antecedendo as palavras a serem impressas.

<AT> - Ativa todos os atributos de formatação  
<AN> - Ativa negrito  
<AS> - Ativa sublinhado  
<AL> - Ativa largura dupla  
<AA> - Ativa altura Dupla  
<AI> - Ativa cor invertida  
<DT> - Desativa todos  
<DN> - Desativa negrito  
<DS> - Desativa sublinhado

<DL> - Desativa largura dupla  
<DA> - Desativa altura dupla  
<DI> - Desativa cor invertida

Por padrão, a impressora já permite utilizar os tags para formatar uma linha texto inteira.

Para que seja possível utilizar formatação somente por palavra separadamente, é necessário habilitar a função de atributo nos jumpers de configuração da impressora. Para isto, basta entrar em modo de Intervenção Técnica, desligar a impressora e então alterar seus estados nas microchaves de seleção (Dip-Switch).

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

---

**Requisitos:**

Um documento não fiscal aberto e o ECF com papel.

---

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

```
Retorno = EPSON_NaoFiscal_Imprimir_15Linhas ( "<AN>Linha 1 em Negrito",  
"Linha 2"," Linha 3","Linha 4"," Linha 5"," <AN>Linha 6 em Negrito","<AS>  
Linha 7 sublinhada"," Linha 8","Linha 9","Linha 10","Linha 11","Linha  
12"," Linha 13","Linha 14","Linha 15" );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Imprimir_15Linhas ( "Linha 1", "Linha 2","  
Linha 3","Linha 4"," Linha 5"," <AN>Linha 6 em Negrito","<AS> Linha 7  
sublinhada"," Linha 8","Linha 9","Linha 10","Linha 11","Linha 12"," Linha  
13","Linha 14","Linha 15" )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Imprimir_15Linhas ( 'Linha 1', 'Linha 2',  
'Linha 3' , 'Linha 4' , 'Linha 5', '<AN>Linha 6 em Negrito' , '<AS> Linha  
7 sublinhada' , 'Linha 8' , 'Linha 9' , 'Linha 10' , 'Linha 11' , 'Linha  
12' , 'Linha 13' , 'Linha 14' , 'Linha 15' );
```

**4.2.66 EPSON\_NaoFiscal\_Fechar\_CCD**

Esta função fecha o Comprovante de Crédito ou Débito.

**Sintaxe:**

```
function EPSON_NaoFiscal_Fechar_CCD(bStatusCortarPapel:Boolean):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bStatusCortarPapel	BOOLEAN	-	<b>True</b> - Cortar o papel após a impressão do rodapé. <b>False</b> - Não cortar o papel após a impressão do rodapé.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

CCD aberto e o ECF com papel.

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

```
Retorno = EPSON_NaoFiscal_Fechar_CCD ( true );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Fechar_CCD ( true )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Fechar_CCD ( true );
```



**4.2.67 EPSON\_NaoFiscal\_Fechar\_Relatorio\_Gerencial**

Esta função fecha o Relatório Gerencial.

**Sintaxe:**

```
function EPSON_NaoFiscal_Fechar_Relatorio_Gerencial(bStatusCortarPapel:Boolean):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bStatusCortarPapel	BOOLEAN	-	<b>True</b> - Cortar o papel após a impressão do rodapé. <b>False</b> - Não cortar o papel após a impressão do rodapé.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Relatório gerencial aberto e o ECF com papel.

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

```
Retorno = EPSON_NaoFiscal_Fechar_Relatorio_Gerencial ( true );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Fechar_Relatorio_Gerencial ( true )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Fechar_Relatorio_Gerencial ( true );
```

**4.2.68 EPSON\_NaoFiscal\_Cancelar\_CCD**

Esta função cancela um Comprovante de Crédito ou Débito.

**Sintaxe:**

```
function EPSON_NaoFiscal_Cancelar_CCD( pszNumeroDescricaoPagamento:PChar;  
                                         pszValorPagamento:PChar; dwCasasDecimaisPagamento:Integer;  
                                         pszParcelas:PChar; pszNumeroCooCCD:PChar): Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroDescricaoPagamento	PCHAR	2 (O,N) / 15 (A)	Número da forma de pagamento ou Descritivo da forma de pagamento.
pszValorPagamento	PCHAR	13 (O,N)	Valor do pagamento.
dwCasasDecimaisPagamento	INTEGER	-	Número de casas decimais usada.
pszParcelas	PCHAR	3 (O,N)	Número de parcelas.
pszNumeroCooCCD	PCHAR	6 (O,N)	Número do COO do Comprovante de Crédito ou Débito a ser cancelado.

Obs.: Caso o parâmetro pszNumeroCupom seja preenchido, os outros parâmetros não precisam ser informados. Caso este parâmetro não seja preenchido, todos os outros precisarão ser informados.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Período de vendas aberto, ECF com papel, o CCD a ser cancelado deve ser o atual ou último documento impresso.

---

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

```
Retorno = EPSON_NaoFiscal_Cancelar_CCD ( "", "", 0, "", "234" );  
Retorno = EPSON_NaoFiscal_Cancelar_CCD ( "1", "45000", 3, "1", "" );  
Retorno = EPSON_NaoFiscal_Cancelar_CCD ( "Cartao", "45000", 3, "1", "" );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Cancelar_CCD ( "", "", 0, "", "234" )  
Retorno = EPSON_NaoFiscal_Cancelar_CCD ( "1", "45000", 3, "1", "" )  
Retorno = EPSON_NaoFiscal_Cancelar_CCD ( "Cartao", "45000", 3, "1", "" )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Cancelar_CCD ( '', '', 0, '', '234' );  
Retorno := EPSON_NaoFiscal_Cancelar_CCD ( '1', '45000', 3, '1', '' );  
Retorno := EPSON_NaoFiscal_Cancelar_CCD ( 'Cartao', '45000', 3, '1', '' );
```

**4.2.69 EPSON\_NaoFiscal\_Cancelar\_Pagamento**

Esta função efetua a Troca de uma Forma de Pagamento por outra.

**Sintaxe:**

```
function EPSON_NaoFiscal_Cancelar_Pagamento(pszNumeroDescricaoCancelar:PChar;  
                                             pszNumeroDescricaoEfetivar:PChar;  
                                             pszValorPagamento:PChar;  
                                             dwCasasDecimaisPagamento:Integer):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroDescricaoCancelar	PCHAR	2 (N) / 15 (A)	Número ou descritivo da forma de pagamento a ser decrescida.
pszNumeroDescricaoEfetivar	PCHAR	2 (N) / 15 (A)	Número ou descritivo da forma de pagamento a ser acrescida.
pszValorPagamento	PCHAR	13 (N)	Valor do pagamento.
dwCasasDecimaisPagamento	INTEGER	-	Número de casas decimais usada.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.

O último documento emitido deve ser um Cupom Fiscal ou Comprovante Não Fiscal (exceto Comprovante de Cancelamento), com o respectivo meio de pagamento a ser cancelado.

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

```
Retorno = EPSON_NaoFiscal_Cancelar_Pagamento ( "1", "2", "37500", 3 );  
Retorno = EPSON_NaoFiscal_Cancelar_Pagamento ( "Cartao", "2", "37500", 3  
);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Cancelar_Pagamento ( "1", "2", "37500", 3 )  
Retorno = EPSON_NaoFiscal_Cancelar_Pagamento ( "Cartao", "2", "37500", 3 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Cancelar_Pagamento ( '1', '2', '37500', 3 );  
Retorno := EPSON_NaoFiscal_Cancelar_Pagamento ( 'Cartao', '2', '37500', 3  
);
```

**4.2.70 EPSON\_NaoFiscal\_Nova\_Parcela\_CCD**

Esta função imprime o Comprovante de Crédito ou Débito de uma nova parcela.

**Sintaxe:**

```
function EPSON_NaoFiscal_Nova_Parcela_CCD():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel. O número de parcelas informadas no CCD deverá ser maior que 1.

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

```
Retorno = EPSON_NaoFiscal_Nova_Parcela_CCD ( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Nova_Parcela_CCD ( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Nova_Parcela_CCD ( );
```

**4.2.71 EPSON\_NaoFiscal\_Nova\_Via\_CCD**

Esta função imprime a 2º via do Comprovante de Crédito ou Débito.

**Sintaxe:**

```
function EPSON_NaoFiscal_Nova_Via_CCD():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel. A primeira via do CCD deve ser o último documento impresso.

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

```
Retorno = EPSON_NaoFiscal_Nova_Via_CCD ( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Nova_Via_CCD ( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Nova_Via_CCD ( );
```

**4.2.72 EPSON\_NaoFiscal\_Reimprimir\_CCD**

Esta função reimprime o Cupom de Crédito ou Débito.

**Sintaxe:**

```
function EPSON_NaoFiscal_Reimprimir_CCD():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel. O CCD a ser reimpresso deve ser o último documento impresso.

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

```
Retorno = EPSON_NaoFiscal_Reimprimir_CCD ( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Reimprimir_CCD ( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Reimprimir_CCD ( );
```

**4.2.73 EPSON\_NaoFiscal\_Sangria**

Esta função é usada para retirar uma quantia de dinheiro (Sangria).

**Sintaxe:**

```
function EPSON_NaoFiscal_Sangria(pszValor:PChar; dwCasasDecimaisValor:Integer):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszValor	PCHAR	11 (N)	Valor a ser retirado.
dwCasasDecimaisValor	INTEGER	-	Número de casas decimais usada.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.

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

```
Retorno = EPSON_NaoFiscal_Sangria ( "35000", 2 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Sangria ( "35000", 2 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Sangria ( '35000', 2 );
```



**4.2.74 EPSON\_NaoFiscal\_Fundo\_Troco**

Esta função é usada para inserir uma quantia de dinheiro (Fundo de Troco).

**Sintaxe:**

```
function EPSON_NaoFiscal_Fundo_Troco(pszValor:PChar; dwCasasDecimaisValor:Integer):Integer; StdCall;  
    External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszValor	PCHAR	11 (N)	Valor a ser inserido.
dwCasasDecimaisValor	INTEGER	-	Número de casas decimais usada.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.

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

```
Retorno = EPSON_NaoFiscal_Fundo_Troco ( "35000", 2 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Fundo_Troco ( "35000", 2 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Fundo_Troco ( '35000', 2 );
```

**4.2.75 EPSON\_NaoFiscal\_Imprimir\_Codigo\_Barras**

Esta função imprime código de barras durante o texto livre de relatórios gerenciais e comprovantes de crédito/debito.

Obs: Para efetuar a impressão de código de barras no campo de linhas promocionais de um comprovante não fiscal, deve-se utilizar o comando EPSON\_Fiscal\_Configurar\_Codigo\_Barras\_Mensagem.

**Sintaxe:**

```
function EPSON_NaoFiscal_Imprimir_Codigo_Barras( dwTipo:Integer; dwAltura:Integer; dwLargura:Integer;  
dwPosicaoImpressao:Integer; dwTipoCaracter:Integer;  
pszCodigo:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwTipo	INTEGER	-	Tipo de código de barras: <b><i>Veja tabela na próxima página.</i></b>
dwAltura	INTEGER	-	Altura de impressão do código de barras, entre 1 e 255.
dwLargura	INTEGER	-	Largura do código de barras, entre 2 e 6.
dwPosicaoImpressão	INTEGER	-	Posição de impressão do texto do código de barras: <b>0</b> - sem impressão de texto. <b>1</b> - impressão acima do código. <b>2</b> - impressão abaixo do código. <b>3</b> - impressão acima e abaixo do código.
dwTipoCaracter	INTEGER	-	Tipo de caractere usado no texto do código: <b>0</b> – letra tipo A (12x24) <b>1</b> – letra tipo B (9x17)
pszCodigo	PCHAR	Ver tabela	Texto contendo os dados do código de barras a ser impresso.

Valor (dwTipo)	Padrão	Quantidade de caracteres	Tipo de dados
0	UPC-A	11 à 12	0 – 9 (apenas números)
1	UPC-E	11 à 12	0 – 9 (apenas números)
2	JAN13 (EAN13)	12 à 13	0 - 9 (apenas números)
3	JAN8 (EAN8)	7 à 8	0 - 9 (apenas números)
4	CODE39	Maior ou igual a 1	0 à 9, A à Z e os caracteres: [espaço] \$ % * + - . , /
5	ITF	Maior ou igual a 1 e somente quantidade par	0 à 9
6	CODABAR (NW7)	Maior ou igual a 1	0 à 9, A à D e os caracteres: \$ + - . / :
65	UPC-A	11 à 12	0 à 9
66	UPC-E	11 à 12	0 à 9
67	JAN13 (EAN13)	12 à 13	0 à 9
68	JAN8 (EAN8)	7 à 8	0 à 9
69	CODE39	1 à 255	0 à 9, A à Z e os caracteres: [espaço] \$ % * + - . , /
70	ITF	2 à 255 (par)	0 à 9
71	CODABAR (NW7)	1 à 255	0 à 9, A à D e os caracteres: \$ + - . / :
72	CODE93	1 à 255	NULL à [CTRL]
73	CODE128	2 à 255	NULL à [CTRL]

## Observação:

**UPC-E:** Deve ser preenchido com 0, obrigatoriamente seu primeiro caractere.

**CODE128:** Os dois primeiros bytes dos dados, selecionam as tabelas abaixo A, B e C, conforme os códigos:

{ **A**(0x7B 0x41) = Seleciona tabela A

{ **B**(0x7B 0x42) = Seleciona tabela B

{ **C**(0x7B 0x43) = Seleciona tabela C

d		Character		
Hex	Decimal	CODE A	CODE B	CODE C
00	0	NUL		00
01	1	SOH		01
02	2	STX		02
03	3	ETX		03
04	4	EDT		04
05	5	ENQ		05
06	6	ACK		06
07	7	BEL		07
08	8	BS		08
09	9	HT		09
0A	10	LF		10
0B	11	VT		11
0C	12	FF		12
0D	13	CR		13
0E	14	SO		14
0F	15	SI		15
10	16	DLE		16
11	17	DC1		17
12	18	DC2		18
13	19	DC3		19
14	20	DC4		20
15	21	NAK		21
16	22	SYN		22
17	23	ETB		23
18	24	CAN		24
19	25	EM		25
1A	26	SUB		26
1B	27	ESC		27
1C	28	FS		28
1D	29	GS		29
1E	30	RS		30
1F	31	US		31
20	32	SP	SP	32
21	33	!	!	33
22	34	"	"	34
23	35	#	#	35
24	36	\$	\$	36

d		Character		
Hex	Decimal	CODE A	CODE B	CODE C
25	37	%	%	37
26	38	&	&	38
27	39	'	'	39
28	40	(	(	40
29	41	)	)	41
2A	42	*	*	42
2B	43	+	+	43
2C	44	,	,	44
2D	45	-	-	45
2E	46	.	.	46
2F	47	/	/	47
30	48	0	0	48
31	49	1	1	49
32	50	2	2	50
33	51	3	3	51
34	52	4	4	52
35	53	5	5	53
36	54	6	6	54
37	55	7	7	55
38	56	8	8	56
39	57	9	9	57
3A	58	:	:	58
3B	59	;	;	59
3C	60	<	<	60
3D	61	=	=	61
3E	62	>	>	62
3F	63	?	?	63
40	64	@	@	64
41	65	A	A	65
42	66	B	B	66
43	67	C	C	67
44	68	D	D	68
45	69	E	E	69
46	70	F	F	70
47	71	G	G	71
48	72	H	H	72
49	73	I	I	73

d		Character		
Hex	Decimal	CODE A	CODE B	CODE C
4A	74	J	J	74
4B	75	K	K	75
4C	76	L	L	76
4D	77	M	M	77
4E	78	N	N	78
4F	79	O	O	79
50	80	P	P	80
51	81	Q	Q	81
52	82	R	R	82
53	83	S	S	83
54	84	T	T	84
55	85	U	U	85
56	86	V	V	86
57	87	W	W	87
58	88	X	X	88
59	89	Y	Y	89
5A	90	Z	Z	90
5B	91	[	[	91
5C	92	\	\	92
5D	93	]	]	93
5E	94	^	^	94
5F	95	_	_	95
60	96		'	96
61	97		a	97
62	98		b	98
63	99		c	99
64	100		d	
65	101		e	
66	102		f	
67	103		g	
68	104		h	
69	105		i	
6A	106		j	
6B	107		k	
6C	108		l	
6D	109		m	
6E	110		n	

d		Character		
Hex	Decimal	CODE A	CODE B	CODE C
6F	111		o	
70	112		p	
71	113		q	
72	114		r	
73	115		s	
74	116		t	
75	117		u	
76	118		v	
77	119		w	
78	120		x	
79	121		y	
7A	122		z	
7B, 7B	123, 123		{	
7C	124			
7D	125		}	
7E	126		~	
7F	127		DEL	

d		Special Character		
Hex	Decimal	CODE A	CODE B	CODE C
7B, 31	123, 49	FNC1	FNC1	FNC1
7B, 32	123, 50	FNC2	FNC2	
7B, 33	123, 51	FNC3	FNC3	
7B, 34	123, 52	FNC4	FNC4	
7B, 41	123, 65		CODE A	CODE A
7B, 42	123, 66	CODE B		CODE B
7B, 43	123, 67	CODE C	CODE C	
7B, 53	123, 83	SHIFT	SHIFT	

### Saídas:

Nenhum.

### Retornos:

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

---

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, deve haver um Relatório Gerencial ou um documento CCD aberto e o ECF deve estar com papel.

Para a correta impressão do código de barras, especialmente nos padrões CODE93 e CODE128, a quantidade de dados a ser inserida no mesmo deve ser compatível com a área disponível no cupom. Caso um número muito grande de caracteres seja informado, ou caso o parâmetro dwLargura informe uma largura muito grande nas barras, o código pode não caber no documento e neste caso não será impresso.

---

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

```
Retorno = EPSON_NaoFiscal_Imprimir_Codigo_Barras ( 2, 100, 4, 2, 0,  
"1234567890123" );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_NaoFiscal_Imprimir_Codigo_Barras ( 2, 100, 4, 2, 0,  
"1234567890123" )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_NaoFiscal_Imprimir_Codigo_Barras ( 2, 100, 4, 2, 0,  
'1234567890123' );
```

---

**Exemplos de impressão do código de barras no padrão CODE 128:**

```
// Code 128-A
```

```
Retorno := EPSON_NaoFiscal_Imprimir_Codigo_Barras(73, 60, 2, 2, 0,  
'{ADADOS012345}');
```

```
// Code 128-B
```

```
Retorno := EPSON_NaoFiscal_Imprimir_Codigo_Barras(73, 60, 2, 2, 0,  
'{Bdados012345}');
```

```
// Code 128-C
```

```
Retorno := EPSON_NaoFiscal_Imprimir_Codigo_Barras(73, 60, 2, 2, 0,  
'{C00000012345}');
```

**4.2.76 EPSON\_NaoFiscal\_Obter\_SubTotal**

Esta função retorna o subtotal do cupom.

**Sintaxe:**

```
function EPSON_NaoFiscal_Obter_SubTotal(pszSubTotalCupom:PChar):Integer;StdCall;External  
    'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszSubTotalCupom	PCHAR	13 (N)	Subtotal do cupom.

**Retornos:**


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

**Requisitos:**


Para obter subtotal do comprovante corrente: a Jornada fiscal deve estar aberta, em período de vendas, comprovante não fiscal aberto e o ECF com papel.

Para obter subtotal do último comprovante (somente versões de ECF FBIII): a Jornada fiscal deve estar aberta, em período de vendas, documentos fechados, ECF com papel e o último documento deve ser um comprovante não fiscal.


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

```
 char szSubtotalCupom[20];  
Retorno = EPSON_NaoFiscal_Obter_SubTotal ( szSubtotalCupom );
```


**Exemplo em C# :**

```
 StringBuilder szSubtotalCupom = new StringBuilder(20,20);  
Retorno = EPSON_NaoFiscal_Obter_SubTotal ( szSubtotalCupom );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
 Dim szSubTotalCupom As String  
szSubTotalCupom = Space(20)  
Retorno = EPSON_NaoFiscal_Obter_SubTotal ( szSubTotalCupom )
```

**Exemplo em Delphi 7:**

```
 szSubTotalCupom: array[0..20] of Char;  
Retorno := EPSON_NaoFiscal_Obter_SubTotal ( szSubTotalCupom );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szSubTotalCupom := AnsiString(StringOfChar(' ', 20));  
Retorno := EPSON_NaoFiscal_Obter_SubTotal(PAnsiChar(szSubTotalCupom));
```

**4.2.77 EPSON\_ESC\_ECF\_NaoFiscal\_Abrir\_CCD**

Esta função abre um Comprovante de Crédito ou Débito (TEF) , seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Abrir_CCD(szIndice:PChar; szCodigo:PChar; szParcelas:PChar;  
szNumeroParcela:PChar; szCPFCNPJ:PChar;  
szNome:PChar; szEndereco:PChar; szCOO:PChar;  
szDataHora:PChar;szVendaBruta:PChar;  
szNumeroSerie:PChar; szSequencia:PChar;  
szNumeroParcelaImpressa:PChar;  
szNumeroParcelaFaltante:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (O,N)	Seqüência do pagamento no Cupom Fiscal ou Comprovante Não Fiscal.
szCodigo	PCHAR	2 (N)	Código do meio de pagamento cadastrado no ECF.
szParcelas	PCHAR	2 (N)	Quantidade total de parcelas (1 para pagamento a vista).
szNumeroParcela	PCHAR	2 (O,N)	Número desta parcela (1 para pagamento a vista).
szCPFCNPJ	PCHAR	14 (O,N)	CPF ou CNPJ do consumidor.
szNome	PCHAR	30 (O,A)	Nome do consumidor.
szEndereco	PCHAR	79 (O,A)	Endereço do consumidor.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do comprovante.
szDataHora	PCHAR	16 (A)	Data e Hora de abertura do comprovante.
szVendaBruta	PCHAR	15 (N)	Venda Bruta do ECF.
szNumeroSerie	PCHAR	21 (A)	Número de Série do ECF.
szSequencia	PCHAR	3 (N)	Seqüência do pagamento.
szNumeroParcelaImpressa	PCHAR	3 (N)	Número da parcela impressa no comprovante.
szNumeroParcelaFaltante	PCHAR	3 (N)	Número de parcelas faltantes (não impressas).

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados, o ECF com papel, o valor e o número da forma de pagamento deve coincidir com o valor e a forma de pagamento no último cupom. Comando disponível apenas para ECF modelo TM-T800F.



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

```
char szCOO[10], szDataHora[16], szVendaBruta[15], szNumeroSerie[21],  
szSequencia[3], szNumeroParcelaImpressa[3], szNumeroParcelaFaltante[3];  
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_CCD( "1", "2", "1", "1", "", "",  
"", szCOO, szDataHora, szVendaBruta, szNumeroSerie, szSequencia,  
szNumeroParcelaImpressa, szNumeroParcelaFaltante);
```

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);  
StringBuilder szDataHora = new StringBuilder(16, 16);  
StringBuilder szVendaBruta = new StringBuilder(15, 15);  
StringBuilder szNumeroSerie = new StringBuilder(21, 21);  
StringBuilder szSequencia = new StringBuilder(3, 3);  
StringBuilder szNumeroParcelaImpressa = new StringBuilder(3, 3);  
StringBuilder szNumeroParcelaFaltante = new StringBuilder(3, 3);  
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_CCD( "1", "2", "1", "1", "", "",  
"", szCOO, szDataHora, szVendaBruta, szNumeroSerie, szSequencia,  
szNumeroParcelaImpressa, szNumeroParcelaFaltante);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String  
Dim szDataHora As String  
Dim szVendaBruta As String  
Dim szNumeroSerie As String  
Dim szSequencia As String  
Dim szNumeroParcelaImpressa As String  
Dim szNumeroParcelaFaltante As String  
szCOO = Space(10)  
szDataHora = Space(16)  
szVendaBruta = Space(15)  
szNumeroSerie = Space(21)  
szSequencia = Space(3)  
szNumeroParcelaImpressa = Space(3)  
szNumeroParcelaFaltante = Space(3)  
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_CCD( "1", "2", "1", "1", "", "",  
"", szCOO, szDataHora, szVendaBruta, szNumeroSerie, szSequencia,  
szNumeroParcelaImpressa, szNumeroParcelaFaltante)
```

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;  
szDataHora: array[0..16] of Char;  
szVendaBruta: array[0..15] of Char;  
szNumeroSerie: array[0..21] of Char;  
szSequencia: array[0..3] of Char;  
szNumeroParcelaImpressa: array[0..3] of Char;  
szNumeroParcelaFaltante: array[0..3] of Char;  
Retorno := EPSON_ESC_ECF_NaoFiscal_Abrir_CCD( '1', '2', '1', '1', '', '',  
'', szCOO, szDataHora, szVendaBruta, szNumeroSerie, szSequencia,  
szNumeroParcelaImpressa, szNumeroParcelaFaltante);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));  
szDataHora := AnsiString(StringOfChar(' ', 16));  
szVendaBruta := AnsiString(StringOfChar(' ', 15));  
szNumeroSerie := AnsiString(StringOfChar(' ', 21));  
szSequencia := AnsiString(StringOfChar(' ', 3));  
szNumeroParcelaImpressa := AnsiString(StringOfChar(' ', 3));  
szNumeroParcelaFaltante := AnsiString(StringOfChar(' ', 3));  
Retorno := EPSON_ESC_ECF_NaoFiscal_Abrir_CCD( '1', '2', '1', '1', '', '',  
'', PAnsiChar(szCOO), PAnsiChar(szDataHora), PAnsiChar(szVendaBruta),  
PAnsiChar(szNumeroSerie), PAnsiChar(szSequencia),  
PAnsiChar(szNumeroParcelaImpressa), PAnsiChar(szNumeroParcelaFaltante));
```

**4.2.78 EPSON\_ESC\_ECF\_NaoFiscal\_Abrir\_CCDEX**

Esta função abre um Comprovante de Crédito ou Débito (TEF) , seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Abrir_CCDEX(szIndice:PChar; szCodigo:PChar; szParcelas:PChar;  
szNumeroParcela:PChar; szCPF CNPJ:PChar;  
szNome:PChar; szEndereco:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (O,N)	Seqüência do pagamento no Cupom Fiscal ou Comprovante Não Fiscal.
szCodigo	PCHAR	2 (N)	Código do meio de pagamento cadastrado no ECF.
szParcelas	PCHAR	2 (N)	Quantidade total de parcelas (1 para pagamento a vista).
szNumeroParcela	PCHAR	2 (O,N)	Número desta parcela (1 para pagamento a vista).
szCPF CNPJ	PCHAR	14 (O,N)	CPF ou CNPJ do consumidor.
szNome	PCHAR	30 (O,A)	Nome do consumidor.
szEndereco	PCHAR	79 (O,A)	Endereço do consumidor.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados, o ECF com papel, o valor e o número da forma de pagamento deve coincidir com o valor e a forma de pagamento no último cupom. Comando disponível apenas para ECF modelo TM-T800F.

---

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_CCDEX( "1","2","1","1","","","" );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_CCDEX( "1","2","1","1","","","" )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Abrir_CCDEX('1','2','1','1','','','');
```

**4.2.79 EPSON\_ESC\_ECF\_NaoFiscal\_Imprimir\_Linha**

Esta função imprime uma linha texto no Relatório Gerencial aberto ou no Comprovante de Crédito ou Débito aberto, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Imprimir_Linha(szTexto:PChar):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szTexto	PCHAR	1024 (A)	Linha de texto livre.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Um documento não fiscal aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Imprimir_Linha( "Linha de texto" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Imprimir_Linha( "Linha de texto" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Imprimir_Linha( 'Linha de texto' );
```

**4.2.80 EPSON\_ESC\_ECF\_NaoFiscal\_Fechar\_CCD\_RG**

Esta função fecha o Relatório Gerencial ou Comprovante de Crédito ou Débito, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RG(szGuilhotina:PChar; szCOO:PChar;  
szDataHora:PChar; szVendaBruta:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szGuilhotina	PCHAR	1 (N)	Acionamento da guilhotina no final da impressão. 0 - Não aciona. 1 - Aciona

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do comprovante que foi encerrado.
szDataHora	PCHAR	16 (A)	Data e Hora de fechamento do comprovante.
szVendaBruta	PCHAR	15 (N)	Venda Bruta.

**Retornos:**

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

**Requisitos:**

CCD ou RG aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szCOO[10], szDataHora[16], szVendaBruta[15];  
Retorno = EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RG("1", szCOO, szDataHora,  
szVendaBruta);
```

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);  
StringBuilder szDataHora = new StringBuilder(16, 16);  
StringBuilder szVendaBruta = new StringBuilder(15, 15);  
Retorno = EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RG("1", szCOO, szDataHora,  
szVendaBruta);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String
Dim szDataHora As String
Dim szVendaBruta As String
szCOO = Space(10)
szDataHora = Space(16)
szVendaBruta = Space(15)
Retorno = EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RG("1", szCOO, szDataHora,
szVendaBruta)
```

---

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;
szDataHora: array[0..16] of Char;
szVendaBruta: array[0..15] of Char;
Retorno := EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RG('1', szCOO, szDataHora,
szVendaBruta);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));
szDataHora := AnsiString(StringOfChar(' ', 16));
szVendaBruta := AnsiString(StringOfChar(' ', 15));
Retorno := EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RG('1', PAnsiChar(szCOO),
PAnsiChar(szDataHora), PAnsiChar(szVendaBruta));
```

**4.2.81 EPSON\_ESC\_ECF\_NaoFiscal\_Fechar\_CCD\_RGEX**

Esta função fecha o Relatório Gerencial ou Comprovante de Crédito ou Débito, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RGEX(szGuilhotina:PChar):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szGuilhotina	PCHAR	1 (N)	Acionamento da guilhotina no final da impressão. 0 - Não aciona. 1 - Aciona

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

CCD ou RG aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

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

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Fechar_CCD_RGEX("1")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

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



**4.2.82 EPSON\_ESC\_ECF\_NaoFiscal\_Abrir\_Relatorio\_Gerencial**

Esta função abre um Relatório Gerencial, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_Gerencial(szIndice:PChar;szCOO:PChar;  
szDataHora:PChar; szVendaBruta:PChar; szNumeroSerie:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice do relatório que será aberto, cadastrado no ECF.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do comprovante que foi iniciado.
szDataHora	PCHAR	16 (A)	Data e Hora de início do comprovante.
szVendaBruta	PCHAR	15 (N)	Venda Bruta.
szNumeroSerie	PCHAR	21 (A)	Número de Série do ECF.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szCOO[10], szDataHora[16], szVendaBruta[15], szNumeroSerie[21];  
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_Gerencial("1", szCOO,  
szDataHora, szVendaBruta, szNumeroSerie);
```

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);  
StringBuilder szDataHora = new StringBuilder(16, 16);  
StringBuilder szVendaBruta = new StringBuilder(15, 15);  
StringBuilder szNumeroSerie = new StringBuilder(21, 21);  
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_Gerencial("1", szCOO,  
szDataHora, szVendaBruta, szNumeroSerie);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String
Dim szDataHora As String
Dim szVendaBruta As String
Dim szNumeroSerie As String
szCOO = Space(10)
szDataHora = Space(16)
szVendaBruta = Space(15)
szNumeroSerie = Space(21)
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_Gerencial("1", szCOO,
szDataHora, szVendaBruta, szNumeroSerie)
```

---

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;
szDataHora: array[0..16] of Char;
szVendaBruta: array[0..15] of Char;
szNumeroSerie : array[0..21] of Char;
Retorno := EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_Gerencial('1', szCOO,
szDataHora, szVendaBruta, szNumeroSerie);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));
szDataHora := AnsiString(StringOfChar(' ', 16));
szVendaBruta := AnsiString(StringOfChar(' ', 15));
szNumeroSerie := AnsiString(StringOfChar(' ', 21));
Retorno := EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_Gerencial('1',
PAnsiChar(szCOO), PAnsiChar(szDataHora), PAnsiChar(szVendaBruta),
PAnsiChar(szNumeroSerie));
```

**4.2.83 EPSON\_ESC\_ECF\_NaoFiscal\_Abrir\_Relatorio\_GerencialEX**

Esta função abre um Relatório Gerencial, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_GerencialEX(szIndice:PChar):Integer; StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice do relatório que será aberto, cadastrado no ECF.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

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

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_Relatorio_GerencialEX( "1" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

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

**4.2.84 EPSON\_ESC\_ECF\_NaoFiscal\_Cancelar\_CCD**

Esta função cancela um Comprovante de Crédito ou Débito, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Cancelar_CCD(szCOO:PChar; szCPFCNPJ:PChar; szNome:PChar;  
szEndereco:PChar; szCOORet:PChar; szDataHora:PChar;  
szVendaBruta:PChar; szNumeroSerie:PChar; szIndice:PChar;  
szParcela:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	9 (N)	COO do CCD que será cancelado.
szCPFCNPJ	PCHAR	14 (O,N)	CPF ou CNPJ do consumidor.
szNome	PCHAR	30 (O,A)	Nome do consumidor.
szEndereco	PCHAR	79 (O,A)	Endereço do consumidor.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do comprovante.
szDataHora	PCHAR	16 (A)	Data e Hora de abertura do comprovante.
szVendaBruta	PCHAR	15 (N)	Venda Bruta do ECF.
szNumeroSerie	PCHAR	21 (A)	Número de Série do ECF.
szIndice	PCHAR	3 (N)	Sequência do pagamento.
szParcela	PCHAR	3 (N)	Número da parcela impressa no comprovante.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szCOO[10], szDataHora[16], szVendaBruta[15], szNumeroSerie[21],  
szIndice[3], szParcela[3];  
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_CCD( "123", "", "", "", szCOO,  
szDataHora, szVendaBruta, szNumeroSerie, szIndice, szParcela);
```

---

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);
StringBuilder szDataHora = new StringBuilder(16, 16);
StringBuilder szVendaBruta = new StringBuilder(15, 15);
StringBuilder szNumeroSerie = new StringBuilder(21, 21);
StringBuilder szIndice = new StringBuilder(3, 3);
StringBuilder szParcela = new StringBuilder(3, 3);
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_CCD( "123", "", "", "", szCOO,
szDataHora, szVendaBruta, szNumeroSerie, szIndice, szParcela);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String
Dim szDataHora As String
Dim szVendaBruta As String
Dim szNumeroSerie As String
Dim szIndice As String
Dim szParcela As String
szCOO = Space(10)
szDataHora = Space(16)
szVendaBruta = Space(15)
szNumeroSerie = Space(21)
szIndice = Space(3)
szParcela = Space(3)
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_CCD( "123", "", "", "", szCOO,
szDataHora, szVendaBruta, szNumeroSerie, szIndice, szParcela )
```

---

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;
szDataHora: array[0..16] of Char;
szVendaBruta: array[0..15] of Char;
szNumeroSerie: array[0..21] of Char;
szIndice: array[0..3] of Char;
szParcela: array[0..3] of Char;
Retorno := EPSON_ESC_ECF_NaoFiscal_Cancelar_CCD( '123', '', '', '', szCOO,
szDataHora, szVendaBruta, szNumeroSerie, szIndice, szParcela );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));
szDataHora := AnsiString(StringOfChar(' ', 16));
szVendaBruta := AnsiString(StringOfChar(' ', 15));
szNumeroSerie := AnsiString(StringOfChar(' ', 21));
szIndice := AnsiString(StringOfChar(' ', 3));
szParcela := AnsiString(StringOfChar(' ', 3));
Retorno := EPSON_ESC_ECF_NaoFiscal_Cancelar_CCD( '123', '', '', '',
PAnsiChar(szCOO), PAnsiChar(szDataHora), PAnsiChar(szVendaBruta),
PAnsiChar(szNumeroSerie), PAnsiChar(szIndice), PAnsiChar(szParcela));
```

**4.2.85 EPSON\_ESC\_ECF\_NaoFiscal\_Cancelar\_CCDEX**

Esta função cancela um Comprovante de Crédito ou Débito, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Cancelar_CCDEX(szCOO:PChar; szCPFCNPJ:PChar; szNome:PChar;  
szEndereco:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	9 (N)	COO do CCD que será cancelado.
szCPFCNPJ	PCHAR	14 (O,N)	CPF ou CNPJ do consumidor.
szNome	PCHAR	30 (O,A)	Nome do consumidor.
szEndereco	PCHAR	79 (O,A)	Endereço do consumidor.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_CCDEX( "123", "", "", "" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_CCDEX( "123", "", "", "" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Cancelar_CCDEX( '123', '', '', '' );
```

**4.2.86 EPSON\_ESC\_ECF\_NaoFiscal\_Nova\_Via\_CCD**

Esta função imprime a 2º via do Comprovante de Crédito ou Débito, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Nova_Via_CCD():Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Nova_Via_CCD();
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Nova_Via_CCD()
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Nova_Via_CCD();
```

**4.2.87 EPSON\_ESC\_ECF\_NaoFiscal\_Reimprimir\_CCD**

Esta função reimprime o Cupom de Crédito ou Débito, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Reimprimir_CCD():Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Reimprimir_CCD();
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Reimprimir_CCD()
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Reimprimir_CCD();
```



**4.2.88 EPSON\_ESC\_ECF\_NaoFiscal\_Abrir\_Comprovante**

Abre um Comprovante Não-Fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Abrir_Comprovante(szCPFCNPJ:PChar; szNome:PChar;  
                                                    szEndereco:PChar; szCOO:PChar; szDataHora:PChar;  
                                                    szVendaBruta:PChar; szNumeroSerie:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCPFCNPJ	PCHAR	14 (O,RT)	Número CPF ou CNPJ do cliente.
szRazaoSocialNomeCliente	PCHAR	30 (O, RT)	Razão Social / Nome do cliente.
szEndereco	PCHAR	79 (O, RT)	Endereço do cliente.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do comprovante aberto.
szDataHora	PCHAR	16 (N)	Data e hora de abertura do comprovante.
szVendaBruta	PCHAR	15 (N)	Valor da Venda Bruta.
szNumeroSerie	PCHAR	21 (A)	Numero de Série do ECF.

**Retornos:**

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

**Requisitos:**

O ECF deve estar ligado, com papel e com a jornada fiscal aberta ou sem redução Z pendente.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szCOO[10], szDataHora[16], szVendaBruta[15], szNumeroSerie[21];  
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_Comprovante( "12345678901230",  
"Empresa XYZ", "", szCOO, szDataHora, szVendaBruta, szNumeroSerie);
```

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);  
StringBuilder szDataHora = new StringBuilder(16, 16);  
StringBuilder szVendaBruta = new StringBuilder(15, 15);  
StringBuilder szNumeroSerie = new StringBuilder(21, 21);  
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_Comprovante( "12345678901230",  
"Empresa XYZ", "", szCOO, szDataHora, szVendaBruta, szNumeroSerie);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String
Dim szDataHora As String
Dim szVendaBruta As String
Dim szNumeroSerie As String
szCOO = Space(10)
szDataHora = Space(16)
szVendaBruta = Space(15)
szNumeroSerie = Space(21)
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_Comprovante( "12345678901230",
"Empresa XYZ", "", szCOO, szDataHora, szVendaBruta, szNumeroSerie)
```

---

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;
szDataHora: array[0..16] of Char;
szVendaBruta: array[0..15] of Char;
szNumeroSerie: array[0..21] of Char;
Retorno := EPSON_ESC_ECF_NaoFiscal_Abrir_Comprovante( '12345678901230',
'Empresa XYZ', '', szCOO, szDataHora, szVendaBruta, szNumeroSerie);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));
szDataHora := AnsiString(StringOfChar(' ', 16));
szVendaBruta := AnsiString(StringOfChar(' ', 15));
szNumeroSerie := AnsiString(StringOfChar(' ', 21));
Retorno := EPSON_ESC_ECF_NaoFiscal_Abrir_Comprovante( '12345678901230',
'Empresa XYZ', '', PAnsiChar(szCOO), PAnsiChar(szDataHora),
PAnsiChar(szVendaBruta), PAnsiChar(szNumeroSerie));
```

**4.2.89 EPSON\_ESC\_ECF\_NaoFiscal\_Abrir\_ComprovanteEX**

Abre um Comprovante Não-Fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Abrir_ComprovanteEX(szCPFCNPJ:PChar; szNome:PChar;  
szEndereco:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCPFCNPJ	PCHAR	14 (O,RT)	Número CPF ou CNPJ do cliente.
szRazaoSocialNomeCliente	PCHAR	30 (O, RT)	Razão Social / Nome do cliente.
szEndereco	PCHAR	79 (O, RT)	Endereço do cliente.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

O ECF deve estar ligado, com papel e com a jornada fiscal aberta ou sem redução Z pendente.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_ComprovanteEX( "12345678901230",  
"Empresa XYZ", "");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Abrir_ComprovanteEX( "12345678901230",  
"Empresa XYZ", "")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Abrir_ComprovanteEX( '12345678901230',  
'Empresa XYZ', '');
```

**4.2.90 EPSON\_ESC\_ECF\_NaoFiscal\_Vender\_Item**

Esta função registra no Comprovante um item não-fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Vender_Item(szIndice:PChar; szValor:PChar; szNumeroItem:PChar;  
szSubtotal:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice do totalizador não fiscal cadastrado no ECF (3 a 30).
szValor	PCHAR	13 (N)	Valor do item registrado.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szNumeroItem	PCHAR	11 (N)	Número do item registrado.
szSubtotal	PCHAR	11 (N)	Subtotal do comprovante.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel.

Para executar as operações 1 ou 2, utilize a função

[EPSON\\_ESC\\_ECF\\_NaoFiscal\\_Sangria\\_Fundo\\_Troco.](#)

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szNumeroItem[14], szSubtotal[14];  
Retorno = EPSON_ESC_ECF_NaoFiscal_Vender_Item( "3", "1000", szNumeroItem,  
szSubtotal);
```

**Exemplo em C#:**

```
StringBuilder szNumeroItem= new StringBuilder(14, 14);  
StringBuilder szSubtotal= new StringBuilder(14, 14);  
Retorno = EPSON_ESC_ECF_NaoFiscal_Vender_Item( "3", "1000", szNumeroItem,  
szSubtotal);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szNumeroItem As String
Dim szSubtotal As String
szNumeroItem = Space(14)
szSubtotal = Space(14)
Retorno = EPSON_ESC_ECF_NaoFiscal_Vender_Item( "3", "1000", szNumeroItem,
szSubtotal)
```

---

**Exemplo em Delphi 7:**

```
szNumeroItem: array[0..14] of Char;
szSubtotal: array[0..14] of Char;
Retorno := EPSON_ESC_ECF_NaoFiscal_Vender_Item( '3', '1000', szNumeroItem,
szSubtotal);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szNumeroItem := AnsiString(StringOfChar(' ', 14));
szSubtotal := AnsiString(StringOfChar(' ', 14));
Retorno := EPSON_ESC_ECF_NaoFiscal_Vender_Item( '3', '1000',
PAnsiChar(szNumeroItem), PAnsiChar( szSubtotal));
```

**4.2.91 EPSON\_ESC\_ECF\_NaoFiscal\_Vender\_ItemEX**

Esta função registra no Comprovante um item não-fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Vender_ItemEX(szIndice:PChar; szValor:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice do totalizador não fiscal cadastrado no ECF (3 a 30).
szValor	PCHAR	13 (N)	Valor do item registrado.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel.  
Para executar as operações 1 ou 2, utilize a função

[EPSON\\_ESC\\_ECF\\_NaoFiscal\\_Sangria\\_Fundo\\_Troco.](#)

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Vender_ItemEX( "3", "1000" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Vender_ItemEX( "3", "1000" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Vender_ItemEX( '3', '1000' );
```

**4.2.92 EPSON\_ESC\_ECF\_NaoFiscal\_Fechar\_Comprovante**

Esta função fecha o Comprovante Não-Fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Fechar_Comprovante(szGuilhotina:PChar; szInformacoes:PChar;  
szCOO:PChar; szDataHora:PChar;  
szVendaBruta:PChar; pszIndices:PChar;  
pszCodigos:PChar; pszValores:PChar;  
pszParcelas:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szGuilhotina	PCHAR	1 (N)	Aciona Guilhotina. <b>0</b> – Não Aciona, <b>1</b> - Aciona
szInformacoes	PCHAR	1024(O,A)	Informações adicionais do cupom.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do cupom que foi encerrado.
szDataHora	PCHAR	16 (A)	Data e Hora de fechamento do cupom.
szVendaBruta	PCHAR	15 (N)	Venda Bruta.
szIndices	PCHAR	60 (N)	Sequencial do meio de pagamento no cupom (até 20 pagamentos).
szCodigos	PCHAR	60 (N)	Número do meio de pagamento (até 20 pagamentos).
szValores	PCHAR	280 (N)	Valor recebido no meio de pagamento (até 20 pagamentos).
szParcelas	PCHAR	60 (N)	Número de parcelas utilizadas no meio de pagamento (até 20 pagamentos).

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel e deve possuir os devidos pagamentos.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szCOO[10], szDataHora[16], szVendaBruta[15], szIndices[60],  
szCodigos[60], szValores[280], szParcelas[60];  
Retorno = EPSON_ESC_ECF_NaoFiscal_Fechar_Comprovante( "1", "Obrigado e  
Volte Sempre!", szCOO, szDataHora, szVendaBruta, szIndices, szCodigos,  
szValores, szParcelas);
```

---

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);
StringBuilder szDataHora = new StringBuilder(16, 16);
StringBuilder szVendaBruta = new StringBuilder(15, 15);
StringBuilder szIndices = new StringBuilder(60, 60);
StringBuilder szCodigos = new StringBuilder(60, 60);
StringBuilder szValores = new StringBuilder(280, 280);
StringBuilder szParcelas = new StringBuilder(60, 60);
Retorno = EPSON_ESC_ECF_NaoFiscal_Fechar_Comprovante("1", "Obrigado e
Volte Sempre!", szCOO, szDataHora, szVendaBruta, szIndices, szCodigos,
szValores, szParcelas);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String
Dim szDataHora As String
Dim szVendaBruta As String
Dim szIndices As String
Dim szCodigos As String
Dim szValores As String
Dim szParcelas As String
szCOO = Space(10)
szDataHora = Space(16)
szVendaBruta = Space(15)
szIndices = Space(60)
szCodigos = Space(60)
szValores = Space(280)
szParcelas = Space(60)
Retorno = EPSON_ESC_ECF_NaoFiscal_Fechar_Comprovante("1", "Obrigado e Volte
Sempre!", szCOO, szDataHora, szVendaBruta, szIndices, szCodigos,
szValores, szParcelas)
```

---

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;
szDataHora: array[0..16] of Char;
szVendaBruta: array[0..15] of Char;
szIndices: array[0..60] of Char;
szCodigos: array[0..60] of Char;
szValores: array[0..280] of Char;
szParcelas: array[0..60] of Char;
Retorno := EPSON_ESC_ECF_NaoFiscal_Fechar_Comprovante('1', 'Obrigado e
Volte Sempre!', szCOO, szDataHora, szVendaBruta, szIndices, szCodigos,
szValores, szParcelas);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));
szDataHora := AnsiString(StringOfChar(' ', 16));
szVendaBruta := AnsiString(StringOfChar(' ', 15));
szIndices := AnsiString(StringOfChar(' ', 60));
szCodigos := AnsiString(StringOfChar(' ', 60));
szValores := AnsiString(StringOfChar(' ', 280));
szParcelas := AnsiString(StringOfChar(' ', 60));
Retorno := EPSON_ESC_ECF_NaoFiscal_Fechar_Comprovante('1', 'Obrigado e
Volte Sempre!', PAnsiChar(szCOO), PAnsiChar(szDataHora),
PAnsiChar(szVendaBruta), PAnsiChar(szIndices), PAnsiChar(szCodigos),
PAnsiChar(szValores), PAnsiChar(szParcelas));
```



**4.2.93 EPSON\_ESC\_ECF\_NaoFiscal\_Fechar\_ComprovanteEX**

Esta função fecha o Comprovante Não-Fiscal, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Fechar_ComprovanteEX(szGuilhotina:PChar;  
                                                         szInformacoes:PChar):Integer; StdCall; External  
                                                         'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szGuilhotina	PCHAR	1 (N)	Aciona Guilhotina. <b>0</b> – Não Aciona, <b>1</b> - Aciona
szInformacoes	PCHAR	1024(O,A)	Informações adicionais do cupom.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom não fiscal aberto, o ECF com papel e deve possuir os devidos pagamentos.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Fechar_ComprovanteEX( "1", "Obrigado e  
Volte Sempre!");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Fechar_ComprovanteEX("1", "Obrigado e  
Volte Sempre!")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Fechar_ComprovanteEX('1', 'Obrigado e  
Volte Sempre!');
```

**4.2.94 EPSON\_ESC\_ECF\_NaoFiscal\_Cancelar\_Pagamento**

Esta função efetua a Troca de uma Forma de Pagamento por outra, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Cancelar_Pagamento(szIndiceEstornado:PChar;  
szIndiceEfetivado:PChar; szValor:PChar;  
szNumeroParcelas:PChar; szMensagem:PChar;  
szCOO:PChar; szDataHora:PChar;  
szVendaBruta:PChar; szNumeroSerie:PChar;  
szIndices:PChar; szCodigos:PChar; szValores:PChar;  
szParcelas:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndiceEstornado	PCHAR	2 (N)	Índice do meio de pagamento cadastrado no ECF que será estornado.
szIndiceEfetivado	PCHAR	2 (N)	Índice do novo meio de pagamento cadastrado no ECF que será utilizado.
szValor	PCHAR	13 (N)	Valor do novo meio de pagamento
szNumeroParcelas	PCHAR	2 (N)	Número de parcelas do pagamento.
szMensagem	PCHAR	1024(O,A)	Mensagem que será impressa no comprovante.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do comprovante.
szDataHora	PCHAR	16 (A)	Data e Hora de fechamento do comprovante.
szVendaBruta	PCHAR	15 (N)	Venda Bruta.
szNumeroSerie	PCHAR	21 (N)	Número de série do ECF.
szIndices	PCHAR	60 (N)	Sequencial do meio de pagamento no cupom (até 20 pagamentos).
szCodigos	PCHAR	60 (N)	Número do meio de pagamento (até 20 pagamentos).
szValores	PCHAR	280 (N)	Valor recebido no meio de pagamento (até 20 pagamentos).
szParcelas	PCHAR	60 (N)	Número de parcelas utilizadas no meio de pagamento (até 20 pagamentos).

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

O último documento emitido deve ser um Cupom Fiscal ou Comprovante Não Fiscal (exceto Comprovante de Cancelamento), com o respectivo meio de pagamento a ser cancelado.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szCOO[10], szDataHora[16], szVendaBruta[15], szNumeroSerie[21],  
szIndices[60], szCodigos[60], szValores[280], szParcelas[60];  
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_Pagamento("1", "2", "1000",  
"1", "Pagamento Estornado", szCOO, szDataHora, szVendaBruta,  
szNumeroSerie, szIndices, szCodigos, szValores, szParcelas);
```

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);  
StringBuilder szDataHora = new StringBuilder(16, 16);  
StringBuilder szVendaBruta = new StringBuilder(15, 15);  
StringBuilder szNumeroSerie = new StringBuilder(21, 21);  
StringBuilder szIndices = new StringBuilder(60, 60);  
StringBuilder szCodigos = new StringBuilder(60, 60);  
StringBuilder szValores = new StringBuilder(280, 280);  
StringBuilder szParcelas = new StringBuilder(60, 60);  
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_Pagamento("1", "2", "1000", "1",  
"Pagamento Estornado", szCOO, szDataHora, szVendaBruta, szNumeroSerie,  
szIndices, szCodigos, szValores, szParcelas);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String  
Dim szDataHora As String  
Dim szVendaBruta As String  
Dim szNumeroSerie As String  
Dim szIndices As String  
Dim szCodigos As String  
Dim szValores As String  
Dim szParcelas As String  
szCOO = Space(10)  
szDataHora = Space(16)  
szVendaBruta = Space(15)  
szNumeroSerie = Space(21)  
szIndices = Space(60)  
szCodigos = Space(60)  
szValores = Space(280)  
szParcelas = Space(60)  
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_Pagamento("1", "2", "1000", "1",  
"Pagamento Estornado", szCOO, szDataHora, szVendaBruta, szNumeroSerie,  
szIndices, szCodigos, szValores, szParcelas)
```

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;  
szDataHora: array[0..16] of Char;  
szVendaBruta: array[0..15] of Char;  
szNumeroSerie: array[0..21] of Char;  
szIndices: array[0..60] of Char;  
szCodigos: array[0..60] of Char;  
szValores: array[0..280] of Char;  
szParcelas: array[0..60] of Char;  
Retorno := EPSON_ESC_ECF_NaoFiscal_Cancelar_Pagamento('1', '2', '1000',  
'1', 'Pagamento Estornado', szCOO, szDataHora, szVendaBruta,  
szNumeroSerie, szIndices, szCodigos, szValores, szParcelas);
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));  
szDataHora := AnsiString(StringOfChar(' ', 16));  
szVendaBruta := AnsiString(StringOfChar(' ', 15));  
szNumeroSerie := AnsiString(StringOfChar(' ', 21));  
szIndices := AnsiString(StringOfChar(' ', 60));  
szCodigos := AnsiString(StringOfChar(' ', 60));  
szValores := AnsiString(StringOfChar(' ', 280));  
szParcelas := AnsiString(StringOfChar(' ', 60));  
Retorno := EPSON_ESC_ECF_NaoFiscal_Cancelar_Pagamento('1', '2', '1000',  
'1', 'Pagamento Estornado', PAnsiChar(szCOO), PAnsiChar(szDataHora),  
PAnsiChar(szVendaBruta), PAnsiChar(szNumeroSerie), PAnsiChar(szIndices),  
PAnsiChar(szCodigos), PAnsiChar(szValores), PAnsiChar(szParcelas)) ;
```

**4.2.95 EPSON\_ESC\_ECF\_NaoFiscal\_Cancelar\_PagamentoEX**

Esta função efetua a Troca de uma Forma de Pagamento por outra, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Cancelar_PagamentoEX(szIndiceEstornado:PChar;  
szIndiceEfetivado:PChar; szValor:PChar;  
szNumeroParcelas:PChar; szMensagem:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndiceEstornado	PCHAR	2 (N)	Índice do meio de pagamento cadastrado no ECF que será estornado.
szIndiceEfetivado	PCHAR	2 (N)	Índice do novo meio de pagamento cadastrado no ECF que será utilizado.
szValor	PCHAR	13 (N)	Valor do novo meio de pagamento
szNumeroParcelas	PCHAR	2 (N)	Número de parcelas do pagamento.
szMensagem	PCHAR	1024(O,A)	Mensagem que será impressa no comprovante.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

O último documento emitido deve ser um Cupom Fiscal ou Comprovante Não Fiscal (exceto Comprovante de Cancelamento), com o respectivo meio de pagamento a ser cancelado.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_PagamentoEX("1", "2", "1000",  
"1", "Pagamento Estornado");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Cancelar_PagamentoEX("1", "2", "1000",  
"1", "Pagamento Estornado")
```

**Exemplo em Delphi Delphi 7 / 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Cancelar_PagamentoEX('1', '2', '1000',  
'1', 'Pagamento Estornado');
```

**4.2.96 EPSON\_ESC\_ECF\_NaoFiscal\_Sangria\_Fundo\_Troco**

Esta função é usada para retirar uma quantia de dinheiro (Sangria) ou para inserir uma quantia de dinheiro (Fundo de Troco), seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_Troco(szTipo:PChar; szValor:PChar;  
szInformacoes:PChar; szCOO:PChar;  
szDataHora:PChar; szVendaBruta:PChar;  
szNumeroSerie:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szTipo	PCHAR	1 (N)	Tipo da operação. <b>1</b> – Fundo de Troco, <b>2</b> – Sangria.
szValor	PCHAR	13 (N)	Valor da operação.
szInformacoes	PCHAR	1024 (A)	Mensagem que será impressa no comprovante.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	10 (N)	COO do comprovante.
szDataHora	PCHAR	16 (A)	Data e Hora de fechamento do comprovante.
szVendaBruta	PCHAR	15 (N)	Venda Bruta.
szNumeroSerie	PCHAR	21 (N)	Número de série do ECF.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.

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

```
char szCOO[10], szDataHora[16], szVendaBruta[15], szNumeroSerie[21];  
Retorno = EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_Troco("1", "1000",  
"Realizando fundo de troco", szCOO, szDataHora, szVendaBruta,  
szNumeroSerie);
```

**Exemplo em C#:**

```
StringBuilder szCOO = new StringBuilder(10, 10);  
StringBuilder szDataHora = new StringBuilder(16, 16);  
StringBuilder szVendaBruta = new StringBuilder(15, 15);  
StringBuilder szNumeroSerie = new StringBuilder(21, 21);  
Retorno = EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_Troco("1", "1000",  
"Realizando fundo de troco", szCOO, szDataHora, szVendaBruta,  
szNumeroSerie);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCOO As String
Dim szDataHora As String
Dim szVendaBruta As String
Dim szNumeroSerie As String
szCOO = Space(10)
szDataHora = Space(16)
szVendaBruta = Space(15)
szNumeroSerie = Space(21)
Retorno = EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_Troco("1", "1000",
"Realizando fundo de troco", szCOO, szDataHora, szVendaBruta,
szNumeroSerie)
```

---

**Exemplo em Delphi 7:**

```
szCOO: array[0..10] of Char;
szDataHora: array[0..16] of Char;
szVendaBruta: array[0..15] of Char;
szNumeroSerie: array[0..21] of Char;
Retorno := EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_Troco('1', '1000',
'Realizando fundo de troco', szCOO, szDataHora, szVendaBruta,
szNumeroSerie);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCOO := AnsiString(StringOfChar(' ', 10));
szDataHora := AnsiString(StringOfChar(' ', 16));
szVendaBruta := AnsiString(StringOfChar(' ', 15));
szNumeroSerie := AnsiString(StringOfChar(' ', 21));
Retorno := EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_Troco('1', '1000',
'Realizando fundo de troco', PAnsiChar(szCOO), PAnsiChar(szDataHora),
PAnsiChar(szVendaBruta), PAnsiChar(szNumeroSerie));
```

**4.2.97 EPSON\_ESC\_ECF\_NaoFiscal\_Sangria\_Fundo\_TrocoEX**

Esta função é usada para retirar uma quantia de dinheiro (Sangria) ou para inserir uma quantia de dinheiro (Fundo de Troco), seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_TrocoEX(szTipo:PChar; szValor:PChar;  
szInformacoes:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szTipo	PCHAR	1 (N)	Tipo da operação. <b>1</b> – Fundo de Troco, <b>2</b> - Sangria
szValor	PCHAR	13 (N)	Valor da operação.
szInformacoes	PCHAR	1024 (A)	Mensagem que será impressa no comprovante.

**Saídas:**

Nenhuma.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_TrocoEX("1", "1000",  
"Realizando fundo de troco");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_TrocoEX("1", "1000",  
"Realizando fundo de troco")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Sangria_Fundo_TrocoEX('1', '1000',  
'Realizando fundo de troco');
```



**4.2.98 EPSON\_ESC\_ECF\_NaoFiscal\_Imprime\_QRCode**

Esta função é usada para imprimir um QRCode no Relatório Gerencial aberto ou no Comprovante de Crédito ou Débito aberto, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_NaoFiscal_Imprime_QRCode (szModelo:PChar; szModulo:PChar;  
                                                    szNivelCorrecao:PChar, szDados:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szModelo	PCHAR	2 (N)	Modelo do QRCode que será impresso. <b>49</b> – Modelo 01, <b>50</b> – Modelo 2.
szModulo	PCHAR	2 (N)	Tamanho do módulo do QRCode (1 a 16).
szNivelCorrecao	PCHAR	1 (N)	Nível de correção do QRCode. <b>48</b> – Nível L (7%), <b>49</b> – Nível M (15%), <b>50</b> – Nível Q (25%), <b>51</b> Nível H (30%).
szDados	PCHAR	1024 (A)	Texto que será adicionado no QRCode.

**Saídas:**

Nenhuma.

**Retornos:**

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

**Requisitos:**

Um documento não fiscal aberto e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Imprime_QRCode("50", "3", "49",  
"www.epson.com.br");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_NaoFiscal_Imprime_QRCode("50", "3", "49",  
"www.epson.com.br")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_NaoFiscal_Imprime_QRCode('50', '3', '49',  
'www.epson.com.br');
```

### 4.3 Grupo de Relatórios Fiscais

Estas funções são usadas para controlar a Jornada Fiscal e retirar informações dela.

#### 4.3.1 EPSON\_RelatorioFiscal\_LeituraX

Esta função imprime uma Leitura X.

---

**Sintaxe:**

```
function EPSON_RelatorioFiscal_LeituraX():Integer;StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

---

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, documentos fechados e o ECF com papel.

---

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

```
Retorno = EPSON_RelatorioFiscal_LeituraX( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_RelatorioFiscal_LeituraX( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_RelatorioFiscal_LeituraX( );
```

### 4.3.2 EPSON\_RelatorioFiscal\_RZ

Esta função imprime uma Redução Z e fecha a Jornada Fiscal corrente.

Este comando gera automaticamente o Ato/COTEPE 17/04 para a CAT52. Vide Capítulo 7.

#### Sintaxe:

EPSON\_RelatorioFiscal\_RZ( pszData:PChar; pszHora:PChar; dwHorarioVerao:Integer; pszCRZ:PChar );  
Integer;StdCall; External 'InterfaceEpson.dll';

#### Entradas:

Variável	Tipo	Tam.	Descrição
pszData	PCHAR	8 (O,D)	Nova data (ddmmaaaa). Preencher apenas se desejar alterar o relório do ECF.
pszHora	PCHAR	6 (O,T)	Novo horário (hhmmss). Preencher apenas se desejar alterar o relório do ECF.
dwHorarioVerao	INTEGER	-	Parâmetro de entrada ou saída de horário de verão. <b>0</b> – Sair do horário de verão. <b>1</b> – Entrar em horário de verão. <b>qualquer outro valor</b> – Não afeta o estado atual.

#### Saídas:

Variável	Tipo	Tam.	Descrição
pszCRZ	PCHAR	4 (N)	Valor do contador CRZ com 4 dígitos (max).

#### Retornos:

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

#### Requisitos:

A Jornada fiscal deve estar aberta e o ECF com papel. Se a jornada fiscal corresponder ao dia atual não será mais possível emitir cupons até que se inicie a jornada do dia seguinte.

#### Observação:

Se a jornada fiscal corresponder ao dia atual não será mais possível emitir cupons até que se inicie a jornada do dia seguinte.

Os campos pszData e pszHora são opcionais e somente devem ser utilizados para alterar o horário da impressora. Esta mudança de horário é limitada em até cinco minutos para mais ou para menos em relação ao relório do ECF. Caso se faça o retrocesso do relógio, este não pode ser inferior ao horário do último documento impresso.

A entrada ou saída de horário de verão pode ser efetuada também pelo Registro do Windows através da chave HorariodeVerao localizada no caminho HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson. Seus valores possíveis são os mesmos informados na tabela acima. A configuração desta chave prevalece sobre a opção informada no comando da ReduçãoZ. Para desativar o controle de horário de verão do Registro do Windows, configurar a chave com a opção 2.

---

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

```
char szCRZ[5];  
Retorno = EPSON_RelatorioFiscal_RZ( "", "", 2, szCRZ );
```

---

**Exemplo em C# :**

```
StringBuilder szCRZ = new StringBuilder(5,5);  
Retorno = EPSON_RelatorioFiscal_RZ( "", "", 2, szCRZ );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET :**

```
Dim szCRZ As String  
szCRZ = Space(4)  
Retorno = EPSON_RelatorioFiscal_RZ( "", "", 2, szCRZ );
```

---

**Exemplo em Delphi 7:**

```
szCRZ: array[0..5] of Char;  
Retorno := EPSON_RelatorioFiscal_RZ(' ', ' ', 2, szCRZ );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCRZ := AnsiString(StringOfChar(' ', 5));  
Retorno := EPSON_RelatorioFiscal_RZ(' ', ' ', 2, PAnsiChar(szCRZ));
```

### 4.3.3 EPSON\_RelatorioFiscal\_RZEX

Esta função imprime um Relatório Z e fecha a Jornada Fiscal corrente.  
Este comando gera automaticamente o Ato/COTEPE 17/04 para a CAT52. Vide Capítulo 7.

**Sintaxe:**

```
function EPSON_RelatorioFiscal_RZEX(dwHorarioVerao:Integer):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwHorarioVerao	INTEGER	-	Parâmetro de entrada ou saída de horário de verão. <b>0</b> – Sair do horário de verão. <b>1</b> – Entrar em horário de verão. <b>qualquer outro valor</b> – Não afeta o estado atual.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta e o ECF com papel.

A entrada ou saída de horário de verão pode ser efetuada também pelo Registro do Windows através da chave HorariodeVerao localizada no caminho HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson. Seus valores possíveis são os mesmos informados na tabela acima. A configuração desta chave prevalece sobre a opção informada no comando da ReduçãoZ. Para desativar o controle de horário de verão do Registro do Windows, configurar a chave com a opção 2.

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

```
Retorno = EPSON_RelatorioFiscal_RZEX( 2 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_RelatorioFiscal_RZEX( 2 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_RelatorioFiscal_RZEX( 2 );
```

## 4.3.4 EPSON\_RelatorioFiscal\_Leitura\_MF

Esta função imprime uma Leitura da Memória Fiscal, salva em um arquivo ou armazena em uma variável (buffer).

### Sintaxe:

```
function EPSON_RelatorioFiscal_Leitura_MF( pszInicio:PChar; pszFim:PChar; dwTipolImpressao:Integer;
                                           pszDadosMF:PChar; pszArquivo:PChar;
                                           pdwTamanhoBuffer:PInteger; dwTamBuffer:Integer): Integer;
StdCall; External 'InterfaceEpson.dll';
```

### Entradas:

Variável	Tipo	Tam.	Descrição																														
pszInicio	PCHAR	4 (O,N)	CRZ ou data inicial (ddmmaaaa).																														
pszFim	PCHAR	4 (O,N)	CRZ ou data final (ddmmaaaa).																														
dwTipolImpressao	INTEGER	-	<table><tr><th>Parcelas</th><th>Valor</th><th>Descrição</th></tr><tr><td rowspan="4">X</td><td>0</td><td>Modo Completo, faixa em CRZ</td></tr><tr><td>1</td><td>Modo Completo, faixa em Data</td></tr><tr><td>2</td><td>Modo Simplificado, faixa em CRZ</td></tr><tr><td>3</td><td>Modo Simplificado, faixa em Data</td></tr><tr><td colspan="3">+</td></tr><tr><td rowspan="4">Y</td><td>4</td><td>Imprime</td></tr><tr><td>8</td><td>Copia para variável (buffer)</td></tr><tr><td>16</td><td>Armazena em um arquivo</td></tr><tr><td>24</td><td>Copia para variável (buffer) e armazena em um arquivo.</td></tr><tr><td colspan="3">dwTipolImpressao = X+Y</td></tr><tr><td colspan="3">Exemplo: Modo completo, faixa em data em uma variável. <u>dwTipolImpressao = 1 + 8 = 9</u></td></tr></table>	Parcelas	Valor	Descrição	X	0	Modo Completo, faixa em CRZ	1	Modo Completo, faixa em Data	2	Modo Simplificado, faixa em CRZ	3	Modo Simplificado, faixa em Data	+			Y	4	Imprime	8	Copia para variável (buffer)	16	Armazena em um arquivo	24	Copia para variável (buffer) e armazena em um arquivo.	dwTipolImpressao = X+Y			Exemplo: Modo completo, faixa em data em uma variável. <u>dwTipolImpressao = 1 + 8 = 9</u>		
			Parcelas	Valor	Descrição																												
			X	0	Modo Completo, faixa em CRZ																												
				1	Modo Completo, faixa em Data																												
				2	Modo Simplificado, faixa em CRZ																												
				3	Modo Simplificado, faixa em Data																												
			+																														
			Y	4	Imprime																												
				8	Copia para variável (buffer)																												
				16	Armazena em um arquivo																												
				24	Copia para variável (buffer) e armazena em um arquivo.																												
dwTipolImpressao = X+Y																																	
Exemplo: Modo completo, faixa em data em uma variável. <u>dwTipolImpressao = 1 + 8 = 9</u>																																	
pszDadosMF	PCHAR	dwTamBuffer	Variável para armazenamento dos dados lidos da Memória Fiscal.																														
pszArquivo	PCHAR	10KB	Nome do arquivo onde serão armazenados os dados (não obrigatório). * vale lembrar que uma leitura da MF. Pode chegar a ocupar 3 MB dependendo do tempo de utilização da maquina.																														

**Saídas:**

Variável	Tipo	Tam.	Descrição
pdwTamanhoBuffer	INTEGER	-	Número de bytes efetivamente copiados para a variável.
dwTamBuffer	INTEGER	-	Tamanho da variável (pszBuffer) para armazenamento dos dados. Este valor informa a LIB o número máximo de bytes a serem copiados para a variável para que não haja perda de dados.

**Retornos:**

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

**Observação:**

Por padrão a Leitura da MF em arquivo será gerada com assinatura digital EAD, para desabilitar ou habilitar essa assinatura deve ser utilizado o comando EPSON\_Config\_Habilita\_EAD.

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.

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

```
char pszDadosMF[1024];  
int tambuff;  
Retorno = EPSON_RelatorioFiscal_Leitura_MF( "1", "10", 24, pszDados,  
"c:\\Temp\\teste.txt", &tambuff, 1024 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim tambuff As Integer  
Dim pszDadosMF As String  
pszDadosMF = Space(1024)  
Retorno = EPSON_RelatorioFiscal_Leitura_MF( "1", "10", 16, pszDadosMF,  
"C:\\temp\\teste.txt", tambuff, 1024)
```

**Exemplo em Delphi 7:**

```
iTamBuff: Integer;  
pszDadosMF: array[0..1024] of Char;  
Retorno := EPSON_RelatorioFiscal_Leitura_MF( '1', '10', 16, pszDadosMF, '  
c:\\Temp\\teste.txt', @iTamBuff, 1024 );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
tambuff: Integer;  
pszDadosMF: AnsiString;  
Retorno := EPSON_RelatorioFiscal_Leitura_MF( '1', '10', 4,  
PAnsiChar(pszDadosMF), '', @tambuff, 10);
```

**4.3.5 EPSON\_RelatorioFiscal\_Salvar\_LeituraX**

Esta função guarda uma Leitura X no disco.

**Sintaxe:**

```
function EPSON_RelatorioFiscal_Salvar_LeituraX(pszNomeArquivo:PChar):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNomeArquivo	PCHAR	1024	Nome do arquivo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Documentos fechados.

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

```
Retorno = EPSON_RelatorioFiscal_Salvar_LeituraX("c:\\Temp\\LeituraX.txt");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_RelatorioFiscal_Salvar_LeituraX( "c:\\Temp\\LeituraX.txt" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno:=EPSON_RelatorioFiscal_Salvar_LeituraX('c:\\Temp\\LeituraX.txt');
```



### 4.3.6 EPSON\_RelatorioFiscal\_Abrir\_Jornada

Esta função abre a Jornada Fiscal.

---

**Sintaxe:**

```
function EPSON_RelatorioFiscal_Abrir_Jornada():Integer; StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

---

**Requisitos:**

A Jornada fiscal deve estar fechada, documentos fechados e o ECF com papel. Se a jornada correspondente ao dia atual já foi encerrada, esta função retornará erro.

---

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

```
Retorno = EPSON_RelatorioFiscal_Abrir_Jornada( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_RelatorioFiscal_Abrir_Jornada( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_RelatorioFiscal_Abrir_Jornada( );
```

**4.3.7 EPSON\_ESC\_ECF\_RelatorioFiscal\_LeituraX**

Esta função imprime uma Leitura X, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_RelatorioFiscal_LeituraX():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar fechada, documentos fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_LeituraX( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_LeituraX( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_RelatorioFiscal_LeituraX( );
```

**4.3.8 EPSON\_ESC\_ECF\_RelatorioFiscal\_LeituraXEX**

Esta função retorna os dados do espelho ou imprime uma Leitura X, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_RelatorioFiscal_LeituraXEX(szTipo:PChar; szDados:PChar):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szTipo	PCHAR	1 (N)	Tipo da Leitura. 0 – Impressa. 1 – Dados do espelho.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szDados	PCHAR	-	Dados do espelho da LX (caso o tipo selecionado seja 1). O tamanho do espelho pode variar de acordo com a configuração do ECF.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar fechada, documentos fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szDados[8192];  
Retorno = EPSON_ESC_ECF_RelatorioFiscal_LeituraXEX( "1",  szDados );
```

**Exemplo em C#:**

```
StringBuilder szDados = new StringBuilder(8192, 8192);  
Retorno = EPSON_ESC_ECF_RelatorioFiscal_LeituraXEX( "1", szDados );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDados As String  
szDados = Space(8192)  
Retorno = EPSON_ESC_ECF_RelatorioFiscal_LeituraXEX( "1", szDados )
```

---

**Exemplo em Delphi 7:**

```
szDados: array[0..8192] of Char;  
Retorno := EPSON_ESC_ECF_RelatorioFiscal_LeituraXEX( '1', szDados );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDados := AnsiString(StringOfChar(' ', 8192));  
Retorno := EPSON_ESC_ECF_RelatorioFiscal_LeituraXEX( '1',  
PAnsiChar(szDados));
```

**4.3.9 EPSON\_ESC\_ECF\_RelatorioFiscal\_RZ**

Esta função imprime uma Redução Z e fecha a Jornada Fiscal corrente, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_RelatorioFiscal_RZ(szData:PChar; szHora:PChar; szTransmite:PChar;  
szDataMovimento:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szData	PCHAR	8 (O,D)	Nova data (ddmmaaaa). Preencher apenas se desejar alterar o relório do ECF.
szHora	PCHAR	6 (O,T)	Novo horário (hhmmss). Preencher apenas se desejar alterar o relório do ECF.
szTransmite	PCHAR	1 (N)	Transmite dados RZ para o Fisco. <b>0</b> – Não transmite. <b>1</b> – Transmite.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szDataMovimento	PCHAR	9 (A)	Data de movimento da RZ impressa.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta e o ECF com papel. Se a jornada fiscal corresponder ao dia atual não será mais possível emitir cupons até que se inicie a jornada do dia seguinte.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szDataMovimento [9];  
Retorno = EPSON_ESC_ECF_RelatorioFiscal_RZ("", "", "0", szDataMovimento);
```

**Exemplo em C#:**

```
StringBuilder szDataMovimento = new StringBuilder(9,9);  
Retorno = EPSON_ESC_ECF_RelatorioFiscal_RZ("", "", "0", szDataMovimento);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDataMovimento As String  
szDataMovimento = Space(9)  
Retorno = EPSON_ESC_ECF_RelatorioFiscal_RZ("", "", "0", szDataMovimento)
```

---

**Exemplo em Delphi 7:**

```
szDataMovimento: array[0..9] of Char;  
Retorno := EPSON_ESC_ECF_RelatorioFiscal_RZ( '', '', '0',  
szDataMovimento);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDataMovimento := AnsiString(StringOfChar(' ', 9));  
Retorno := EPSON_ESC_ECF_RelatorioFiscal_RZ('', '', '0',  
PAnsiChar(szDataMovimento));
```

**4.3.10 EPSON\_ESC\_ECF\_RelatorioFiscal\_RZEX**

Esta função imprime uma Redução Z e fecha a Jornada Fiscal corrente, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_RelatorioFiscal_RZEX(szData:PChar; szHora:PChar;  
                                             szTransmite:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szData	PCHAR	8 (O,D)	Nova data (ddmmaaaa). Preencher apenas se desejar alterar o relório do ECF.
szHora	PCHAR	6 (O,T)	Novo horário (hhmmss). Preencher apenas se desejar alterar o relório do ECF.
szTransmite	PCHAR	1 (N)	Transmite dados RZ para o Fisco. <b>0</b> – Não transmite. <b>1</b> – Transmite.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta e o ECF com papel. Se a jornada fiscal corresponder ao dia atual não será mais possível emitir cupons até que se inicie a jornada do dia seguinte.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_RZEX("", "", "0");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_RZEX("", "", "0")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_RelatorioFiscal_RZEX('', '', '0');
```

**4.3.11 EPSON\_ESC\_ECF\_RelatorioFiscal\_Leitura\_MF**

Esta função imprime uma Leitura da Memória Fiscal ou armazena em uma variável (buffer), seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_RelatorioFiscal_Leitura_MF(szMidia:PChar; szTipo:PChar; szModo:PChar;  
szInicial:PChar; szFinal:PChar;  
szDados:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szMidia	PCHAR	1 (N)	Tipo da operação que será executada. <b>0</b> – Imprime leitura. <b>1</b> – Retorna espelho da leitura através da variável.
szTipo	PCHAR	1 (N)	Tipo do relatório que será gerado. <b>1</b> – Analítico. <b>2</b> – Simplificado.
szModo	PCHAR	1 (N)	Modo de geração do relatório. <b>1</b> – Por Data. <b>2</b> – Por CRZ.
szInicial	PCHAR	8 (N)	Valor Inicial (Data ou CRZ).
szFinal	PCHAR	8 (N)	Valor Final (Data ou CRZ).

**Saídas:**

Variável	Tipo	Tam.	Descrição
szDados	PCHAR	-	Dados do espelho da LMF (caso a midia selecionada seja 1). O tamanho do espelho pode variar de acordo com a configuração do ECF e o periodo escolhido.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szDados[8192];  
Retorno = EPSON_ESC_ECF_RelatorioFiscal_Leitura_MF("1", "1", "2", "1",  
"3500", szDados);
```



---

**Exemplo em C#:**

```
StringBuilder szDados = new StringBuilder(8192,8192);  
Retorno = EPSON_ESC_ECF_RelatorioFiscal_Leitura_MF("1", "1", "2", "1",  
"3500", szDados);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDados As String  
szDados = Space(8192)  
Retorno = EPSON_ESC_ECF_RelatorioFiscal_Leitura_MF("1", "1", "2", "1",  
"3500", szDados)
```

---

**Exemplo em Delphi 7:**

```
szDados: array[0..8192] of Char;  
Retorno := EPSON_ESC_ECF_RelatorioFiscal_Leitura_MF('1', '1', '2', '1',  
'3500', szDados);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDados := AnsiString(StringOfChar(' ', 8192));  
Retorno := EPSON_ESC_ECF_RelatorioFiscal_Leitura_MF('1', '1', '2', '1',  
'3500', PAnsiChar(szDados));
```

**4.3.12 EPSON\_ESC\_ECF\_RelatorioFiscal\_Leitura\_MFEX**

Esta função imprime uma Leitura da Memória Fiscal ou armazena em uma variável (buffer), seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_RelatorioFiscal_Leitura_MFEX(szTipo:PChar; szModo:PChar; szInicial:PChar;  
szFinal:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szTipo	PCHAR	1 (N)	Tipo do relatório que será gerado. 1 – Analítico. 2 – Simplificado.
szModo	PCHAR	1 (N)	Modo de geração do relatório. 1 – Por Data. 2 – Por CRZ.
szInicial	PCHAR	8 (N)	Valor Inicial (Data ou CRZ).
szFinal	PCHAR	8 (N)	Valor Final (Data ou CRZ).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

---

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

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_Leitura_MFEX("1","2","1", "3500");
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_Leitura_MFEX("1","2","1","3500")
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno:=EPSON_ESC_ECF_RelatorioFiscal_Leitura_MFEX('1','2','1', '3500');
```

**4.3.13 EPSON\_ESC\_ECF\_RelatorioFiscal\_Imprime\_RTD\_RZ**

Esta função envia os dados do Recibo de Transmissão de Dados, para a impressão da Redução Z, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_RelatorioFiscal_Imprime_RTD_RZ(szMensagem:PChar):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szMensagem	PCHAR	1024 (A)	Mensagem do RTD.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_Imprime_RTD_RZ ("Mensagem RTD");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_Imprime_RTD_RZ ("Mensagem RTD")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_RelatorioFiscal_Imprime_RTD_RZ ('Mensagem RTD');
```

**4.3.14 EPSON\_ESC\_ECF\_RelatorioFiscal\_Reimprimir\_MFD**

Esta função reimprime os documentos gravados na MFD do ECF, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_RelatorioFiscal_Reimprimir_MFD(szTipo:PChar; szInicial:PChar;  
szFinal:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szTipo	PCHAR	1 (N)	Modo de geração do relatório. 1 – Por Data. 2 – Por COO.
szInicial	PCHAR	9 (N)	Valor Inicial (Data ou COO).
szFinal	PCHAR	9 (N)	Valor Final (Data ou COO).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_Reimprimir_MFD("2", "1", "10" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_RelatorioFiscal_Reimprimir_MFD("2", "1", "10" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_RelatorioFiscal_Reimprimir_MFD('2', '1', '10' );
```

## 4.4 Grupo Obter Informações da Impressora.

Estas funções são usadas para a leitura das configurações e informações da Impressora Fiscal:

### 4.4.1 EPSON\_Obter\_Dados\_Usuario

Esta função lê os dados de identificação do usuário. São eles: CNPJ, IE e IM.

---

**Sintaxe:**

```
function EPSON_Obter_Dados_Usuario(pszDadosUsuario:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDadosUsuario	PCHAR	49	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-17	:	Número CNPJ
Byte 18-32	:	Inscrição Estadual
Byte 33-47	:	Inscrição Municipal

Todos os campos têm tamanho fixo.

---

**Retornos:**

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

---

**Requisitos:**

Nenhum.

---

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

```
char szDadosUsuario[49];  
Retorno = EPSON_Obter_Dados_Usuario ( szDadosUsuario );
```

---

**Exemplo em C# :**

```
StringBuilder szDadosUsuario = new StringBuilder(49, 49);  
Retorno = EPSON_Obter_Dados_Usuario ( szDadosUsuario );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDadosUsuario As String  
szDadosUsuario = Space(49)  
Retorno = EPSON_Obter_Dados_Usuario ( szDadosUsuario )
```

---

**Exemplo em Delphi 7:**

```
szDadosUsuario: array[0..49] of Char;  
Retorno := EPSON_Obter_Dados_Usuario ( szDadosUsuario );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDadosUsuario := AnsiString(StringOfChar(' ', 49));  
Retorno := EPSON_Obter_Dados_Usuario(PAnsiChar(szDadosUsuario));
```

#### 4.4.2 EPSON\_Obter\_Tabela\_Aliquotas

Esta função retorna todos as situações tributárias cadastradas na impressora fiscal. Os valores informados em cada situação tributária, são relativos à jornada fiscal atual, e estes valores são reiniciados a cada nova jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Tabela_Aliquotas(pszTabelaAliquotas:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTabelaAliquotas	PCHAR	553 bytes	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-1 : Símbolo da situação tributária

Byte 2-5 : Valor da taxa

Byte 6-22 : Valor vendido

...

O buffer retornado na variável pszTabelaAliquotas, é composto por 24 sequencias de 23 caracteres cada. Cada sequência corresponderá a uma das 24 situações tributárias possíveis de serem cadastradas no ECF, sendo 18 alíquotas de ICMS/ISSQN cadastráveis pelo usuário e 6 alíquotas pré cadastradas (F, I, N, FS, IS e NS), totalizando 552 bytes a serem retornados. Caso o número de alíquotas cadastradas no ECF seja menor que o limite, as sequências referentes a estas posições na tabela de alíquotas serão retornadas preenchidas com espaços.

Este buffer termina com um byte nulo (\0) para sinalizar o fim de string.

**Retornos:**

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

**Requisitos:**

Nenhum.



---

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

```
char szTabelaAliquotas[553];  
Retorno = EPSON_Obter_Tabela_Aliquotas ( szTabelaAliquotas );
```

---

**Exemplo em C# :**

```
StringBuilder szTabelaAliquotas = new StringBuilder(553, 553);  
Retorno = EPSON_Obter_Tabela_Aliquotas ( szTabelaAliquotas );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTabelaAliquotas As String  
szTabelaAliquotas = Space(553)  
Retorno = EPSON_Obter_Tabela_Aliquotas ( szTabelaAliquotas )
```

---

**Exemplo em Delphi 7:**

```
szTabelaAliquotas: array[0..553] of Char;  
Retorno := EPSON_Obter_Tabela_Aliquotas ( szTabelaAliquotas );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTabelaAliquotas := AnsiString(StringOfChar(' ', 553));  
Retorno := EPSON_Obter_Tabela_Aliquotas(PAnsiChar(szTabelaAliquotas));
```

**4.4.3 EPSON\_Obter\_Tabela\_Aliquotas\_Cupom**

Esta função retorna todos as situações tributárias referentes ao cupom fiscal atual. Os valores informados em cada situação tributária, são relativos ao cupom fiscal em andamento, e estes valores são reiniciados a cada novo cupom fiscal.

**Sintaxe:**

```
function EPSON_Obter_Tabela_Aliquotas_Cupom( pszTabelaAliquotas:PChar; pszTotalICMS:PChar;  
                                             pszTotalISS:PChar):Integer;StdCall; External  
                                             'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTabelaAliquotas	PCHAR	553 bytes	Dados recebidos da impressora.
pszTotalICMS	PCHAR	18 bytes	Total devido de ICMS no cupom.
pszTotalISS	PCHAR	18 bytes	Total devido de ISS no cupom.

Os dados da variável pszTabelaAliquotas serão retornados no seguinte formato:

Byte 0-1 : Símbolo da situação tributária

Byte 2-5 : Valor da taxa

Byte 6-22 : Valor vendido no cupom.

...

O buffer retornado na variável pszTabelaAliquotas, é composto por 24 sequências de 23 caracteres cada. Cada sequência corresponderá a uma das 24 situações tributárias possíveis de serem cadastradas no ECF, sendo 18 alíquotas de ICMS/ISSQN cadastráveis pelo usuário e 6 alíquotas pré cadastradas (F, I, N, FS, IS e NS), totalizando 552 bytes a serem retornados. Caso o número de alíquotas cadastradas no ECF seja menor que o limite, as sequências referentes a estas posições na tabela de alíquotas serão retornadas preenchidas com espaços.

Este buffer termina com um byte nulo (\0) para sinalizar o fim de string.

**Retornos:**

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

**Requisitos:**

Cupom Fiscal aberto. Caso não exista um cupom aberto no momento da execução do comando, os valores serão preenchidos com zero.

---

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

```
char szTabelaAliquotas[553];  
char szTotalICMS[18];  
char szTotalISS[18];  
Retorno = EPSON_Obter_Tabela_Aliquotas_Cupom( szTabelaAliquotas,  
szTotalICMS, szTotalISS );
```

---

**Exemplo em C# :**

```
StringBuilder szTabelaAliquotas = new StringBuilder(553, 553);  
StringBuilder szTotalICMS = new StringBuilder(18, 18);  
StringBuilder szTotalISS = new StringBuilder(18, 18);  
Retorno = EPSON_Obter_Tabela_Aliquotas_Cupom( szTabelaAliquotas,  
szTotalICMS, szTotalISS );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTabelaAliquotas As String  
szTabelaAliquotas = Space(553)  
Dim szTotalICMS As String  
szTotalICMS = Space(18)  
Dim szTotalISS As String  
szTotalISS = Space(18)  
Retorno = EPSON_Obter_Tabela_Aliquotas_Cupom( szTabelaAliquotas,  
szTotalICMS, szTotalISS )
```

---

**Exemplo em Delphi 7:**

```
szTabelaAliquotas: array[0..553] of Char;  
szTotalICMS: array[0..18] of Char;  
szTotalISS: array[0..18] of Char;  
Retorno := EPSON_Obter_Tabela_Aliquotas_Cupom(szTabelaAliquotas,  
szTotalICMS, szTotalISS);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTabelaAliquotas := AnsiString(StringOfChar(' ', 553));  
szTotalICMS := AnsiString(StringOfChar(' ', 18));  
szTotalISS := AnsiString(StringOfChar(' ', 18));  
Retorno :=  
EPSON_Obter_Tabela_Aliquotas_Cupom(PAnsiChar(szTabelaAliquotas),  
PAnsiChar(szTotalICMS), PAnsiChar(szTotalISS));
```

#### 4.4.4 EPSON\_Obter\_Tabela\_Pagamentos

Esta função retorna todos os meios de pagamentos cadastrados na impressora fiscal. Os valores informados em cada meio de pagamento, são relativos à jornada fiscal atual, e estes valores são reiniciados a cada nova jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Tabela_Pagamentos(pszTabelaPagamentos:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTabelaPagamentos	PCHAR	881 bytes	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-1	:	Número da posição
Byte 2-16	:	Descrição do meio de pagamento
Byte 17-29	:	Valor acumulado
Byte 30-42	:	Valor pago no documento
Byte 43	:	'N' – Não vinculado ao CCD 'S' – Vinculado ao CCD

O buffer retornado na variável pszTabelaPagamentos, é composto por 20 sequencias de 44 caracteres cada. Cada sequência corresponderá a uma das 20 formas de pagamento possíveis de serem cadastradas no ECF, totalizando 880 bytes a serem retornados.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szTabelaPagamentos[881];  
Retorno = EPSON_Obter_Tabela_Pagamentos ( szTabelaPagamentos );
```

---

**Exemplo em C# :**

```
StringBuilder szTabelaPagamentos = new StringBuilder(881, 881);  
Retorno = EPSON_Obter_Tabela_Pagamentos ( szTabelaPagamentos );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTabelaPagamentos As String  
szTabelaPagamentos = Space(881)  
Retorno = EPSON_Obter_Tabela_Pagamentos ( szTabelaPagamentos )
```

---

**Exemplo em Delphi 7:**

```
szTabelaPagamentos: array[0..881] of Char;  
Retorno := EPSON_Obter_Tabela_Pagamentos ( szTabelaPagamentos );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTabelaPagamentos := AnsiString(StringOfChar(' ', 881));  
Retorno := EPSON_Obter_Tabela_Pagamentos(PAnsiChar(szTabelaPagamentos));
```

#### 4.4.5 EPSON\_Obter\_Tabela\_NaoFiscais

Esta função retorna todos os totalizadores não fiscais cadastrados na impressora fiscal. Os valores informados em cada totalizador não fiscal, são relativos à jornada fiscal atual, e estes valores são reiniciados a cada nova jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Tabela_NaoFiscais(pszTabelaNaoFiscais:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTabelaNaoFiscais	PCHAR	681 bytes	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-1	:	Número da posição.
Byte 2-16	:	Descrição do totalizador não fiscal
Byte 17-29	:	Valor acumulado
Byte 30-33	:	Valor do contador específico do totalizador não fiscal
...		

O buffer retornado na variável pszTabelaNaoFiscais, é composto por 20 sequências de 34 caracteres cada. Cada sequência corresponderá a uma das 20 operações não fiscais possíveis de serem cadastradas no ECF, totalizando 680 bytes a serem retornados.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szTabelaNaoFiscais[681];  
Retorno = EPSON_Obter_Tabela_NaoFiscais ( szTabelaNaoFiscais );
```

---

**Exemplo em C# :**

```
StringBuilder szTabelaNaoFiscais = new StringBuilder(681, 681);  
Retorno = EPSON_Obter_Tabela_NaoFiscais ( szTabelaNaoFiscais );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTabelaNaoFiscais As String  
szTabelaNaoFiscais = Space(681)  
Retorno = EPSON_Obter_Tabela_NaoFiscais ( szTabelaNaoFiscais )
```

---

**Exemplo em Delphi 7:**

```
szTabelaNaoFiscais: array[0..681] of Char;  
Retorno := EPSON_Obter_Tabela_NaoFiscais ( szTabelaNaoFiscais );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTabelaNaoFiscais := AnsiString(StringOfChar(' ', 681));  
Retorno := EPSON_Obter_Tabela_NaoFiscais(PAnsiChar(szTabelaNaoFiscais));
```

**4.4.6 EPSON\_Obter\_Tabela\_Relatorios\_Gerenciais**

Esta função retorna todos os relatórios gerenciais cadastrados na impressora fiscal. Os contadores específicos dos relatórios gerenciais informados, são relativos à jornada fiscal atual, e estes contadores são reiniciados a cada nova jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Tabela_Relatorios_Gerenciais(pszTabelaRelatoriosGerenciais:PChar):Integer;StdCall;  
    External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTabelaRelatoriosGerenciais	PCHAR	421 bytes	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-1	:	Número da posição
Byte 2-16	:	Descrição do relatório gerencial
Byte 17-20	:	Contador específico do relatório gerencial

O buffer retornado na variável pszTabelaRelatoriosGerencias, é composto por 20 sequências de 420 caracteres cada. Cada sequência corresponderá a um dos 20 Relatórios Gerenciais possíveis de serem cadastradas no ECF, totalizando 420 bytes a serem retornados.

**Retornos:**

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

**Requisitos:**

Nenhum.



---

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

```
char szTabelaRelatoriosGerenciais[421];  
Retorno = EPSON_Obter_Tabela_Relatorios_Gerenciais(  
szTabelaRelatoriosGerenciais );
```

---

**Exemplo em C# :**

```
StringBuilder szTabelaRelatoriosGerenciais = new StringBuilder(421, 421);  
Retorno = EPSON_Obter_Tabela_Relatorios_Gerenciais (  
szTabelaRelatoriosGerenciais );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTabelaRelatoriosGerenciais As String  
szTabelaRelatoriosGerenciais = Space(421)  
Retorno = EPSON_Obter_Tabela_Relatorios_Gerenciais(  
szTabelaRelatoriosGerenciais )
```

---

**Exemplo em Delphi 7:**

```
szTabelaRelatoriosGerenciais: array[0..421] of Char;  
Retorno := EPSON_Obter_Tabela_Relatorios_Gerenciais (  
szTabelaRelatoriosGerenciais );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTabelaRelatoriosGerenciais := AnsiString(StringOfChar(' ', 421));  
Retorno := EPSON_Obter_Tabela_Relatorios_Gerenciais(PAnsiChar  
(szTabelaRelatoriosGerenciais));
```

#### 4.4.7 EPSON\_Obter\_Total\_Cancelado

Esta função lê o valor total cancelado. Os valores informados, são relativos à jornada fiscal atual, e estes valores são reiniciados a cada nova jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Total_Cancelado(pszTotalCancelado:PChar):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalCancelado	PCHAR	52	Dados recebidos da impressora

Os dados serão retornados no seguinte formato:

Byte 0-16 : Total cancelado em ICMS

Byte 17-33 : Total cancelado em ISS

Byte 34-50 : Total cancelado Não-Fiscal

Todos os campos são de tamanho fixo e completados com zeros.

**Retornos:**

FUNC\_SUCESSO Operação realizada com sucesso.

FUNC\_ERRO Erro durante a execução.

**Requisitos:**

Nenhum.

---

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

```
char szTotalCancelado[52];  
Retorno = EPSON_Obter_Total_Cancelado ( szTotalCancelado );
```

---

**Exemplo em C# :**

```
StringBuilder szTotalCancelado = new StringBuilder(52, 52);  
Retorno = EPSON_Obter_Total_Cancelado ( szTotalCancelado );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalCancelado As String  
szTotalCancelado = Space(52)  
Retorno = EPSON_Obter_Total_Cancelado ( szTotalCancelado )
```

---

**Exemplo em Delphi 7:**

```
szTotalCancelado: array[0..52] of Char;  
Retorno := EPSON_Obter_Total_Cancelado ( szTotalCancelado );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalCancelado := AnsiString(StringOfChar(' ', 52));  
Retorno := EPSON_Obter_Total_Cancelado(PAnsiChar(szTotalCancelado));
```

**4.4.8 EPSON\_Obter\_Total\_Aliquotas**

Esta função lê o somatório do total vendido para nas alíquotas de ICMS, ISS e Não-Fiscais. Os valores informados, são relativos à jornada fiscal atual, e estes valores são reiniciados a cada nova jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Total_Aliquotas(pszTotalAliquotas:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalAliquotas	PCHAR	52	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-16	:	Total em ICMS
Byte 17-33	:	Total em ISS
Byte 34-50	:	Total Não-Fiscal

Todos os campos são de tamanho fixo e completados com zeros.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szTotalAliquotas[52];  
Retorno = EPSON_Obter_Total_Aliquotas ( szTotalAliquotas );
```

---

**Exemplo em C# :**

```
StringBuilder szTotalAliquotas = new StringBuilder(52, 52);  
Retorno = EPSON_Obter_Total_Aliquotas ( szTotalAliquotas );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalAliquotas As String  
szTotalAliquotas = Space(52)  
Retorno = EPSON_Obter_Total_Aliquotas ( szTotalAliquotas )
```

---

**Exemplo em Delphi 7:**

```
szTotalAliquotas: array[0..52] of Char;  
Retorno := EPSON_Obter_Total_Aliquotas ( szTotalAliquotas );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalAliquotas := AnsiString(StringOfChar(' ', 52));  
Retorno := EPSON_Obter_Total_Aliquotas(PAnsiChar(szTotalAliquotas));
```

#### 4.4.9 EPSON\_Obter\_Total\_Bruto

Esta função lê o total bruto vendido, com 17 dígitos. As casas da esquerda serão preenchidas com zeros. Os valores informados, são relativos à jornada fiscal atual, e estes valores são reiniciados a cada nova jornada fiscal.

##### Sintaxe:

```
function EPSON_Obter_Total_Bruto(pszTotalVendaBruta:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

##### Entradas:

Nenhum.

##### Saídas:

Variável	Tipo	Tam.	Descrição
pszTotalVendaBruta	PCHAR	19	Total bruto com 17 dígitos.

##### Retornos:

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

##### Requisitos:

Nenhum.

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



```
char szTotalVendaBruta[19];  
Retorno = EPSON_Obter_Total_Bruto ( szTotalVendaBruta );
```

##### Exemplo em C# :



```
StringBuilder szTotalVendaBruta = new StringBuilder(19, 19);  
Retorno = EPSON_Obter_Total_Bruto ( szTotalVendaBruta );
```

##### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Dim szTotalVendaBruta As String  
szTotalVendaBruta = Space(19)  
Retorno = EPSON_Obter_Total_Bruto ( szTotalVendaBruta )
```

##### Exemplo em Delphi 7:



```
szTotalVendaBruta: array[0..19] of Char;  
Retorno := EPSON_Obter_Total_Bruto ( szTotalVendaBruta );
```

##### Exemplo em Delphi 2010 / Delphi XE:



```
szTotalVendaBruta := AnsiString(StringOfChar(' ', 19));  
Retorno := EPSON_Obter_Total_Bruto(PAnsiChar(szTotalVendaBruta));
```

**4.4.10 EPSON\_Obter\_Total\_Descontos**

Esta função lê o total de descontos dados. Os valores informados, são relativos à jornada fiscal atual, e estes valores são reiniciados a cada nova jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Total_Descontos(pszTotalDescontos:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalDescontos	PCHAR	52	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-16 : Total de desconto ICMS

Byte 17-33 : Total de desconto ISS

Byte 34-50 : Total de desconto Não-Fiscal

Todos os campos são de tamanho fixo e completados com zeros.

**Retornos:**

FUNC\_SUCESSO Operação realizada com sucesso.

FUNC\_ERRO Erro durante a execução.

**Requisitos:**

Nenhum.

---

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

```
char szTotalDescontos[52];  
Retorno = EPSON_Obter_Total_Descontos ( szTotalDescontos );
```

---

**Exemplo em C# :**

```
StringBuilder szTotalDescontos = new StringBuilder(52, 52);  
Retorno = EPSON_Obter_Total_Descontos ( szTotalDescontos );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalDescontos As String  
szTotalDescontos = Space(52)  
Retorno = EPSON_Obter_Total_Descontos ( szTotalDescontos )
```

---

**Exemplo em Delphi 7:**

```
szTotalDescontos: array[0..52] of Char;  
Retorno := EPSON_Obter_Total_Descontos ( szTotalDescontos );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalDescontos := AnsiString(StringOfChar(' ', 52));  
Retorno := EPSON_Obter_Total_Descontos(PAnsiChar(szTotalDescontos));
```



**4.4.11 EPSON\_Obter\_Total\_Acrescimos**

Esta função lê o total de acréscimos dados. Os valores informados, são relativos à jornada fiscal atual, e estes valores são reiniciados a cada nova jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Total_Acrescimos(pszTotalAcrescimos:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalAcrescimos	PCHAR	52	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-16 : Total de acréscimo ICMS

Byte 17-33 : Total de acréscimo ISS

Byte 34-50 : Total de acréscimo Não-Fiscal

Todos os campos são de tamanho fixo e completados com zeros.

**Retornos:**

FUNC\_SUCESSO Operação realizada com sucesso.

FUNC\_ERRO Erro durante a execução.

**Requisitos:**

Nenhum.

---

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

```
char szTotalAcrescimos[52];  
Retorno = EPSON_Obter_Total_Acrescimos ( szTotalAcrescimos );
```

---

**Exemplo em C# :**

```
StringBuilder szTotalAcrescimos = new StringBuilder(52, 52);  
Retorno = EPSON_Obter_Total_Acrescimos ( szTotalAcrescimos );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalAcrescimos As String  
szTotalAcrescimos = Space(52)  
Retorno = EPSON_Obter_Total_Acrescimos ( szTotalAcrescimos )
```

---

**Exemplo em Delphi 7:**

```
szTotalAcrescimos: array[0..52] of Char;  
Retorno := EPSON_Obter_Total_Acrescimos ( szTotalAcrescimos );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalAcrescimos := AnsiString(StringOfChar(' ', 52));  
Retorno := EPSON_Obter_Total_Acrescimos(PAnsiChar(szTotalAcrescimos));
```

**4.4.12 EPSON\_Obter\_Total\_Troco**

Esta função lê o total de troco acumulado na jornada até o momento. Os valores informados, são relativos à jornada fiscal atual, e estes valores são reiniciados a cada nova jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Total_Troco(pszTotalTroco:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalTroco	PCHAR	14	Total do Troco acumulado na jornada atual.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char szTotalTroco[14];  
Retorno = EPSON_Obter_Total_Troco ( szTotalTroco );
```

**Exemplo em C# :**

```
StringBuilder szTotalTroco = new StringBuilder(14, 14);  
Retorno = EPSON_Obter_Total_Troco ( szTotalTroco );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalTroco As String  
szTotalTroco = Space(14)  
Retorno = EPSON_Obter_Total_Troco ( szTotalTroco )
```

**Exemplo em Delphi 7:**

```
szTotalTroco: array[0..14] of Char;  
Retorno := EPSON_Obter_Total_Troco ( szTotalTroco );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalTroco := AnsiString(StringOfChar(' ', 14));  
Retorno := EPSON_Obter_Total_Troco (PAnsiChar(szTotalTroco));
```

**4.4.13 EPSON\_Obter\_Venda\_Liquida\_ICMS**

Esta função lê o valor de venda líquida de ICMS acumulado na jornada até o momento.

**Sintaxe:**

```
function EPSON_Obter_Venda_Liquida_ICMS(pszTotalValorICMS:PChar):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalValorICMS	PCHAR	14	Valor da venda líquida de ICMS. Este campo é preenchido com zeros à esquerda.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char szTotalValorICMS[14];  
Retorno = EPSON_Obter_Venda_Liquida_ICMS( szTotalValorICMS );
```

**Exemplo em C# :**

```
StringBuilder szTotalValorICMS = new StringBuilder(14, 14);  
Retorno = EPSON_Obter_Venda_Liquida_ICMS( szTotalValorICMS );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalValorICMS As String  
szTotalValorICMS = Space(14)  
Retorno = EPSON_Obter_Venda_Liquida_ICMS( szTotalValorICMS )
```

**Exemplo em Delphi 7:**

```
szTotalValorICMS: array[0..14] of Char;  
Retorno := EPSON_Obter_Venda_Liquida_ICMS( szTotalValorICMS );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalValorICMS := AnsiString(StringOfChar(' ', 14));  
Retorno := EPSON_Obter_Venda_Liquida_ICMS(PAnsiChar(szTotalValorICMS));
```

**4.4.14 EPSON\_Obter\_Venda\_Liquida\_ISSQN**

Esta função lê o valor de venda líquida de ISSQN acumulado na jornada até o momento.

**Sintaxe:**

```
function EPSON_Obter_Venda_Liquida_ISSQN(pszTotalValorISSQN:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalValorISSQN	PCHAR	14	Valor da venda líquida de ISSQN. Este campo é preenchido com zeros à esquerda.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szTotalValorISSQN[14];  
Retorno = EPSON_Obter_Venda_Liquida_ISSQN( szTotalValorISSQN );
```

---

**Exemplo em C# :**

```
StringBuilder szTotalValorISSQN = new StringBuilder(14, 14);  
Retorno = EPSON_Obter_Venda_Liquida_ISSQN( szTotalValorISSQN );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalValorISSQN As String  
szTotalValorISSQN = Space(14)  
Retorno = EPSON_Obter_Venda_Liquida_ISSQN( szTotalValorISSQN )
```

---

**Exemplo em Delphi 7:**

```
szTotalValorISSQN: array[0..14] of Char;  
Retorno := EPSON_Obter_Venda_Liquida_ISSQN( szTotalValorISSQN );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalValorISSQN := AnsiString(StringOfChar(' ', 14));  
Retorno := EPSON_Obter_Venda_Liquida_ISSQN(PAnsiChar(szTotalValorISSQN));
```

**4.4.15 EPSON\_Obter\_Total\_ICMS**

Esta função lê o valor total de ICMS acumulado na jornada até o momento.

**Sintaxe:**

```
function EPSON_Obter_Total_ICMS(pszTotalICMS:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalICMS	PCHAR	14	Valor total de ICMS. Este campo é preenchido com zeros à esquerda.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char szTotalICMS [14];  
Retorno = EPSON_Obter_Total_ICMS( szTotalICMS );
```

**Exemplo em C# :**

```
StringBuilder szTotalICMS = new StringBuilder(14, 14);  
Retorno = EPSON_Obter_Total_ICMS( szTotalICMS );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalICMS As String  
szTotalICMS = Space(14)  
Retorno = EPSON_Obter_Total_ICMS( szTotalICMS )
```

**Exemplo em Delphi 7:**

```
szTotalICMS: array[0..14] of Char;  
Retorno := EPSON_Obter_Total_ICMS( szTotalICMS );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalICMS: array[0..14] of Char;  
Retorno := EPSON_Obter_Total_ICMS( szTotalICMS );
```

**4.4.16 EPSON\_Obter\_Total\_ISSQN**

Esta função lê o valor total de ISSQN acumulado na jornada até o momento.

**Sintaxe:**

```
function EPSON_Obter_Total_ISSQN(pszTotalISSQN:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalISSQN	PCHAR	14	Valor total de ISSQN. Este campo é preenchido com zeros à esquerda.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char szTotalISSQN[14];  
Retorno = EPSON_Obter_Total_ISSQN( szTotalISSQN );
```

**Exemplo em C# :**

```
StringBuilder szTotalISSQN = new StringBuilder(14, 14);  
Retorno = EPSON_Obter_Total_ISSQN( szTotalISSQN );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalISSQN As String  
szTotalISSQN = Space(14)  
Retorno = EPSON_Obter_Total_ISSQN( szTotalISSQN )
```

**Exemplo em Delphi 7:**

```
szTotalISSQN: array[0..14] of Char;  
Retorno := EPSON_Obter_Total_ISSQN( szTotalISSQN );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalISSQN := AnsiString(StringOfChar(' ', 14));  
Retorno := EPSON_Obter_Total_ISSQN(PAnsiChar(szTotalISSQN));
```



**4.4.17 EPSON\_Obter\_Dados\_Impressora**

Esta função retorna os dados de serialização da impressora.

**Sintaxe:**

```
function EPSON_Obter_Dados_Impressora(pszDadosImpressora:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDadosImpressora	PCHAR	109	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-19	:	Número de série da memória fiscal
Byte 20-39	:	Número de série da MFD
Byte 40-59	:	Fabricante do ECF
Byte 60-79	:	Modelo do ECF
Byte 80-99	:	Tipo do ECF
Byte 100-107	:	Versão do Software Básico

Todos os campos têm tamanho fixo.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szDadosImpressora[109];  
Retorno = EPSON_Obter_Dados_Impressora ( szDadosImpressora );
```

---

**Exemplo em C# :**

```
StringBuilder szDadosImpressora = new StringBuilder(109, 109);  
Retorno = EPSON_Obter_Dados_Impressora ( szDadosImpressora );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDadosImpressora As String  
szDadosImpressora = Space(109)  
Retorno = EPSON_Obter_Dados_Impressora ( szDadosImpressora )
```

---

**Exemplo em Delphi 7:**

```
szDadosImpressora: array[0..109] of Char;  
Retorno := EPSON_Obter_Dados_Impressora ( szDadosImpressora );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDadosImpressora := AnsiString(StringOfChar(' ', 109));  
Retorno := EPSON_Obter_Dados_Impressora(PAnsiChar(szDadosImpressora));
```

**4.4.18 EPSON\_Obter\_Cliche\_Usuario**

Esta função lê as informações do contribuinte cadastrado na impressora.

**Sintaxe:**

```
function EPSON_Obter_Cliche_Usuario(pszDadosUsuario:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDadosUsuario	PCHAR	161	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-39	:	Razão Social
Byte 40-79	:	Nome Fantasia
Byte 80-119	:	Endereço Linha1
Byte 120-159	:	Endereço Linha2

Todos os campos, quando preenchidos, têm tamanho fixo.

**Retornos:**

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

**Requisitos:**

Documentos fechados.

---

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

```
char szDadosUsuario[161];  
Retorno = EPSON_Obter_Cliche_Usuario ( szDadosUsuario );
```

---

**Exemplo em C# :**

```
StringBuilder szDadosUsuario = new StringBuilder(161, 161);  
Retorno = EPSON_Obter_Cliche_Usuario ( szDadosUsuario );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDadosUsuario As String  
szDadosUsuario = Space(161)  
Retorno = EPSON_Obter_Cliche_Usuario ( szDadosUsuario )
```

---

**Exemplo em Delphi 7:**

```
szDadosUsuario: array[0..161] of Char;  
Retorno := EPSON_Obter_Cliche_Usuario ( szDadosUsuario );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDadosUsuario := AnsiString(StringOfChar(' ', 160));  
Retorno := EPSON_Obter_Cliche_Usuario(PAnsiChar(szDadosUsuario));
```

**4.4.19 EPSON\_Obter\_Cliche\_UsuarioEX**

Esta função lê as informações do contribuinte cadastrado na impressora.

**Sintaxe:**

```
function EPSON_Obter_Cliche_UsuarioEX(pszDadosUsuario:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDadosUsuario	PCHAR	225	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-55	:	Razão Social
Byte 56-111	:	Nome Fantasia (Opcional)
Byte 112-167	:	Endereço Linha1
Byte 168-223	:	Endereço Linha2 (Opcional)

Todos os campos, quando preenchidos, têm tamanho fixo.

**Retornos:**

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

**Requisitos:**

Documentos fechados.

---

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

```
char szDadosUsuario[225];  
Retorno = EPSON_Obter_Cliche_UsuarioEX( szDadosUsuario );
```

---

**Exemplo em C# :**

```
StringBuilder szDadosUsuario = new StringBuilder(225, 225);  
Retorno = EPSON_Obter_Cliche_UsuarioEX( szDadosUsuario );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDadosUsuario As String  
szDadosUsuario = Space(225)  
Retorno = EPSON_Obter_Cliche_UsuarioEX( szDadosUsuario )
```

---

**Exemplo em Delphi 7:**

```
szDadosUsuario: array[0..225] of Char;  
Retorno := EPSON_Obter_Cliche_UsuarioEX( szDadosUsuario );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDadosUsuario := AnsiString(StringOfChar(' ', 225));  
Retorno := EPSON_Obter_Cliche_UsuarioEX(PAnsiChar(szDadosUsuario));
```

**4.4.20 EPSON\_Obter\_Data\_Hora\_Jornada**

Este comando lê a data e a hora da abertura fiscal da impressora.

**Sintaxe:**

```
function EPSON_Obter_Data_Hora_Jornada(pszDataHora:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDataHora	PCHAR	15	Dados recebidos da impressora.(ddmmaaaa)

Os dados serão retornados no seguinte formato:

Byte 0-7 : Data de abertura

Byte 8-13 : Hora de abertura

Todos os campos têm tamanho fixo.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szDataHora[15];  
Retorno = EPSON_Obter_Data_Hora_Jornada ( szDataHora );
```

---

**Exemplo em C# :**

```
StringBuilder szDataHora = new StringBuilder(15, 15);  
Retorno = EPSON_Obter_Data_Hora_Jornada ( szDataHora );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDataHora As String  
szDataHora = Space(15)  
Retorno = EPSON_Obter_Data_Hora_Jornada ( szDataHora )
```

---

**Exemplo em Delphi 7:**

```
szDataHora: array[0..15] of Char;  
Retorno := EPSON_Obter_Data_Hora_Jornada ( szDataHora );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDataHora: array[0..15] of Char;  
Retorno := EPSON_Obter_Data_Hora_Jornada ( szDataHora );
```



**4.4.21 EPSON\_Obter\_Numero\_ECF\_Loja**

Esta função lê o número da impressora e da loja.

**Sintaxe:**

```
function EPSON_Obter_Numero_ECF_Loja(pszECFLoja:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszECFLoja	PCHAR	8	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-2 : Número da Impressora

Byte 3-6 : Número da Loja

Todos os campos têm tamanho fixo.

**Retornos:**

FUNC\_SUCESSO Operação realizada com sucesso.

FUNC\_ERRO Erro durante a execução.

**Requisitos:**

Nenhum.

---

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

```
char szNumeroECFLoja[8];  
Retorno = EPSON_Obter_Numero_ECF_Loja ( szNumeroECFLoja );
```

---

**Exemplo em C#:**

```
StringBuilder szNumeroECFLoja = new StringBuilder(8, 8);  
Retorno = EPSON_Obter_Numero_ECF_Loja ( szNumeroECFLoja );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szNumeroECFLoja As String  
szNumeroECFLoja = Space(8)  
Retorno = EPSON_Obter_Numero_ECF_Loja ( szNumeroECFLoja )
```

---

**Exemplo em Delphi 7:**

```
szNumeroECFLoja: array[0..8] of Char;  
Retorno := EPSON_Obter_Numero_ECF_Loja ( szNumeroECFLoja );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szNumeroECFLoja := AnsiString(StringOfChar(' ', 8));  
Retorno := EPSON_Obter_Numero_ECF_Loja(PAnsiChar(szNumeroECFLoja));
```

**4.4.22 EPSON\_Obter\_Hora\_Relogio**

Esta função lê a data e a hora atuais da impressora.

**Sintaxe:**

```
function EPSON_Obter_Hora_Relogio(pszDataHoraRelogio:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDataHoraRelogio	PCHAR	15	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-7 : Data da impressora

Byte 8-13 : Hora da impressora

Todos os campos têm tamanho fixo.

**Retornos:**

FUNC\_SUCESSO Operação realizada com sucesso.

FUNC\_ERRO Erro durante a execução.

**Requisitos:**

Nenhum.

---

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

```
char szDataHoraRelogio[15];  
Retorno = EPSON_Obter_Hora_Relogio ( szDataHoraRelogio );
```

---

**Exemplo em C# :**

```
StringBuilder szDataHoraRelogio = new StringBuilder(15, 15);  
Retorno = EPSON_Obter_Hora_Relogio ( szDataHoraRelogio );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDataHoraRelogio As String  
szDataHoraRelogio = Space(15)  
Retorno = EPSON_Obter_Hora_Relogio ( szDataHoraRelogio )
```

---

**Exemplo em Delphi 7:**

```
szDataHoraRelogio: array[0..15] of Char;  
Retorno := EPSON_Obter_Hora_Relogio ( szDataHoraRelogio );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDataHoraRelogio := AnsiString(StringOfChar(' ', 15));  
Retorno := EPSON_Obter_Hora_Relogio(PAnsiChar(szDataHoraRelogio));
```

**4.4.23 EPSON\_Obter\_Simbolo\_Moeda**

Esta função retorna o símbolo da moeda utilizado no momento.

**Sintaxe:**

```
function EPSON_Obter_Simbolo_Moeda(pszSimboloMoeda:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszSimboloMoeda	PCHAR	5	Dados recebidos da impressora.


**Retornos:**

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


**Requisitos:**

Nenhum.


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

```
 char szSimboloMoeda[5];  
Retorno = EPSON_Obter_Simbolo_Moeda ( szSimboloMoeda );
```


**Exemplo em C# :**

```
 StringBuilder szSimboloMoeda = new StringBuilder(5, 5);  
Retorno = EPSON_Obter_Simbolo_Moeda ( szSimboloMoeda );
```


**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
 Dim szSimboloMoeda As String  
szSimboloMoeda = Space(5)  
Retorno = EPSON_Obter_Simbolo_Moeda ( szSimboloMoeda )
```

**Exemplo em Delphi 7:**

```
 szSimboloMoeda: array[0..5] of Char;  
Retorno := EPSON_Obter_Simbolo_Moeda ( szSimboloMoeda );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
 szSimboloMoeda := AnsiString(StringOfChar(' ', 5));  
Retorno := EPSON_Obter_Simbolo_Moeda(PAnsiChar(szSimboloMoeda));
```

**4.4.24 EPSON\_Obter\_Casas\_Decimais**

Esta função retorna o número de casas decimais usadas para os campos quantidade e preço.

**Sintaxe:**

```
function EPSON_Obter_Casas_Decimais(pszCasasDecimais:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszCasasDecimais	PCHAR	7	Dados recebidos da impressora

Os dados serão retornados no seguinte formato:

Byte 0-2	:	Número de casas decimais para a quantidade.
Byte 3-5	:	Número de casas decimais para o preço.

Todos os campos são de tamanho fixo e completados com zeros.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szCasasDecimais[7];  
Retorno = EPSON_Obter_Casas_Decimais ( szCasasDecimais );
```

---

**Exemplo em C# :**

```
StringBuilder szCasasDecimais = new StringBuilder(7, 7);  
Retorno = EPSON_Obter_Casas_Decimais ( szCasasDecimais );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCasasDecimais As String  
szCasasDecimais = Space(7)  
Retorno = EPSON_Obter_Casas_Decimais ( szCasasDecimais )
```

---

**Exemplo em Delphi 7:**

```
szCasasDecimais: array[0..7] of Char;  
Retorno := EPSON_Obter_Casas_Decimais ( szCasasDecimais );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCasasDecimais := AnsiString(StringOfChar(' ', 7));  
Retorno := EPSON_Obter_Casas_Decimais(PAnsiChar(szCasasDecimais));
```

**4.4.25 EPSON\_Obter\_Desconto\_Iss**

Esta função retorna se é possível ou não efetuar a operação de desconto sobre ISSQN.

**Sintaxe:**

```
function EPSON_Obter_Desconto_Iss(pszDescontoISS:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDescontoISS	PCHAR	2	Indicação da possibilidade de efetuar o desconto. <b>S</b> - desconto ISSQN habilitado. <b>N</b> - desconto ISSQN desabilitado.

**Retornos:**

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

**Requisitos:**

Nenhum.



---

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

```
char szDescontoIss[2];  
Retorno = EPSON_Obter_Desconto_Iss (  szDescontoIss  );
```

---

**Exemplo em C# :**

```
StringBuilder szDescontoIss = new StringBuilder(2, 2);  
Retorno =  EPSON_Obter_Desconto_Iss (  szDescontoIss  );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDescontoIss As String  
szDescontoIss = Space(2)  
Retorno =  EPSON_Obter_Desconto_ISS (  szDescontoIss  )
```

---

**Exemplo em Delphi 7:**

```
szDescontoIss: array[0..2] of Char;  
Retorno :=  EPSON_Obter_Desconto_Iss (  szDescontoIss  );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDescontoIss := AnsiString(StringOfChar(' ', 2));  
Retorno := EPSON_Obter_Desconto_Iss(PAnsiChar(szDescontoIss));
```

**4.4.26 EPSON\_Obter\_Contadores**

Esta função lê os contadores da impressora.

**Sintaxe:**

```
function EPSON_Obter_Contadores(pszContadores:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszContadores	PCHAR	85	Dados recebidos da impressora

**COO CRZ CRO GNF CDC NFC GRG CCF CFC CFD DFI NCN UCF CLX**

Os dados serão retornados no seguinte formato:

Byte 0-5	:	<b>COO</b>	- Contador de Ordem de Operação
Byte 6-11	:	<b>CRZ</b>	- Contador de Redução Z
Byte 12-17	:	<b>CRO</b>	- Contador de Reinício de Operação
Byte 18-23	:	<b>GNF</b>	- Contador Geral Não Fiscal
Byte 24-29	:	<b>CDC</b>	- Contador de Cupom de Débito/Crédito
Byte 30-35	:	<b>NFC</b>	- Contador de Cupom Não-Fiscal
Byte 36-41	:	<b>GRG</b>	- Contador Geral de Relatório Gerencial
Byte 42-47	:	<b>CCF</b>	- Contador de Cupom Fiscal
Byte 48-53	:	<b>CFC</b>	- Contador de Cupom Fiscal Cancelado
Byte 54-59	:	<b>CFD</b>	- Contador de Fita Detalhe
Byte 60-65	:	<b>DFI</b>	- Contador de Documentos Fiscais
Byte 66-71	:	<b>NCN</b>	- Número de Comprovantes de Crédito/Débito não emitidos
Byte 72-77	:	<b>UCF</b>	- COO do Último Cupom Fiscal
Byte 78-83	:	<b>CLX</b>	- Contador de Leitura X

Todos os campos são de tamanho fixo e completados com zeros.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char szContadores[85];  
Retorno = EPSON_Obter_Contadores ( szContadores );
```

**Exemplo em C# :**

```
StringBuilder szContadores = new StringBuilder(85, 85);  
Retorno = EPSON_Obter_Contadores ( szContadores );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szContadores As String  
szContadores = Space(85)  
Retorno = EPSON_Obter_Contadores ( szContadores )
```

**Exemplo em Delphi 7:**

```
szContadores: array[0..85] of Char;  
Retorno := EPSON_Obter_Contadores ( szContadores );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szContadores := AnsiString(StringOfChar(' ', 85));  
Retorno := EPSON_Obter_Contadores(PAnsiChar(szContadores));
```

**4.4.27 EPSON\_Obter\_Estado\_ImpressoraEX**

Esta função lê o estado atual da impressora.

**Sintaxe:**

```
function EPSON_Obter_Estado_ImpressoraEX( szEstadoImpressora:PChar; szEstadoFiscal:PChar;  
                                           szRetornoComando:PChar; szMsgErro:PChar):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szEstadoImpressora	PCHAR	17	Estado do mecanismo impressor – <a href="#">vide tabela 4.5.27.1</a>
szEstadoFiscal	PCHAR	17	Estado fiscal – <a href="#">vide tabela 4.5.27.2</a>
szRetornoComando	PCHAR	5	Retorno do comando – <a href="#">vide tabela 4.5.27.3</a>
szMsgErro	PCHAR	101	Mensagem de retorno do comando – <a href="#">vide tabela 4.5.27.3</a>

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
Char szEstadoImpressora [17];  
Char szEstadoFiscal [17];  
Char szRetornoComando [5];  
Char szMsgErro [101];  
Retorno = EPSON_Obter_Estado_ImpressoraEX(szEstadoImpressora,  
                                           szEstadoFiscal, szRetornoComando);
```

**Exemplo em C# :**

```
StringBuilder szEstadoImpressora = new StringBuilder(17, 17);  
StringBuilder szEstadoFiscal = new StringBuilder(17, 17);  
StringBuilder szRetornoComando = new StringBuilder(5, 5);  
StringBuilder szMsgErro = new StringBuilder(101, 101);  
Retorno = EPSON_Obter_Estado_ImpressoraEX(szEstadoImpressora,  
                                           szEstadoFiscal, szRetornoComando, szMsgErro);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szEstadoImpressora As String
Dim szEstadoFiscalAs String
Dim szRetornoComando As String
Dim szMsgErro As String

szEstadoImpressora = Space(17)
szEstadoFiscal = Space(17)
szRetornoComando = Space(5)
szMsgErro = Space(101)
Retorno = EPSON_Obter_Estado_ImpressoraEX(szEstadoImpressora,
szEstadoFiscal, szRetornoComando, szMsgErro);
```

**Exemplo em Delphi 7:**

```
SzEstadoImpressora: array[0..17] of Char;
szEstadoFiscalAs: array[0..17] of Char;
szRetornoComando: array[0..5] of Char;
szMsgErro: array[0..101] of Char;
Retorno := EPSON_Obter_Estado_ImpressoraEX ( szEstadoImpressora,
szEstadoFiscal, szRetornoComando, szMsgErro);
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
SzEstadoImpressora: AnsiString;
szEstadoFiscalAs: AnsiString;
szRetornoComando: AnsiString;
szMsgErro: AnsiString;

SzEstadoImpressora: AnsiString(StringOfChar(' ', 17);
szEstadoFiscalAs: AnsiString(StringOfChar(' ', 17);
szRetornoComando: AnsiString(StringOfChar(' ', 5);
szMsgErro: AnsiString(StringOfChar(' ', 101);

Retorno := EPSON_Obter_Estado_ImpressoraEX (
PAnsiChar(szEstadoImpressora), PAnsiChar(szEstadoFiscal),
PAnsiChar(szRetornoComando), PAnsiChar(szMsgErro));
```

**4.4.27.1 Estado da Impressora:**

Posição	Valores permitidos
1	Impressora Offline
2	Erro de impressão
3	Tampa superior aberta
4	Estado da gaveta
5	Reservado
6 e 7	00 = Estação recibo 01 = Estação cheque 10 = Estação Autenticação (TMH6000 e TMU675) 11 = Leitura do MICR
8	Aguardando retirada do papel

9	Aguardando inserção do papel
10	Estado do sensor inferior da estação de cheque
11	Estado do sensor superior da estação do cheque
12	Estado do sensor de autenticação (TMH6000 e TMU675)
13 e 14	Reservado
15	Sem papel
16	Pouco papel

## 4.4.27.2 Estado fiscal:

Posição	Valores permitidos
1 e 2	00 = Modo bloqueado 01 = Reservado 10 = Modo manufatura (Não-Fiscalizado) 11 = Modo Fiscalizado
3	Reservado
4	0 = Modo de operação normal 1 = Modo de Intervenção Técnica
5 e 6	00 = Memória Fiscal em operação normal 01 = Memória Fiscal em esgotamento 10 = Memória Fiscal cheia 11 = Erro de leitura/escrita da Memória Fiscal
7 e 8	Reservado
9	0 = Período de vendas fechado 1 = Período de vendas aberto
10 a 12	Reservado
13 a 16	0000 = Documento fechado 0001 = Cupom Fiscal aberto 0010 = Comprovante de Crédito ou Débito 0011 = Estorno de Comprovante de Crédito ou Débito 0100 = Relatório Gerencial 1000 = Comprovante Não-Fiscal 1001 = Cheque ou autenticação (TMH6000 e TMU675)

## 4.4.27.3 Estado de execução do último comando:

Código do erro (Hexa)	Descrição do Erro
Retornos Essenciais (00)	
0000	"Resultado sem erro"
0001	"Erro interno"

0002	"Erro de iniciação do equipamento"
0003	"Erro de processo interno"
Retornos sobre Comandos Genéricos (01)	
0101	"Comando inválido para o estado atual"
0102	"Comando inválido para o documento atual"
0106	"Comando aceito apenas fora de intervenção"
0107	"Comando aceito apenas dentro de intervenção"
0108	"Comando inválido durante processo de scan"
0109	"Excesso de intervenções"
Retornos sobre Campos de Protocolo (02)	
0201	"Comando com frame inválido"
0202	"Comando inválido"
0203	"Campos em excesso"
0204	"Campos em falta"
0205	"Campo não opcional"
0206	"Campo alfanumérico inválido"
0207	"Campo alfabético inválido"
0208	"Campo numérico inválido"
0209	"Campo binário inválido"
020A	"Campo imprimível inválido"
020B	"Campo hexadecimal inválido"
020C	"Campo data inválido"
020D	"Campo hora inválido"
020E	"Campo com atributos de impressão inválidos"
020F	"Campo booleano inválido"
0210	"Campo com tamanho inválido"
0211	"Extensão de comando inválida"
0212	"Código de barra não permitido"
0213	"Atributos de impressão não permitidos"
0214	"Atributos de impressão inválidos"
0215	"Código de barras incorretamente definido"
0217	"Comando inválido para a porta selecionada"
Retornos sobre Problemas de Hardware (03)	
0301	"Erro de hardware"
0302	"Impressora não está pronta"
0303	"Erro de Impressão"
0304	"Falta de papel"
0305	"Pouco papel disponível"
0306	"Erro em carga ou expulsão do papel"
0307	"Característica não suportada pela impressora"
0308	"Erro de display"
0309	"Sequência de scan inválida"
030A	"Número de área de recorte inválido"
030B	"Scanner não preparado"
030C	"Qualidade de Logotipo não suportada pela impressora"
030E	"Erro de leitura do microcódigo"
Retornos de Iniciação (04)	
0401	"Número de série inválido"
0402	"Requer dados de fiscalização já configurados"
Retornos de Configuração (05)	
0501	"Data / Hora não configurada"
0502	"Data inválida"
0503	"Data em intervalo inválido"
0504	"Nome operador inválido"
0505	"Número de caixa inválido"
0508	"Dados de Cabeçalho ou rodapé inválidos"
0509	"Excesso de fiscalização"
050C	"Número máximo de meios de pagamento já definidos"
050D	"Meio de pagamento já definido"
050E	"Meio de pagamento inválido"
050F	"Descrição do meio de pagamento inválido"
0510	"Valor máximo de desconto inválido"
0513	"Logotipo do usuário inválido"
0514	"Sequência de logotipo inválido"
0515	"Configuração de display inválida"
0516	"Dados do MICR inválidos"
0517	"Campo de endereço inválido"
0518	"Nome da loja não definido"
0519	"Dados fiscais não definidos"
051A	"Número sequencial do ECF inválido"
051B	"Simbologia do GT inválida, devem ser todos diferentes"
051C	"Número de CNPJ inválido"

051D	"Senha de fiscalização inválida"
051E	"Último documento deve ser uma redução Z"
051F	"Símbolo da moeda igual ao atualmente cadastrado"
0520	"Identificação da alíquota não cadastrada"
0521	"Alíquota não cadastrada"
Retornos sobre Memória de Fita-detelhe(06)	
0601	"Memória de Fita-detelhe esgotada"
0605	"Número de série inválido para a Memória de Fita-detelhe"
0606	"Memória de Fita-detelhe não iniciada"
0607	"Memória de Fita-detelhe não pode estar iniciada"
0608	"Número de série da Memória de Fita-detelhe não confere"
0609	"Erro Interno na Memória de Fita-detelhe"
Retornos sobre Jornada Fiscal (07)	
0701	"Valor inválido para o número do registro"
0702	"Valor inválido para o número do item"
0703	"Intervalo inválido para a leitura da MFD"
0704	"Número de usuário inválido para MFD"
Retornos sobre Jornada Fiscal (08)	
0801	"Comando inválido com jornada fiscal fechada"
0802	"Comando inválido com jornada fiscal aberta"
0803	"Memória Fiscal esgotada"
0804	"Jornada fiscal deve ser fechada"
0805	"Não há meios de pagamento definidos"
0806	"Excesso de meios de pagamento utilizados na jornada fiscal"
0807	"Jornada fiscal sem movimento de vendas"
0808	"Intervalo de jornada fiscal inválido"
0809	"Existem mais dados para serem lidos"
080A	"Não existem mais dados para serem lidos"
080B	"Não pode abrir jornada fiscal"
080C	"Não pode fechar jornada fiscal"
080D	"Limite máximo do período fiscal atingido"
080E	"Limite máximo do período fiscal não atingido"
080F	"Abertura da jornada fiscal não permitida"
Retornos sobre Transações Genéricas (09)	
0901	"Valor muito grande"
0902	"Valor muito pequeno"
0903	"Itens em excesso"
0904	"Alíquotas em excesso"
0905	"Desconto ou acréscimos em excesso"
0906	"Meios de pagamento em excesso"
0907	"Item não encontrado"
0908	"Meio de pagamento não encontrado"
0909	"Total nulo"
090C	"Tipo de pagamento não definido"
090F	"Alíquota não encontrada"
0910	"Alíquota inválida"
0911	"Excesso de meios de pagamento com CDC"
0912	"Meio de pagamento com CDC já emitido"
0913	"Meio de pagamento com CDC ainda não emitido"
0914	"Leitura da Memória Fiscal – intervalo CRZ inválido"
0915	"Leitura da Memória Fiscal – intervalo de data inválido"
Retornos sobre Cupom Fiscal (0A)	
0A01	"Operação não permitida após desconto / acréscimo"
0A02	"Operação não permitida após registro de pagamentos"
0A03	"Tipo de item inválido"
0A04	"Linha de descrição em branco"
0A05	"Quantidade muito pequena"
0A06	"Quantidade muito grande"
0A07	"Total do item com valor muito alto"
0A08	"Operação não permitida antes do registro de pagamentos"
0A09	"Registro de pagamento incompleto"
0A0A	"Registro de pagamento finalizado"
0A0B	"Valor pago inválido"
0A0C	"Valor de desconto ou acréscimo não permitido"
0A0E	"Valor não pode ser zero"
0A0F	"Operação não permitida antes do registro de itens"
0A11	"Cancelamento de desconto e acréscimo somente para item atual"
0A12	"Não foi possível cancelar último Cupom Fiscal"
0A13	"Último Cupom Fiscal não encontrado"
0A14	"Último Comprovante Não-Fiscal não encontrado"
0A15	"Cancelamento de CDC necessária"
0A16	"Número de item em Cupom Fiscal inválido"



0A17	"Operação somente permitida após subtotalização"
0A18	"Operação somente permitida durante a venda de itens"
0A19	"Operação não permitida em item com desconto ou acréscimo"
0A1A	"Dígitos de quantidade inválidos"
0A1B	"Dígitos de valor unitário inválido"
0A1C	"Não há desconto ou acréscimo a cancelar"
0A1D	"Não há item para cancelar"
0A1E	"Desconto ou acréscimo somente no item atual"
0A1F	"Desconto ou acréscimo já efetuado"
0A20	"Desconto ou acréscimo nulo não permitido"
0A21	"Valor unitário inválido"
0A22	"Quantidade inválida"
0A23	"Código de item inválido"
0A24	"Descrição inválida"
0A25	"Operação de desconto ou acréscimo não permitida"
0A26	"Mensagem promocional já impressa"
0A27	"Linhas adicionais não podem ser impressas"
0A28	"Dados do consumidor já impresso"
0A29	"Dados do consumidor somente no fim do documento"
0A2A	"Dados do consumidor somente no início do documento"
0A2B	"Comando Inválido para o item"
Erros em operações não-fiscais (0E)	
0E01	"Número de linhas em documento excedido"
0E02	"Número do relatório inválido"
0E03	"Operação não permitida após registro de itens"
0E04	"Registro de valor nulo não permitido"
0E05	"Não há desconto a cancelar"
0E06	"Não há acréscimo a cancelar"
0E07	"Operação somente permitida após subtotalização"
0E08	"Operação somente permitida durante registro de itens"
0E09	"Operação não-fiscal inválida"
0E0A	"Último comprovante Não-Fiscal não encontrado"
0E0B	"Meio de pagamento não encontrado"
0E0C	"Não foi possível imprimir nova via"
0E0D	"Não foi possível realizar reimpressão"
0E0E	"Não foi possível imprimir nova parcela"
0E0F	"Não há mais parcelas a imprimir"
0E10	"Registro de item Não-Fiscal inválido"
0E11	"Desconto ou acréscimo já efetuado"
0E12	"Valor de desconto ou acréscimo inválido"
0E13	"Não foi possível cancelar o item"
0E14	"Itens em excesso"
0E15	"Operação Não-Fiscal não cadastrada"
0E16	"Excesso de relatórios / operações não-fiscais cadastradas"
0E17	"Relatório não encontrado"
0E18	"Comando não permitido"
0E19	"Comando não permitido em operações não-fiscais para movimento de monetário"
0E1A	"Comando permitido apenas em operações não-fiscais para movimento de monetário"
0E1B	"Número de parcelas inválido para a emissão de CCD"
0E1C	"Operação não fiscal já cadastrada"
0E1D	"Relatório gerencial já cadastrado"
0E1E	"Relatório Gerencial Inválido"
Erros para impressão de cheque ou autenticação (30)	
3001	"Configuração de cheque não registrada"
3002	"Configuração de cheque não encontrada"
3003	"Valor do cheque já impresso"
3004	"Nominal ao cheque já impresso"
3005	"Linhas adicionais no cheque já impresso"
3006	"Autenticação já impressa"
3007	"Número máximo de autenticações já impresso"
3008	"Aguardando impressão de folha solta"
3009	"Documento não foi inserido"
Outros (FF)	
FFFF	"Erro desconhecido"

**4.4.28 EPSON\_Obter\_Estado\_Impressora**

Esta função lê o estado atual da impressora.

**Atenção:** É recomendado o uso da função EPSON\_Obter\_Estado\_ImpressoraEX (acima) em substituição a esta função.

**Sintaxe:**

```
function EPSON_Obter_Estado_Impressora(pszEstadoImpressora:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszEstadoImpressora	PCHAR	21	Dados recebidos da impressora

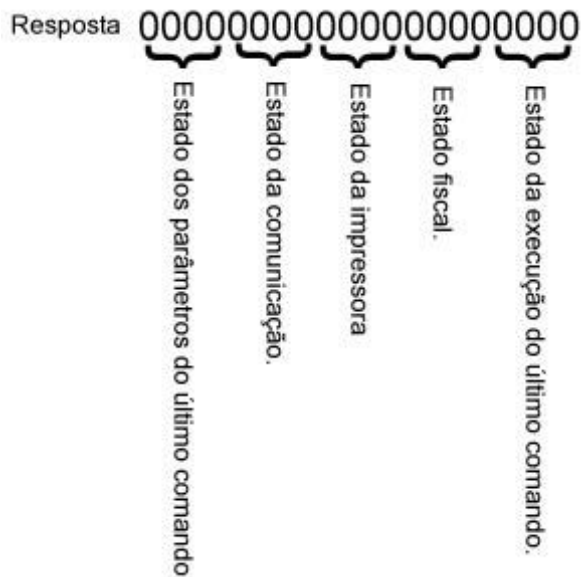
Os dados serão retornados no seguinte formato:

Faixa de Bytes	Descrição
Byte 0-3	Estado dos parâmetros do último comando. <a href="#">Vide tabela 4.5.28.1</a>
Byte 4-7	Estado da comunicação. <a href="#">Vide tabela 4.5.28.3</a>
Byte 8-11	Estado da impressora <a href="#">Vide tabela 4.5.28.4</a>
Byte 12-15	Estado fiscal. <a href="#">Vide tabela 4.5.28.5</a>
Byte 16-19	Estado da execução do último comando. <a href="#">Vide tabela 4.5.28.6</a>

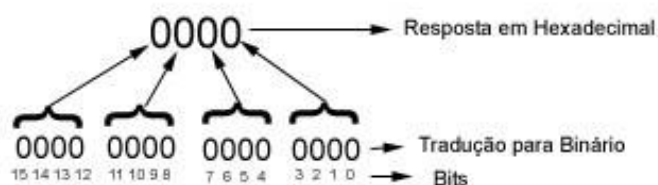
Todos os campos são de tamanho fixo e completados com zeros.

**Obs.:** Veja nas próximas páginas, a explicação de como interpretar cada resposta e suas respectivas tabelas.

## Entendendo as Repostas:



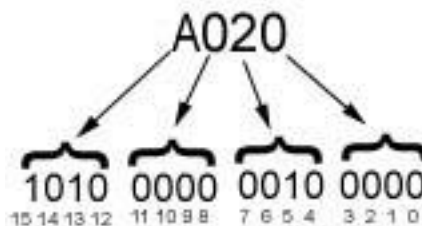
Como podemos observar na figura acima, a resposta é dividida em 5 grupos, analisado um único grupo, verificamos que ele é composto por 4 hexadecimais sendo que cada hexa é composto por 4 bits, totalizando 16 bits, veja a figura abaixo:



**Obs:** Apenas em alguns comandos será necessário transformar o valor hexadecimal em binário e a partir do binário interpretar a resposta por meio dos bits na tabela do respectivo comando.

Tabela de transformação Hexa para Binário:

Binário				Hexadecimal
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F



No exemplo acima, podemos observar que os bits 15, 13 e 5 estão com valor igual a 1 e o restante assumem zero. Com estes dados, basta ir à tabela do respectivo grupo, e interpretar o que está ocorrendo. Exemplo: vamos supor que a resposta acima pertença ao grupo Estado da Impressora, a tradução dele seria: Impressora Offline; Tampa superior aberta e sensor superior da estação do cheque ativo.

## 4.4.28.1 Bytes de 0 a 1 – Parâmetros do último comando:

Erros da Interface	Resposta	Descrição
EPSON_FIELD_STRING_ERR	0x01	Campo deve ser um ponteiro para string
EPSON_FIELD_DATE_ERR	0x02	String com data inválida
EPSON_FIELD_TIME_ERR	0x03	String com hora inválida
EPSON_FIELD_OPT_DATE_ERR	0x04	String não está vazia ou contém data inválida
EPSON_FIELD_OPT_TIME_ERR	0x05	String não está vazia ou contém hora inválida
EPSON_FIELD_STR_NOEMPTY_ERR	0x06	String não pode ser vazia
EPSON_FIELD_BUFFER_ERR	0x07	Range inválido.
EPSON_FIELD_FILENAME_ERR	0x08	Nome de Arquivo Invalido
EPSON_FIELD_NULL_POINTER	0x09	Ponteiro Nulo
EPSON_FIELD_STRING_OVER_LIMIT	0x0A	String maior que o limite
EPSON_FIELD_MFMD_FFILE_ERR	0x0B	Arquivos da MF e MFD não consistem entre si
EPSON_FIELD_MF_FILE_ERR	0x0C	Arquivo da MF inválido
EPSON_FIELD_MFD_FILE_ERR	0x0D	Arquivo da MFD inválido

## 4.4.28.2 Bytes de 2 a 3 – Numero do parâmetro invalido.

## 4.4.28.3 Bytes de 4 a 7 – Estado da Comunicação:

Erros Seriais	Resposta	Descrição
EPSON_PORT_OPEN_SUCCESS	0000	Porta aberta com sucesso
EPSON_PORT_ALREADY_OPEN	0001	Porta já está aberta
EPSON_PORT_IN_USE	0002	Porta usada por outra aplicação
EPSON_INVALID_COM	0004	Número de porta inválido
EPSON_INVALID_BAUD_RATE	0005	Velocidade inválida
EPSON_INTERNAL_ERROR	000F	Erro interno da biblioteca

#### 4.4.28.4 Bytes de 8 a 11 – Estado da Impressora:

Bits	Valores permitidos
15	Impressora Offline
14	Erro de impressão
13	Tampa superior aberta
12	Estado da gaveta
11	Reservado
10 – 9	00 = Estação recibo 01 = Estação cheque 10 = Estação Autenticação (TMH6000 e TMU675) 11 = Leitura do MICR
8	Aguardando retirada do papel
7	Aguardando inserção do papel
6	Estado do sensor inferior da estação de cheque
5	Estado do sensor superior da estação do cheque
4	Estado do sensor de autenticação (TMH6000 e TMU675)
3 e 1	Sem papel
2 e 0	Pouco papel

## 4.4.28.5 Bytes de 12 à 15 – Estado fiscal:

Bits	Valores permitidos
15 – 14	00 = Modo bloqueado 01 = Reservado 10 = Modo manufatura (Não-Fiscalizado) 11 = Modo Fiscalizado
13	Reservado
12	0 = Modo de operação normal 1 = Modo de Intervenção Técnica
11 – 10	00 = Memória Fiscal em operação normal 01 = Memória Fiscal em esgotamento 10 = Memória Fiscal cheia 11 = Erro de leitura/escrita da Memória Fiscal
9 – 8	Reservado
7	0 = Período de vendas fechado 1 = Período de vendas aberto
6 – 4	Reservado
3 – 0	0000 = Documento fechado 0001 = Cupom Fiscal aberto 0010 = Comprovante de Crédito ou Débito 0011 = Estorno de Comprovante de Crédito ou Débito 0100 = Relatório Gerencial 1000 = Comprovante Não-Fiscal 1001 = Cheque ou autenticação (TMH6000 e TMU675)

**4.4.28.6 Bytes de 16 à 19 – Estado de execução do último comando:**[vide tabela 4.5.27.3](#)**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char szEstadoImpressora[21];  
Retorno = EPSON_Obter_Estado_Impressora ( szEstadoImpressora );
```

**Exemplo em C# :**

```
StringBuilder szEstadoImpressora = new StringBuilder(21, 21);  
Retorno = EPSON_Obter_Estado_Impressora ( szEstadoImpressora );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szEstadoImpressora As String  
szEstadoImpressora = Space(21)  
Retorno = EPSON_Obter_Estado_Impressora ( szEstadoImpressora )
```

**Exemplo em Delphi 7:**

```
szEstadoImpressora: array[0..21] of Char;  
Retorno := EPSON_Obter_Estado_Impressora ( szEstadoImpressora );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szEstadoImpressora := AnsiString(StringOfChar(' ', 21));  
Retorno := EPSON_Obter_Estado_Impressora(PAnsiChar(szEstadoImpressora));
```

**4.4.29 EPSON\_Obter\_GT**

Esta função lê o Grande Total (GT) atual.

**Sintaxe:**

```
function EPSON_Obter_GT(pszGT:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszGT	PCHAR	19 (N)	Dados recebidos da impressora. Este campo é preenchido com zeros à esquerda.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char szGT[19];  
Retorno = EPSON_Obter_GT ( szGT );
```

**Exemplo em C# :**

```
StringBuilder szGT = new StringBuilder(19, 19);  
Retorno = EPSON_Obter_GT ( szGT );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szGT As String  
szGT = Space(19)  
Retorno = EPSON_Obter_GT ( szGT )
```

**Exemplo em Delphi 7:**

```
szGT: array[0..19] of Char;  
Retorno := EPSON_Obter_GT ( szGT );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szGT := AnsiString(StringOfChar(' ', 19));  
iRetorno := EPSON_Obter_GT(PAnsiChar(szGT));
```



**4.4.30 EPSON\_Obter\_Linhas\_Impressas**

Esta função lê o número total de linhas impressas.

**Sintaxe:**

```
function EPSON_Obter_Linhas_Impressas(pszLinhasImpressas:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszLinhasImpressas	PCHAR	10	Dados recebidos da impressora.


**Retornos:**

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


**Requisitos:**

Nenhum.


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

```
 char szLinhasImpressas[10];  
Retorno = EPSON_Obter_Linhas_Impressas ( szLinhasImpressas );
```


**Exemplo em C# :**

```
 StringBuilder szLinhasImpressas = new StringBuilder(10, 10);  
Retorno = EPSON_Obter_Linhas_Impressas ( szLinhasImpressas );
```


**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
 Dim szLinhasImpressas As String  
szLinhasImpressas = Space(10)  
Retorno = EPSON_Obter_Linhas_Impressas ( szLinhasImpressas )
```

**Exemplo em Delphi 7:**

```
 szLinhasImpressas: array[0..10] of Char;  
Retorno := EPSON_Obter_Linhas_Impressas ( szLinhasImpressas );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
 szLinhasImpressas := AnsiString(StringOfChar(' ', 10));  
Retorno := EPSON_Obter_Linhas_Impressas (PAnsiChar(szLinhasImpressas));
```

**4.4.31 EPSON\_Obter\_Linhas\_Impressas\_RG**

Esta função lê o número total de linhas impressas em Relatórios Gerenciais.

**Sintaxe:**

```
function EPSON_Obter_Linhas_Impressas_RG(pszLinhasImpressasRG:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszLinhasImpressasRG	PCHAR	10	Dados recebidos da impressora.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szLinhasImpressasRG[10];  
Retorno = EPSON_Obter_Linhas_Impressas_RG ( szLinhasImpressasRG );
```

---

**Exemplo em C# :**

```
StringBuilder szLinhasImpressasRG = new StringBuilder(10, 10);  
Retorno = EPSON_Obter_Linhas_Impressas_RG ( szLinhasImpressasRG );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szLinhasImpressasRG As String  
szLinhasImpressasRG = Space(10)  
Retorno = EPSON_Obter_Linhas_Impressas_RG ( szLinhasImpressasRG )
```

---

**Exemplo em Delphi 7:**

```
szLinhasImpressasRG: array[0..10] of Char;  
Retorno := EPSON_Obter_Linhas_Impressas_RG ( szLinhasImpressasRG );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szLinhasImpressasRG := AnsiString(StringOfChar(' ', 10));  
Retorno:=EPSON_Obter_Linhas_Impressas_RG(PAnsiChar(szLinhasImpressasRG));
```

**4.4.32 EPSON\_Obter\_Linhas\_Impressas\_CCD**

Esta função lê o total de linhas impressas em Comprovantes de Crédito ou Débito.

**Sintaxe:**

```
function EPSON_Obter_Linhas_Impressas_CCD(pszLinhasImpressasCCD:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszLinhasImpressasCCD	PCHAR	10	Dados recebidos da impressora.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char szLinhasImpressasCCD[10];  
Retorno = EPSON_Obter_Linhas_Impressas_CCD ( szLinhasImpressasCCD );
```

**Exemplo em C# :**

```
StringBuilder szLinhasImpressasCCD = new StringBuilder(10, 10);  
Retorno = EPSON_Obter_Linhas_Impressas_CCD ( szLinhasImpressasCCD );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szLinhasImpressasCCD As String  
szLinhasImpressasCCD = Space(10)  
Retorno = EPSON_Obter_Linhas_Impressas_CCD ( szLinhasImpressasCCD )
```

**Exemplo em Delphi 7:**

```
szLinhasImpressasCCD: array[0..10] of Char;  
Retorno := EPSON_Obter_Linhas_Impressas_CCD ( szLinhasImpressasCCD );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szLinhasImpressasCCD := AnsiString(StringOfChar(' ', 10));  
Retorno:=EPSON_Obter_Linhas_Impressas_CCD(PAnsiChar(szLinhasImpressasCCD));
```

**4.4.33 EPSON\_Obter\_Dados\_Jornada**

Esta função recupera informações da jornada fiscal corrente.

**Sintaxe:**

```
function EPSON_Obter_Dados_Jornada(pszDadosJornada:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDadosJornada	PCHAR	68	Dados recebidos da impressora

Os dados serão retornados no seguinte formato:

Byte 0-7	:	Data do início da jornada fiscal
Byte 8-13	:	Hora do início da jornada fiscal
Byte 14-21	:	Data do final da jornada fiscal (se Z emitido)
Byte 22-27	:	Hora do final da jornada fiscal (se Z emitido)
Byte 28-33	:	COO inicial
Byte 34-39	:	COO final (se Z emitido)
Byte 40-45	:	Última redução Z
Byte 46-51	:	Relatórios Z restantes
Byte 52-57	:	Primeiro Cupom Fiscal
Byte 58-63	:	Último Cupom Fiscal
Byte 64	:	Estado da jornada: 0 – fechada, 1 – aberta
Byte 65	:	Período da jornada: 0 – vendas, 1 – limite atingido (RZ pendente)
Byte 66	:	Permissão de início de dia: 0 – não permitido (aguardando novo dia), 1 – permitido

**Retornos:**

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

**Requisitos:**

Modo normal de operação.

---

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

```
char szDadosJornada[68];  
Retorno = EPSON_Obter_Dados_Jornada ( szDadosJornada );
```

---

**Exemplo em C# :**

```
StringBuilder szDadosJornada = new StringBuilder(68, 68);  
Retorno = EPSON_Obter_Dados_Jornada ( szDadosJornada );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDadosJornada As String  
szDadosJornada = Space(68)  
Retorno = EPSON_Obter_Dados_Jornada ( szDadosJornada )
```

---

**Exemplo em Delphi 7:**

```
szDadosJornada: array[0..68] of Char;  
Retorno := EPSON_Obter_Dados_Jornada ( szDadosJornada );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
SzDadosJornada := AnsiString(StringOfChar( ' ', 68));  
Retorno := EPSON_Obter_Dados_Jornada (PAnsiChar(szDadosJornada));
```

**4.4.34 EPSON\_Obter\_Caracteres\_Linha**

Esta função lê o número máximo de caracteres que a impressora imprime em uma única linha.

**Sintaxe:**

```
function EPSON_Obter_Caracteres_Linha(pszCaracteresLinha:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszCaracteresLinha	PCHAR	4	Número de caracteres por linha.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char szCaracteresLinha[4];  
Retorno = EPSON_Obter_Caracteres_Linha ( szCaracteresLinha );
```

**Exemplo em C# :**

```
StringBuilder szCaracteresLinha = new StringBuilder(4, 4);  
Retorno = EPSON_Obter_Caracteres_Linha ( szCaracteresLinha );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCaracteresLinha As String  
szCaracteresLinha = Space(4)  
Retorno = EPSON_Obter_Caracteres_Linha ( szCaracteresLinha )
```

**Exemplo em Delphi 7:**

```
szCaracteresLinha: array[0..4] of Char;  
Retorno := EPSON_Obter_Caracteres_Linha ( szCaracteresLinha );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szCaracteresLinha := AnsiString(StringOfChar(` `, 4));  
Retorno := EPSON_Obter_Caracteres_Linha (PAnsiChar(szCaracteresLinha));
```

**4.4.35 EPSON\_Obter\_Operador**

Esta função retorna o nome do operador corrente.

**Sintaxe:**

```
function EPSON_Obter_Operador(pszNomeOperador:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszNomeOperador	PCHAR	21	Nome do operador.


**Retornos:**

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


**Requisitos:**

Nenhum.


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

```
 char szNomeOperador[21];  
Retorno = EPSON_Obter_Operador ( szNomeOperador );
```


**Exemplo em C# :**

```
 StringBuilder szNomeOperador = new StringBuilder(21, 21);  
Retorno = EPSON_Obter_Operador ( szNomeOperador );
```


**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
 Dim szNomeOperador As String  
szNomeOperador = Space(21)  
Retorno = EPSON_Obter_Operador ( szNomeOperador )
```

**Exemplo em Delphi 7:**

```
 szNomeOperador: array[0..21] of Char;  
Retorno := EPSON_Obter_Operador( szNomeOperador );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
 szNomeOperador := AnsiString(StringOfChar(` `, 21));  
Retorno := EPSON_Obter_Operador(PAnsiChar(szNomeOperador));
```



**4.4.36 EPSON\_Obter\_Numero\_Ultimo\_Item**

Esta função retorna o número do último item vendido no cupom.

**Sintaxe:**

```
function EPSON_Obter_Numero_Ultimo_Item(pszNumeroUltimoItem:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszNumeroUltimoItem	PCHAR	4(N)	Número do ultimo item vendido.

**Retornos:**

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

**Requisitos:**

Cupom aberto e pelo menos um item vendido.

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

```
char szNumeroUltimoItem[4];  
Retorno = EPSON_Obter_Numero_Ultimo_Item ( szNumeroUltimoItem );
```

**Exemplo em C# :**

```
StringBuilder szNumeroUltimoItem = new StringBuilder(4, 4);  
Retorno = EPSON_Obter_Numero_Ultimo_Item ( szNumeroUltimoItem );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szNumeroUltimoItem As String  
szNumeroUltimoItem = Space(4)  
Retorno = EPSON_Obter_Numero_Ultimo_Item ( szNumeroUltimoItem )
```

**Exemplo em Delphi 7:**

```
szNumeroUltimoItem: array[0..4] of Char;  
Retorno := EPSON_Obter_Numero_Ultimo_Item ( szNumeroUltimoItem );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szNumeroUltimoItem AnsiString(StringOfChar(' ', 4);  
Retorno := EPSON_Obter_Numero_Ultimo_Item (PAnsiChar(szNumeroUltimoItem));
```

**4.4.37 EPSON\_Obter\_Informacao\_Item**

Esta função retorna informações de qualquer item vendido no cupom.

**Sintaxe:**

```
function EPSON_Obter_Informacao_Item(pszNumeroltem:PChar; pszDadosItem:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroltem	PCHAR	3 (N)	Número do item a recuperar as informações.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDadosItem	PCHAR	76	Dados recebidos da impressora

Os dados serão retornados no seguinte formato:

Byte 0-13	:	Código do produto
Byte 14-21	:	Quantidade
Byte 22-24	:	Símbolo da unidade
Byte 25-36	:	Valor unitário do produto
Byte 37-38	:	Alíquota
Byte 39-50	:	Valor líquido do item
Byte 51-62	:	Desconto
Byte 63-74	:	Acréscimo

**Retornos:**

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

**Requisitos:**

Cupom fiscal aberto.

---

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

```
char szDadosItem[76];  
Retorno = EPSON_Obter_Informacao_Item ( "001", szDadosItem );
```

---

**Exemplo em C# :**

```
StringBuilder szDadosItem = new StringBuilder(76, 76);  
Retorno = EPSON_Obter_Informacao_Item ( "001", szDadosItem );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDadosItem As String  
szDadosItem = Space(76)  
Retorno = EPSON_Obter_Informacao_Item ( "001", szDadosItem )
```

---

**Exemplo em Delphi 7:**

```
szDadosItem: array[0..76] of Char;  
Retorno := EPSON_Obter_Informacao_Item ( '001', szDadosItem );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
SzDadosItem := AnsiString(StringOfChar(' ', 76));  
Retorno := EPSON_Obter_Informacao_Item ('001', PAnsiChar(szDadosItem));
```

**4.4.38 EPSON\_Obter\_Estado\_Cupom**

Esta função lê informações sobre o cupom atual.

**Sintaxe:**

```
function EPSON_Obter_Estado_Cupom(pszEstadoCupom:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszEstadoCupom	PCHAR	57	Estado do cupom atual.

Os dados serão retornados no seguinte formato:

Byte 0-1	:	Tipo de Cupom:
		00 – Não há cupom aberto.
		01 – Cupom Fiscal aberto.
		02 – Comprovante de Credito ou Debito aberto.
		03 – Cupom de cancelamento de CCD aberto.
		04 – Relatório Gerencial aberto.
		05 – Cupom Não-Fiscal aberto.

Somente se for um Cupom Fiscal, a sequência de caracteres abaixo será anexada aos dois descritos anteriormente:

Byte 2-7	:	Contador de Cupom Fiscal
Byte 8-20	:	Total do Cupom Fiscal
Byte 21-33	:	Total pago
Byte 34-37	:	Número de itens vendidos
Byte 38-41	:	Número de itens permitidos
Byte 42-45	:	Número de itens disponíveis
Byte 46-47	:	Número de alíquotas usadas
Byte 48-49	:	Número de alíquotas disponíveis
Byte 50-51	:	Número de pagamentos usados
Byte 52-53	:	Número de pagamentos permitidos
Byte 54-55	:	Fase atual do Cupom Fiscal segue os seguintes valores:
		'0' – Fase inicial.
		'1' – Fase de venda.
		'2' – Fase de descontos e acréscimos.
		'3' – Fase de pagamentos.
		'4' – Fase total impresso
		'5' – Fase final.

Somente se for um Cupom Não Fiscal, a sequência de caracteres abaixo será anexada aos dois descritos anteriormente:

Byte 2-7	:	Contador Geral Não Fiscal
Byte 8-20	:	Total do Cupom Não Fiscal
Byte 21-33	:	Total pago
Byte 34-37	:	Número de itens vendidos
Byte 38-41	:	Número de itens permitidos
Byte 42-45	:	Número de itens disponíveis
Byte 46-49	:	Não utilizado
Byte 50-51	:	Número de pagamentos usados
Byte 52-53	:	Número de pagamentos permitidos
Byte 54-55	:	Fase atual do Cupom Não Fiscal segue os seguintes valores: '0' – Fase inicial. '1' – Fase de venda. '2' – Fase de descontos e acréscimos. '3' – Fase de pagamentos. '4' – Fase total impresso '5' – Fase final.

**Retornos:**

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

**Requisitos:**

Período de vendas aberto e cupom aberto.

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

```
char szEstadoCupom[57];  
Retorno = EPSON_Obter_Estado_Cupom ( szEstadoCupom );
```

**Exemplo em C# :**

```
StringBuilder szEstadoCupom = new StringBuilder(57, 57);  
Retorno = EPSON_Obter_Estado_Cupom ( szEstadoCupom );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szEstadoCupom As String  
szEstadoCupom = Space(57)  
Retorno = EPSON_Obter_Estado_Cupom ( szEstadoCupom )
```

**Exemplo em Delphi 7:**

```
szEstadoCupom: array[0..57] of Char;  
Retorno := EPSON_Obter_Estado_Cupom ( szEstadoCupom );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szEstadoCupom := AnsiString(StringOfChar(' ', 57));  
Retorno := EPSON_Obter_Estado_Cupom (PAnsiChar(szEstadoCupom));
```

**4.4.39 EPSON\_Obter\_Informacao\_Ultimo\_Documento**

Esta função recupera informações do último documento impresso.

**Sintaxe:**

```
function EPSON_Obter_Informacao_Ultimo_Documento(pszInformacaoDocumento:PChar):Integer;StdCall;  
    External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszInformacaoDocumento	PCHAR	31	Informações do ultimo documento impresso.

Os dados serão retornados no seguinte formato:

Byte 0-1	:	Tipo do documento "0" - Nenhum documento "1" - Cupom Fiscal "2" - Redução Z "3" - Leitura X "5" - Leitura da Memória Fiscal "22" - Cupom adicional "23" - Cupom Fiscal - cancelamento "24" - Comprovante Não-Fiscal "25" - Comprovante Não-Fiscal – cancelamento "26" - Comprovante Não-Fiscal – estorno de meio de pagamento "27" - Comprovante de Crédito ou Débito "28" - Estorno de Comprovante de Crédito ou Débito "29" - Relatório Gerencial
Byte 2-9	:	Data de fechamento
Byte 10-15	:	Hora de fechamento
Byte 16-29	:	Total do cupom

**Retornos:**

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

**Requisitos:**

Modo normal de operação.

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

```
char szInformacaoDocumento[31];  
Retorno = EPSON_Obter_Informacao_Ultimo_Documento(szInformacaoDocumento);
```

**Exemplo em C# :**

```
StringBuilder szInformacaoDocumento = new StringBuilder(31, 31);  
Retorno = EPSON_Obter_Informacao_Ultimo_Documento(szInformacaoDocumento);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szInformacaoDocumento As String  
szInformacaoDocumento = Space(31)  
Retorno = EPSON_Obter_Informacao_Ultimo_Documento(szInformacaoDocumento)
```

**Exemplo em Delphi 7:**

```
szInformacaoDocumento: array[0..31] of Char;  
Retorno := EPSON_Obter_Informacao_Ultimo_Documento  
(szInformacaoDocumento);
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szInformacaoDocumento := AnsiString(StringOfChar(' ', 31));  
Retorno := EPSON_Obter_Informacao_Ultimo_Documento(PAnsiChar(  
szInformacaoDocumento));
```

**4.4.40 EPSON\_Obter\_Estado\_Corte\_Papel**

Esta função lê o status do corte de papel.

**Sintaxe:**

```
function EPSON_Obter_Estado_Corte_Papel(Var bCortePapel:Boolean):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pCortePapel	BOOLEAN	-	<b>True</b> - Corte de papel habilitado. <b>False</b> - Corte de papel desabilitado.

**Retornos:**

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


**Requisitos:**

Nenhum.


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

```
 bool bCortePapel;  
Retorno = EPSON_Obter_Estado_Corte_Papel ( &bCortePapel );
```

**Exemplo em C# :**

```
 Boolean bCortePapel=false;  
Retorno = EPSON_Obter_Estado_Corte_Papel ( ref bCortePapel );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
 Dim bCortePapel as Boolean  
Retorno = EPSON_Obter_Estado_Corte_Papel ( bCortePapel )
```

**Exemplo em Delphi 7:**

```
 bCortePapel: Boolean;  
Retorno := EPSON_Obter_Estado_Corte_Papel ( @bCortePapel );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
 bCortePapel: LongBool;  
Retorno := EPSON_Obter_Estado_Corte_Papel ( bCortePapel );
```



**4.4.41 EPSON\_Obter\_Linhas\_Impressas\_Vendas**

Esta função lê o total de mensagens de venda impressas.

**Sintaxe:**

```
function EPSON_Obter_Linhas_Impressas_Vendas(pszNumeroLinhasImpressas:PChar):Integer;StdCall;  
    External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszNumeroLinhasImpressas	PCHAR	10	Número de linhas impressas.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
unsigned char szNumeroLinhasImpressas[10];  
Retorno = EPSON_Obter_Linhas_Impressas_Vendas ( szNumeroLinhasImpressas );
```

**Exemplo em C# :**

```
StringBuilder szNumeroLinhasImpressas = new StringBuilder(10, 10);  
Retorno = EPSON_Obter_Linhas_Impressas_Vendas ( szNumeroLinhasImpressas );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szNumeroLinhasImpressas As String  
szNumeroLinhasImpressas = Space(10)  
Retorno = EPSON_Obter_Linhas_Impressas_Vendas ( szNumeroLinhasImpressas )
```

**Exemplo em Delphi 7:**

```
szNumeroLinhasImpressas: array[0..10] of Char;  
Retorno := EPSON_Obter_Linhas_Impressas_Vendas ( szNumeroLinhasImpressas  
);
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
SzNumeroLinhasImpressas := AnsiString(StringOfChar(' ', 10));  
Retorno := EPSON_Obter_Linhas_Impressas_Vendas(PAnsiChar(  
szNumeroLinhasImpressas));
```

**4.4.42 EPSON\_Obter\_Linhas\_Impressas\_Pagamentos**

Esta função lê o total de mensagens de pagamento impressas.

**Sintaxe:**

```
functionEPSON_Obter_Linhas_Impressas_Pagamentos(pszNumeroLinhasImpressas:PChar):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszNumeroLinhasImpressas	PCHAR	10	Número de linhas impressas.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
unsigned char szNumeroLinhasImpressas[10];  
Retorno = EPSON_Obter_Linhas_Impressas_Pagamentos (  
szNumeroLinhasImpressas );
```

---

**Exemplo em C# :**

```
StringBuilder szNumeroLinhasImpressas = new StringBuilder(10, 10);  
Retorno = EPSON_Obter_Linhas_Impressas_Pagamentos (  
szNumeroLinhasImpressas );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szNumeroLinhasImpressas As String  
szNumeroLinhasImpressas = Space(10)  
Retorno = EPSON_Obter_Linhas_Impressas_Pagamentos (  
szNumeroLinhasImpressas )
```

---

**Exemplo em Delphi 7:**

```
szNumeroLinhasImpressas: array[0..10] of Char;  
Retorno := EPSON_Obter_Linhas_Impressas_Pagamentos (  
szNumeroLinhasImpressas );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szNumeroLinhasImpressas := AnsiString(StringOfChar(' ', 10));  
Retorno := EPSON_Obter_Linhas_Impressas_Pagamentos(  
PAnsiChar(szNumeroLinhasImpressas));
```

**4.4.43 EPSON\_Obter\_Total\_Itens\_Vendidos**

Esta função lê o número de itens vendidos.

**Sintaxe:**

```
function EPSON_Obter_Total_Itens_Vendidos(pszTotalItens:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalItens	PCHAR	10	Número de itens vendidos.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
unsigned char szTotalItens[10];  
Retorno = EPSON_Obter_Total_Itens_Vendidos ( szTotalItens );
```

**Exemplo em C# :**

```
StringBuilder szTotalItens = new StringBuilder(10, 10);  
Retorno = EPSON_Obter_Total_Itens_Vendidos ( szTotalItens );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalItens As String  
szTotalItens = Space(10)  
Retorno = EPSON_Obter_Total_Itens_Vendidos ( szTotalItens )
```

**Exemplo em Delphi 7:**

```
szTotalItens: array[0..10] of Char;  
Retorno := EPSON_Obter_Total_Itens_Vendidos ( szTotalItens );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalItens := AnsiString(StringOfChar(' ', 10));  
Retorno := EPSON_Obter_Total_Itens_Vendidos(PAnsiChar(szTotalItens));
```

**4.4.44 EPSON\_Obter\_Estado\_Memoria\_Fiscal**

Esta função lê a quantidade total de memória e a quantidade usada da Memória Fiscal.

**Sintaxe:**

```
function EPSON_Obter_Estado_Memoria_Fiscal(pszEstadoMemoriaFiscal:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszEstadoMemoriaFiscal	PCHAR	19	Status da memória.

Os dados serão retornados no seguinte formato:

Byte 0-8	:	Total da memória
Byte 9-17	:	Quantidade usada da memória

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
unsigned char szEstadoMemoriaFiscal [19];  
Retorno = EPSON_Obter_Estado_Memoria_Fiscal ( szEstadoMemoriaFiscal );
```

---

**Exemplo em C# :**

```
StringBuilder szEstadoMemoriaFiscal = new StringBuilder(19, 19);  
Retorno = EPSON_Obter_Estado_Memoria_Fiscal ( szEstadoMemoriaFiscal );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szEstadoMemoriaFiscal As String  
szEstadoMemoriaFiscal = Space(19)  
Retorno = EPSON_Obter_Estado_Memoria_Fiscal ( szEstadoMemoriaFiscal )
```

---

**Exemplo em Delphi 7:**

```
szEstadoMemoriaFiscal: array[0..19] of Char;  
Retorno := EPSON_Obter_Estado_Memoria_Fiscal ( szEstadoMemoriaFiscal );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szEstadoMemoriaFiscal := AnsiString(StringOfChar(' ', 19));  
Retorno:=EPSON_Obter_Estado_Memoria_Fiscal(PAnsiChar(szEstadoMemoriaFiscal  
));
```

**4.4.45 EPSON\_Obter\_Estado\_MFD**

Esta função lê a quantidade total e a quantidade usada da Memória Fita Detalhe.

**Sintaxe:**

```
function EPSON_Obter_Estado_MFD(pszEstadoMFD:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszEstadoMFD	PCHAR	19	Status da memória

Os dados serão retornados no seguinte formato:

Byte 0-8	:	Total da Memória
Byte 9-17	:	Quantidade usada da memória

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
unsigned char szEstadoMFD[19];  
Retorno = EPSON_Obter_Estado_MFD ( szEstadoMFD );
```

---

**Exemplo em C# :**

```
StringBuilder szEstadoMFD = new StringBuilder(19, 19);  
Retorno = EPSON_Obter_Estado_MFD ( szEstadoMFD );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szEstadoMFD As String  
szEstadoMFD = Space(19)  
Retorno = EPSON_Obter_Estado_MFD ( szEstadoMFD )
```

---

**Exemplo em Delphi 7:**

```
szEstadoMFD: array[0..19] of Char;  
Retorno := EPSON_Obter_Estado_MFD ( szEstadoMFD );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szEstadoMFD := AnsiString(StringOfChar(' ', 19));  
Retorno := EPSON_Obter_Estado_MFD(PAnsiChar(szEstadoMFD));
```



**4.4.46 EPSON\_Obter\_Total\_Leituras\_X\_Impressas**

Esta função recupera o número correspondente ao total de Leituras X impressas até o momento.

**Sintaxe:**

```
function EPSON_Obter_Total_Leituras_X_Impressas(pszLeiturasXImpressas:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszLeiturasXImpressas	PCHAR	10	Número de Leituras X impressas.


**Retornos:**

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


**Requisitos:**

Nenhum.


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

```
 unsigned char szLeiturasXImpressas[10];  
Retorno = EPSON_Obter_Total_Leituras_X_Impressas ( szLeiturasXImpressas );
```


**Exemplo em C# :**

```
 StringBuilder szLeiturasXImpressas = new StringBuilder(10, 10);  
Retorno = EPSON_Obter_Total_Leituras_X_Impressas ( szLeiturasXImpressas );
```


**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
 Dim szLeiturasXImpressas As String  
szLeiturasXImpressas = Space(10)  
Retorno = EPSON_Obter_Total_Leituras_X_Impressas ( szLeiturasXImpressas );
```

**Exemplo em Delphi 7:**

```
 szLeiturasXImpressas: array[0..10] of Char;  
Retorno := EPSON_Obter_Total_Leituras_X_Impressas ( szLeiturasXImpressas  
);
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
 SzLeiturasXImpressas := AnsiString(StringOfChar(' ', 10));  
Retorno := EPSON_Obter_Total_Leituras_X_Impressas(PAnsiChar(  
szLeiturasXImpressas));
```

**4.4.47 EPSON\_Obter\_Estado\_Horario\_Verao**

Esta função lê o estado atual do horário de verão.

**Sintaxe:**

```
function EPSON_Obter_Estado_Horario_Verao(Var bEstadoHorarioVerao:Boolean):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
bEstadoHorarioVerao	BOOLEAN	-	Estado do horário de verão: <b>True</b> - Horário de verão ativo. <b>False</b> - Horário de verão inativo.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
int bEstadoHorarioVerao;  
Retorno = EPSON_Obter_Estado_Horario_Verao ( &bEstadoHorarioVerao );
```

---

**Exemplo em C#:**

```
Int16 bEstadoHorarioVerao;  
Retorno = EPSON_Obter_Estado_Horario_Verao ( ref bEstadoHorarioVerao );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim bEstadoHorarioVerao As Boolean  
Retorno = EPSON_Obter_Estado_Horario_Verao ( bEstadoHorarioVerao )
```

---

**Exemplo em Delphi 7:**

```
bEstadoHorarioVerao: Boolean;  
Retorno := EPSON_Obter_Estado_Horario_Verao ( @bEstadoHorarioVerao );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
bEstadoHorarioVerao: LongBool;  
Retorno := EPSON_Obter_Estado_Horario_Verao ( bEstadoHorarioVerao );
```

**4.4.48 EPSON\_Obter\_Venda\_Bruta**

Esta função obtém os dados de Venda Bruta da jornada anterior e da jornada atual.

**Sintaxe:**

```
function EPSON_Obter_Venda_Bruta(pszVendaBrutaAtual:PChar; pszVendaBrutaAnterior:PChar): Integer;  
    StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszVendaBrutaAtual	PCHAR	15	Venda Bruta atual
pszVendaBrutaAnterior	PCHAR	15	Venda Bruta da jornada anterior

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
unsigned char szVendaBrutaAtual[15];  
unsigned char szVendaBrutaAnterior[15];  
Retorno = EPSON_Obter_Venda_Bruta (szVendaBrutaAtual ,  
szVendaBrutaAnterior);
```

---

**Exemplo em C# :**

```
StringBuilder szVendaBrutaAtual = new StringBuilder(15, 15);  
StringBuilder VendaBrutaAnterior = new StringBuilder(15, 15);  
Retorno = EPSON_Obter_Venda_Bruta ( szVendaBrutaAtual,  
VendaBrutaAnterior );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szVendaBrutaAtual As String  
Dim VendaBrutaAnterior As String  
szVendaBrutaAtual = Space(15)  
VendaBrutaAnterior = Space(15)  
Retorno = EPSON_Obter_Venda_Bruta ( szVendaBrutaAtual,  
VendaBrutaAnterior )
```

---

**Exemplo em Delphi 7:**

```
szVendaBrutaAtual: array[0..15] of Char;  
szVendaBrutaAnterior: array[0..15] of Char;  
Retorno := EPSON_Obter_Venda_Bruta ( szVendaBrutaAtual ,  
szVendaBrutaAnterior );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szVendaBrutaAtual := AnsiString(StringOfChar(' ', 15));  
szVendaBrutaAnterior := AnsiString(StringOfChar(' ', 15));  
Retorno := EPSON_Obter_Venda_Bruta(PAnsiChar(szVendaBrutaAtual),  
PAnsiChar(szVendaBrutaAnterior));
```

**4.4.49 EPSON\_Obter\_Mensagem\_Erro**

Esta função traduz um código de erro para o descritivo de um erro, no formato de uma mensagem de texto. Este código de erro é obtido com o comando Epson\_Obter\_Estado\_Impressora, no campo Estado da execução do último comando.

**Sintaxe:**

```
function EPSON_Obter_Mensagem_Erro(pszCodigoErro:PChar; pszMensagemErro:PChar):Integer;StdCall;  
    External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszCodigoErro	PCHAR	4	String contendo o código de erro a ser traduzido, no formato hexadecimal com quatro dígitos ex.: "0304".

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszMensagemErro	PCHAR	100	String com até 100 caracteres contendo o descritivo em texto de um código de erro.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
unsigned char szMensagemErro [100] ;  
Retorno = EPSON_Obter_Mensagem_Erro ("0304" , szMensagemErro );
```

---

**Exemplo em C# :**

```
StringBuilder szMensagemErro = new StringBuilder(100, 100);  
Retorno = EPSON_Obter_Mensagem_Erro ( "0304",szMensagemErro );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szMensagemErro As String  
szMensagemErro = Space(100)  
Retorno = EPSON_Obter_Mensagem_Erro ( "0304", szMensagemErro )
```

---

**Exemplo em Delphi 7:**

```
szMensagemErro: array[0..100] of Char;  
Retorno := EPSON_Obter_Mensagem_Erro ( '0304',szMensagemErro );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
SzMensagemErro := AnsiString(StringOfChar(' ', 100));  
Retorno := EPSON_Obter_Mensagem_Erro('0304',PAnsiChar(szMensagemErro));
```

## 4.4.50 EPSON\_Obter\_Dados\_MF\_MFD

Esta função gera os seguintes arquivos: Espelhos dos Documentos (Fita-detelhe em meio digital), ATO/COTEPE 17/04 (CAT52 e Cupom Mania), relatório Sintegra e SPED Fiscal.

### Sintaxe:

```
function EPSON_Obter_Dados_MF_MFD(pszInicio:PChar; pszFinal:PChar; dwTipoEntrada:Integer;
dwEspelhos:Integer; dwAtoCotepe:Integer; dwSintegraSPED:Integer;
pszArquivoSaida:PChar): Integer;StdCall; External 'InterfaceEpson.dll';
```

### Entradas:

Variável	Tipo	Tam.	Descrição
pszInicio	PCHAR	9	String contendo Data, CRZ ou COO inicial
pszFinal	PCHAR	9	String contendo Data, CRZ ou COO final
dwTipoEntrada	INTEGER	-	<p>Inteiro indicando o formato de range de entrada, onde:</p> <ul style="list-style-type: none"> <li>0 - faixa de Data (campos Início/Fim com 8 dígitos)</li> <li>1 - faixa de CRZ (campos Início/Fim com 4 dígitos)</li> <li>2 - faixa de COO (campos Início/Fim com 6 ou 9 dígitos)</li> <li>3 - total</li> </ul> <p><b>Atenção:</b> Observe as faixas permitidas para cada tipo de leitura abaixo.</p>
dwEspelhos	INTEGER	-	<p>Este parâmetro é utilizado para Leitura de Espelhos ou XML de Documentos. Deve ser um número inteiro cujo valor serve de filtro por tipo de documento, onde:</p> <ul style="list-style-type: none"> <li>0 - Não gerar espelhos</li> <li>1 - CF, 2 - RZ, 4 - LMF, 8 - LX, 16 - RG, 32 - CCD, 64 - NF, 128 - CNF</li> <li>255 - Gerar todos os espelhos</li> <li>256 - Gerar XML de cupom fiscal</li> </ul> <p>Obs.: Para filtrar mais de um documento, basta somar os valores dos documentos desejados.</p>
dwAtoCotepe permitido apenas por faixa de data ou CRZ	INTEGER	-	<p>Este parâmetro é utilizado para geração de ATO/COTEPE 17/04.</p> <p><b>Atenção,</b> este procedimento serve como contingência, pois a InterfaceEpson gera este arquivo automaticamente em toda RZ. Para mais detalhes vide Cap. 8.</p> <p>Deve ser um número inteiro cujo valor especifica o tipo de saída, onde:</p> <ul style="list-style-type: none"> <li>0 - Não gera ATO/COTEPE</li> <li>1 - MF, ATO/COTEPE 17/04 da Memória Fiscal.</li> <li>2 - MFD, (<b>Cupom Mania -RJ</b>) ATO/COTEPE 17/04 da Memória Fita Detalhe.</li> <li>3 - TDM, (<b>CAT52</b>) ATO/COTEPE 17/04 de Todas as memórias.</li> <li>9 - MF, ATO/COTEPE 17/04 da Memória Fiscal mais arquivo de registros do PAF-ECF.</li> <li>10 - MFD, ATO/COTEPE 17/04 da Memória Fita Detalhe mais arquivo de registros do PAF-ECF.</li> <li>11 - TDM, (<b>CAT52</b>) ATO/COTEPE 17/04 de Todas as memórias mais arquivo de registros do PAF-ECF.</li> <li>21 - MF, ATO/COTEPE 17/04 da Memória Fiscal Versão 2.0.</li> </ul>



			<p>22 – MFD, ATO/COTEPE 17/04 da Memória Fita Detalhe Versão 2.0.</p> <p>23 – TDM, ATO/COTEPE 17/04 de Todas as memórias Versão 2.0.</p> <p>Obs.: Para gerar corretamente os registros do PAF-ECF o comando <a href="#">EPSON Config Dados PAFECE</a> deverá ser executado antes deste comando.</p>
<p>dwSintegraSPED</p> <p>permitido apenas por faixa de data</p>	INTEGER	-	<p>Este parâmetro é utilizado apenas para geração de relatório Sintegra. Inteiro cujo valor especifica o tipo de saída, onde:</p> <p>0 - Não gera Sintegra</p> <p>1 - tipo 60M, 2 - tipo 60A, 4 - tipo 60D, 8 - tipo 60I, 16 - tipo 60R, 32 - tipo 75, 64 - tipo 10, 11 e 90</p> <p>127 – Gerar todos registros do Sintegra</p> <p>256 – Gerar SPED perfil A (C400, C405, C410, C420, C460, C470, C490, C495 – se UF = Bahia)</p> <p>512 – Gerar SPED perfil B (C400, C405, C410, C420, C425, C490)</p> <p>768 – Gerar SPED perfil C (C400, C405, C420, C490)</p> <p>1024 – Gerar todos registros SPED</p> <p>1280 – Gerar SPED perfil A para CF-e (C400, C405, C420, C460, C465, C490)</p> <p>Obs.:</p> <ul style="list-style-type: none"> <li>- Para gerar mais de um tipo de registro basta somar os valores desejados.</li> <li>- Para gerar corretamente os registros 10 e 11, o comando <a href="#">EPSON Config Dados Sintegra</a> deverá ser executado antes deste comando.</li> <li>- Para gerar corretamente os registros SPED o comando <a href="#">EPSON Config Dados SPED</a> deverá ser executado antes deste comando.</li> </ul>
pszArquivoSaida	PCHAR	1024	<p>String contendo o nome do arquivo de saída, com o caminho completo, sem a extensão. Serão adicionados automaticamente os seguintes sufixos:</p> <p>"_ESP.txt" - para arquivo de Espelhos de Documentos</p> <p>"_CTP.txt" - para arquivo de ATO/COTEPE</p> <p>"_SIN.txt" - para arquivos de Sintegra</p> <p>"_SPED.txt" - para arquivos de SPED Fiscal</p>

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

ECF em modo normal de funcionamento.

**Observações:**

Caso o parâmetro “dwTipoEntrada” selecionado seja a opção “3 – Total”, todo o conteúdo da MFD será lido, dispensando portanto o preenchimento dos campos 1 e 2. O tempo de execução desta função pode variar entre alguns minutos e algumas horas, dependendo da porcentagem de utilização da MFD, do modelo do equipamento ECF e da velocidade de processamento do computador utilizado, o qual deve ter no mínimo 128 Mbytes de memória RAM e 600 Mbytes de espaço livre em disco.

Para leituras com intervalo grande (maior que 3 meses) é aconselhável o uso da Leitura Total (dwTipoEntrada = 3) para a obtenção de maior performance.

Por padrão os documentos ATO/COTEPE 17/04 e espelhos da MFD serão gerados com assinatura digital EAD, para desabilitar ou Habilitar essa assinatura deve ser utilizado o comando EPSON\_Config\_Habilita\_EAD.

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

```
Retorno = EPSON_Obter_Dados_MF_MFD( "1", "10", 2, 255, 0, 0,  
"c:\\Temp\\saida" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Obter_Dados_MF_MFD( "1", "10", 2, 255, 0, 0,  
"c:\\Temp\\saida" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Obter_Dados_MF_MFD( '1', '10', 2, 255, 0, 0,  
'c:\\Temp\\saida' );
```

## 4.4.51 EPSON\_Obter\_Dados\_Arquivos\_MF\_MFD

Esta função gera os seguintes arquivos: Espelhos dos Documentos (Fita-detelhe em meio digital), ATO/COTEPE 17/04 (arquivo de interface utilizado pelo Fisco – CAT52), relatório Sintegra e SPED Fiscal.

### Sintaxe:

```
function EPSON_Obter_Dados_Arquivos_MF_MFD( pszInicio:PChar;pszFim:Pchar;dwTipoEntrada:Integer;
dwEspelhos:Integer;dwAtoCOTEPE:Integer;
dwSintegra:Integer;pszArquivoSaida:PChar;
pszArquivoMF:PChar; pszArquivoMFD:PChar):
Integer;StdCall; External 'InterfaceEpson.dll';
```

### Entradas:

Variável	Tipo	Tam.	Descrição
pszInicio	PCHAR	9	String contendo Data ou COO inicial
pszFinal	PCHAR	9	String contendo Data ou COO final
dwTipoEntrada	INTEGER	-	Inteiro indicando o formato de range de entrada, onde: 0 - faixa de Data (campos Início/Fim com 8 dígitos) 1 - faixa de CRZ (campos Início/Fim com 4 dígitos) 2 - faixa de COO (campos Início/Fim com 6 ou 9 dígitos) 3 - total <b>Atenção:</b> Observe as faixas permitidas para cada tipo de leitura abaixo.
dwEspelhos	INTEGER	-	Este parâmetro é utilizado para Leitura de Espelhos ou XML de Documentos. Deve ser um número inteiro cujo valor serve de filtro por tipo de documento, onde: 0 – Não gerar espelhos 1 – CF, 2 – RZ, 4 – LMF, 8 – LX, 16 – RG, 32 – CCD, 64 – NF, 128 - CNF 255 – Gerar todos os espelhos 256 – Gerar XML de cupom fiscal Obs.: Para filtrar mais de um documento, basta somar os valores dos documentos desejados.
dwAtoCotepe  permitido apenas por faixa de data ou CRZ	INTEGER	-	Este parâmetro é utilizado para geração de ATO/COTEPE 17/04. <b>Atenção,</b> este procedimento serve como contingência, pois a InterfaceEpson gera este arquivo automaticamente em toda RZ. Para mais detalhes vide Cap. 8. Deve ser um número inteiro cujo valor especifica o tipo de saída, onde: 0 - Não gera ATO/COTEPE 1 – MF, ATO/COTEPE 17/04 da Memória Fiscal. 2 – MFD, ( <b>Cupom Mania – RJ</b> ) ATO/COTEPE 17/04 da Memória Fita Detalhe. 3 – TDM, ( <b>CAT52</b> ) ATO/COTEPE 17/04 de Todas as memórias. 9 – MF, ATO/COTEPE 17/04 da Memória Fiscal mais arquivo de registros do PAF-ECF. 10 – MFD, ATO/COTEPE 17/04 da Memória Fita Detalhe mais arquivo de registros do PAF-ECF. 11 – TDM, ( <b>CAT52</b> )ATO/COTEPE 17/04 de Todas as memórias.

			<p>21 – MF, ATO/COTEPE 17/04 da Memória Fiscal Versão 2.0.</p> <p>22 – MFD, ATO/COTEPE 17/04 da Memória Fita Detalhe Versão 2.0.</p> <p>23 – TDM, ATO/COTEPE 17/04 de Todas as memórias Versão 2.0. mais arquivo de registros do PAF-ECF.</p> <p>Obs.: Para gerar corretamente os registros do PAF-ECF o comando <a href="#">EPSON Config Dados PAFECE</a> deverá ser executado antes deste comando.</p>
<p>dwSintegraSPED</p> <p>permitido apenas por faixa de data</p>	INTEGER	-	<p>Este parâmetro é utilizado apenas para geração de relatório Sintegra. Inteiro cujo valor especifica o tipo de saída, onde:</p> <p>0 - Não gera Sintegra</p> <p>1 - tipo 60M, 2 - tipo 60A, 4 - tipo 60D, 8 - tipo 60I, 16 - tipo 60R, 32 - tipo 75, 64 - tipo 10, 11 e 90</p> <p>127 – Gerar todos registros do Sintegra</p> <p>256 – Gerar SPED perfil A (C400, C405, C410, C420, C460, C470, C490, C495 – se UF = Bahia)</p> <p>512 – Gerar SPED perfil B (C400, C405, C410, C420, C425, C490)</p> <p>768 – Gerar SPED perfil C (C400, C405, C420, C490)</p> <p>1024 – Gerar todos registros SPED</p> <p>1280 – Gerar SPED perfil A para CF-e (C400, C405, C420, C460, C465, C490)</p> <p>Obs.:</p> <ul style="list-style-type: none"> <li>- Para gerar mais de um tipo de registro basta somar os valores desejados.</li> <li>- Para gerar corretamente os registros 10 e 11, o comando <a href="#">EPSON Config Dados Sintegra</a> deverá ser executado antes deste comando.</li> <li>- Para gerar corretamente os registros SPED o comando <a href="#">EPSON Config Dados SPED</a> deverá ser executado antes deste comando.</li> </ul>
pszArquivoSaida	PCHAR	1024	<p>String contendo o nome do arquivo de saída, com o caminho completo, sem a extensão. Serão adicionados automaticamente os seguintes sufixos:</p> <p>"_ESP.txt" - para arquivo de Espelhos de Documentos</p> <p>"_CTP.txt" - para arquivo de ATO/COTEPE</p> <p>"_SIN.txt" - para arquivos de Sintegra</p> <p>"_SPED.txt" - para arquivos de SPED Fiscal</p>
pszArquivoMF	PCHAR	1024	Nome do arquivo da MF com o caminho completo.
pszArquivoMFD	PCHAR	1024	Nome do arquivo da MFD com o caminho completo.

## Saídas:

Nenhum.

## Retornos:

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

Não é necessário estar conectado ao ECF para executar esta função.

Para obter os arquivos binários para processamento utilize as funções EPSON\_Obter\_Arquivo\_Binario\_MFD e EPSON\_Obter\_Arquivo\_Binario\_MF.

**Observação:**

Por padrão os documentos ATO/COTEPE 17/04 e espelhos da MFD serão gerados com assinatura digital EAD, para desabilitar ou habilitar essa assinatura deve ser utilizado o comando EPSON\_Confi\_Habilita\_EAD.

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

```
Retorno = EPSON_Obter_Dados_Arquivos_MF_MFD("08012001", "08012010", 0,  
255, 2, 127, "c:\\saida\\", "C:\\EPSON\\MF.bin", "C:\\EPSON\\MFD.bin" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Obter_Dados_Arquivos_MF_MFD("08012001", "08012010", 0,  
255, 2, 127, "c:\\saida", "C:\\EPSON\\MF.bin" , "C:\\EPSON\\MFD.bin" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Obter_Dados_Arquivos_MF_MFD( '08012001', '08012010', 0,  
255, 2, 127, 'c:\\saida\\', 'C:\\EPSON\\MF.bin', 'C:\\EPSON\\MFD.bin');
```

## 4.4.52 EPSON\_Obter\_Dados\_MF\_MFD\_Progresso

Esta função gera os seguintes arquivos: Espelhos dos Documentos (Fita-detelhe em meio digital), ATO/COTEPE 17/04 (CAT52 e Cupom Mania), relatório Sintegra e SPED Fiscal.

### Sintaxe:

```
function EPSON_Obter_Dados_MF_MFD_Progresso (pszInicio:PChar;pszFim:PChar;dwTipoEntrada:Integer;
dwEspelhos:Integer;dwAtoCOTEPE:Integer;
dwSintegra:Integer;pszArquivoSaida:PChar;
pszProgresso:PChar):Integer;StdCall; External
'InterfaceEpson.dll';
```

### Entradas:

Variável	Tipo	Tam.	Descrição
pszInicio	PCHAR	8	String contendo Data, CRZ ou COO inicial
pszFinal	PCHAR	8	String contendo Data, CRZ ou COO final
dwTipoEntrada	INTEGER	-	Inteiro indicando o formato de range de entrada, onde: 0 - faixa de Data (campos Início/Fim com 8 dígitos) 1 - faixa de CRZ (campos Início/Fim com 4 dígitos) 2 - faixa de COO (campos Início/Fim com 6 dígitos) 3 - total <b>Atenção:</b> Observe as faixas permitidas para cada tipo de leitura abaixo.
dwEspelhos	INTEGER	-	Este parâmetro é utilizado para Leitura de Espelhos ou XML de Documentos. Deve ser um número inteiro cujo valor serve de filtro por tipo de documento, onde: 0 - Não gerar espelhos 1 - CF, 2 - RZ, 4 - LMF, 8 - LX, 16 - RG, 32 - CCD, 64 - NF, 128 - CNF 255 - Gerar todos os espelhos 256 - Gerar XML de cupom fiscal Obs.: Para filtrar mais de um documento, basta somar os valores dos documentos desejados.
dwAtoCotepe  permitido apenas por faixa de data ou CRZ	INTEGER	-	Este parâmetro é utilizado para geração de ATO/COTEPE 17/04. <b>Atenção,</b> este procedimento serve como contingência, pois a InterfaceEpson gera este arquivo automaticamente em toda RZ. Para mais detalhes vide Cap. 8. Deve ser um número inteiro cujo valor especifica o tipo de saída, onde: 0 - Não gera ATO/COTEPE 1 - MF, ATO/COTEPE 17/04 da Memória Fiscal. 2 - MFD, ( <b>Cupom Mania - RJ</b> ) ATO/COTEPE 17/04 da Memória Fita Detalhe. 3 - TDM, ( <b>CAT52</b> ) ATO/COTEPE 17/04 de Todas as memórias. 9 - MF, ATO/COTEPE 17/04 da Memória Fiscal mais arquivo de registros do PAF-ECF. 10 - MFD, ATO/COTEPE 17/04 da Memória Fita Detalhe mais arquivo de registros do PAF-ECF.

			<p>11 – TDM, (CAT52)ATO/COTEPE 17/04 de Todas as memórias mais arquivo de registros do PAF-ECF.</p> <p>21 – MF, ATO/COTEPE 17/04 da Memória Fiscal Versão 2.0.</p> <p>22 – MFD, ATO/COTEPE 17/04 da Memória Fita Detalhe Versão 2.0.</p> <p>23 – TDM, ATO/COTEPE 17/04 de Todas as memórias Versão 2.0.</p> <p>Obs.: Para gerar corretamente os registros do PAF-ECF o comando <a href="#">EPSON Config Dados PAFECE</a> deverá ser executado antes deste comando.</p>
<p>dwSintegraSPED</p> <p>permitido apenas por faixa de data</p>	INTEGER	-	<p>Este parâmetro é utilizado apenas para geração de relatório Sintegra. Inteiro cujo valor especifica o tipo de saída, onde:</p> <p>0 - Não gera Sintegra</p> <p>1 - tipo 60M, 2 - tipo 60A, 4 - tipo 60D, 8 - tipo 60I, 16 - tipo 60R, 32 - tipo 75, 64 - tipo 10, 11 e 90</p> <p>127 – Gerar todos registros do Sintegra</p> <p>256 – Gerar SPED perfil A (C400, C405, C410, C420, C460, C470, C490, C495 – se UF = Bahia)</p> <p>512 – Gerar SPED perfil B (C400, C405, C410, C420, C425, C490)</p> <p>768 – Gerar SPED perfil C (C400, C405, C420, C490)</p> <p>1024 – Gerar todos registros SPED</p> <p>1280 – Gerar SPED perfil A para CF-e (C400, C405, C420, C460, C465, C490)</p> <p>Obs.:</p> <ul style="list-style-type: none"> <li>- Para gerar mais de um tipo de registro basta somar os valores desejados.</li> <li>- Para gerar corretamente os registros 10 e 11, o comando <a href="#">EPSON Config Dados Sintegra</a> deverá ser executado antes deste comando.</li> <li>- Para gerar corretamente os registros SPED o comando <a href="#">EPSON Config Dados SPED</a> deverá ser executado antes deste comando.</li> </ul>
pszArquivoSaida	PCHAR	1024	<p>String contendo o nome do arquivo de saída, com o caminho completo, sem a extensão. Serão adicionados automaticamente os seguintes sufixos:</p> <p>"_ESP.txt" - para arquivo de Espelhos de Documentos</p> <p>"_CTP.txt" - para arquivo de ATO/COTEPE</p> <p>"_SIN.txt" - para arquivos de Sintegra</p> <p>"_SPED.txt" - para arquivos de SPED Fiscal</p>

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszProgresso	PCHAR	4	Percentual realizado da operação.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

ECF em modo normal de funcionamento.

**Observações:**

Este comando é similar ao comando EPSON\_Obter\_Dados\_MF\_MFD, apenas incluindo o campo pszProgresso como parâmetro de saída. Esta variável será preenchida pela InterfaceEpson com o percentual do andamento da leitura dos dados, permitindo o aplicativo exibir este percentual decorrente do andamento para o usuário.

Caso o parâmetro "dwTipoEntrada" selecionado seja a opção "3 – Total", todo o conteúdo da MFD será lido, dispendendo portanto o preenchimento dos campos 1 e 2. O tempo de execução desta função pode variar entre alguns minutos e algumas horas, dependendo da porcentagem de utilização da MFD, do modelo do equipamento ECF e da velocidade de processamento do computador utilizado, o qual deve ter no mínimo 128 Mbytes de memória RAM e 600 Mbytes de espaço livre em disco.

Para leituras com intervalo grande (maior que 3 meses) é aconselhável o uso da Leitura Total (dwTipoEntrada = 3) para a obtenção de maior performance.

Por padrão os documentos ATO/COTEPE 17/04 e espelhos da MFD serão gerados com assinatura digital EAD, para desabilitar ou Habilitar essa assinatura deve ser utilizado o comando EPSON\_Config\_Habilita\_EAD.

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

```
unsigned char szProgresso [5];  
Retorno = EPSON_Obter_Dados_MF_MFD_Progresso( "1", "10", 2, 255, 0, 0,  
"c:\\Temp\\saida", szProgresso );
```

**Exemplo em C# :**

```
StringBuilder szProgresso = new StringBuilder(5,5);  
Retorno = EPSON_Obter_Dados_MF_MFD_Progresso ( "1", "10", 2, 255, 0, 0,  
"c:\\Temp\\saida", szProgresso );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szProgresso As String  
szProgresso = Space(5)  
Retorno = EPSON_Obter_Dados_MF_MFD_Progresso ( "1", "10", 2, 255, 0, 0,  
"c:\\Temp\\saida", szProgresso );
```

**Exemplo em Delphi 7:**

```
szProgresso: array[0...5] of Char;  
Retorno := EPSON_Obter_Dados_MF_MFD_Progresso ( '1', '10', 2, 255, 0, 0,  
'c:\\Temp\\saida', szProgresso );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
SzProgresso := AnsiString(StringOfChar( ' ', 5));  
Retorno := EPSON_Obter_Dados_MF_MFD_Progresso( '1', '10', 2, 255, 0, 0,  
'c:\\Temp\\saida', PAnsiChar(SzProgresso));
```



**4.4.53 EPSON\_Obter\_Versao\_DLL**

Esta função recupera a versão da DLL.

**Sintaxe:**

```
function EPSON_Obter_Versao_DLL(pszVersaoDLL:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszVersaoDLL	PCHAR	10	String com até 9 caracteres contendo o número de versão da DLL.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
unsigned char szVersaoDLL[10];  
Retorno = EPSON_Obter_Versao_DLL ( szVersaoDLL );
```

**Exemplo em C# :**

```
StringBuilder szVersaoDLL = new StringBuilder(10, 10);  
Retorno = EPSON_Obter_Versao_DLL ( szVersaoDLL );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szVersaoDLL As String  
szVersaoDLL = Space(10)  
Retorno = EPSON_Obter_Versao_DLL ( szVersaoDLL )
```

**Exemplo em Delphi 7:**

```
szVersaoDLL: array[0..10] of Char;  
Retorno := EPSON_Obter_Versao_DLL ( szVersaoDLL );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szVersaoDLL := AnsiString(StringOfChar(' ', 10));  
Retorno := EPSON_Obter_Versao_DLL(PAnsiChar(szVersaoDLL));
```

**4.4.54 EPSON\_Obter\_Total\_JornadaEX**

Retorna os totais da jornada fiscal.

**Sintaxe:**

```
function EPSON_Obter_Total_JornadaEX( pszOpclcmsISS:Pchar;pszNumeroCRZ:PChar;  
                                     pszTotalJornada:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszOpclcmsISS	PCHAR	1	T para ICMS, Diferente de T para ISS.
pszNumeroCRZ	PCHAR	4	Número da Redução Z.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTotalJornada	PCHAR	602 bytes	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0 - 3 :	Número da Redução Z
Byte 4 - 9 :	Número do COO
Byte 10 - 15 :	CRO
Byte 16 - 29 :	Venda Bruta Diária
Byte 30 - 42 :	Total F
Byte 43 - 55 :	Total I
Byte 56 - 68 :	Total N
Byte 69 - 81 :	Total FS
Byte 82 - 94 :	Total IS
Byte 95 - 107 :	Total NS
Byte 108 - 120 :	Total Canc. ICMS
Byte 121 - 133 :	Total Canc. ISS
Byte 134 - 146 :	Total Canc. NF
Byte 147 - 159 :	Total Desc. ICMS
Byte 160 - 172 :	Total Desc. ISS
Byte 173 - 185 :	Total Desc. NF
Byte 186 - 198 :	Total Acre. ICMS
Byte 199 - 211 :	Total Acre. ISS
Byte 212 - 224 :	Total Acre. NF
Byte 225 - 237 :	Total ICMS
Byte 238 - 250 :	Total ISS
Byte 251 - 263 :	Total NF
Byte 264 - 271 :	Data de Fechamento RZ

Byte 272 - 277 :	Hora de Fechamento RZ
Byte 278 - 294 :	Totalizador Geral
Byte X - X+3:	Percentual do Totalizador parcial
Byte Y - Y+12:	Total vendido do Totalizador parcial

X igual  $295+n*17(0\leq n\leq 17)$ Y igual  $299+n*17(0\leq n\leq 17)$ **Retornos:**

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

**Requisitos:**

Modo normal de operação.

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

```
unsigned char szTotalJornada[602];  
Retorno = EPSON_Obter_Total_JornadaEX("T", "15", szTotalJornada);
```

**Exemplo em C# :**

```
StringBuilder szTotalJornada = new StringBuilder(602, 602);  
Retorno = EPSON_Obter_Total_JornadaEX("T", "15", szTotalJornada);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTotalJornada As String  
szTotalJornada = Space(602)  
Retorno = EPSON_Obter_Total_JornadaEX("T", "15", szTotalJornada)
```

**Exemplo em Delphi 7:**

```
szTotalJornada: array[0..602] of Char;  
Retorno := EPSON_Obter_Total_JornadaEX('T', '15', szTotalJornada);
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTotalJornada := AnsiString(StringOfChar(' ', 602));  
Retorno:=EPSON_Obter_Total_JornadaEX('T', '15', PAnsiChar(szTotalJornada));
```

**4.4.55 EPSON\_Obter\_Dados\_Ultima\_RZ**

Retorna informações referentes à última redução fiscal efetuada na máquina.

**Sintaxe:**

```
function EPSON_Obter_Dados_Ultima_RZ(pszUltimaRZ:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszUltimaRZ	PCHAR	1168 bytes	Dados recebidos da impressora, terminados com um caractere nulo.

Os dados serão retornados no seguinte formato:

Byte 0 – 7 :	Data de Emissão da Redução Z
Byte 8 – 13:	Hora de Emissão da Redução Z
Byte 14 – 19:	COO Inicial (Abertura da Jornada Fiscal)
Byte 20 – 25:	COO Final (Redução Z)
Byte 26- 31:	CRZ
Byte 32 - 37:	CRO
Byte 38 - 43:	GNF
Byte 44 - 49:	CDC
Byte 50 - 55:	NFC
Byte 56 - 61:	GRG
Byte 62 - 67:	CCF
Byte 68 - 73:	CFC
Byte 74 - 79:	CFD
Byte 80 - 85:	NCN
Byte 86 - 103:	Totalizador Geral
Byte 104 - 120:	Cancelamento ICMS
Byte 121 - 137:	Cancelamento ISSQN
Byte 138 - 154:	Cancelamento Não-Fiscal
Byte 155 - 171:	Desconto ICMS
Byte 172 - 188:	Desconto ISSQN
Byte 189 - 205:	Desconto Não-Fiscal
Byte 206 - 222:	Acréscimo ICMS
Byte 223 - 239:	Acréscimo ISSQN
Byte 240 - 256:	Acréscimo Não-Fiscal
Byte 257 - 376:	Tributos (ICMS, ISSQN, F, I, N, FS, IS, NS) – cada tributo ocupa 5 caracteres

Byte 377 - 382:	Reservado
Byte 383 - 790:	Totalizadores Parciais Tributados – Cada totalizador ocupa 17 caracteres e corresponde, na ordem, a cada Tributo do campo anterior
Byte 791 - 816:	Reservado
Byte 817 - 1156:	Totalizadores Não Fiscais. São 20 totalizadores sendo que cada totalizador é composto por 17 bytes (os 13 primeiros com o valor acumulado e os 4 últimos bytes indicam o CON - Contador de Totalizadores Não-Fiscais).
Byte 1157 - 1158:	Número de Alíquotas cadastradas.
Byte 1159 - 1166:	Data do Movimento.

**Retornos:**

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

**Requisitos:**

O comando só retornará tais informações se a última redução for efetuada através dos comandos [EPSON RelatorioFiscal RZ](#) ou [EPSON RelatorioFiscal RZEX](#). Caso contrário, retornará alguns dos campos preenchidos com interrogações “?” indicando a falta da informação.

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

```
unsigned char  szUltimaRZ [1168];  
Retorno =     EPSON_Obter_Dados_Ultima_RZ( szUltimaRZ );
```

**Exemplo em C# :**

```
StringBuilder szUltimaRZ = new StringBuilder(1168, 1168);  
Retorno =     EPSON_Obter_Dados_Ultima_RZ ( szUltimaRZ );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szUltimaRZ As String  
szUltimaRZ = Space(1168)  
Retorno =     EPSON_Obter_Dados_Ultima_RZ ( szUltimaRZ )
```

**Exemplo em Delphi 7:**

```
szUltimaRZ: array[0..1168] of Char;  
Retorno :=     EPSON_Obter_Dados_Ultima_RZ ( szUltimaRZ );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
SzUltimaRZ := AnsiString(StringOfChar(' ', 1168));  
Retorno :=     EPSON_Obter_Dados_Ultima_RZ( PAnsiChar(SzUltimaRZ));
```

**4.4.56 EPSON\_Obter\_AtoCOTEPE\_SeparadoEX**

Essa função separa as jornadas de um relatório Ato COTEPE 17/04 (CAT52).

**Sintaxe:**

```
function EPSON_Obter_AtoCOTEPE_SeparadoEX( pszNomeArquivo:PChar; dwOpcaoNomeacao:Integer )  
Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNomeArquivo	PCHAR	1024	Nome do arquivo.
dwOpcaoNomeacao	INTEGER	-	0 – Nomeia os arquivos por CRZ 1 – Nomeia os arquivos de acordo com a CAT52.

**Saídas:**

Será gerado um arquivo para cada jornada encontrada.

**dwOption = 0:**

O nome dos arquivos de saída serão iguais ao do arquivo de origem acrescido do numero da redução Z.

Ex.: Ato\_COTEPE\_0001.txt (RZ 0001)

Ato\_COTEPE\_0002.txt (RZ 0002)

**dwOption = 1:**

O nome dos arquivos de saída serão gerados de acordo com a CAT52.

Ex.:

EP000001.J88

EP000001.838

**Retornos:**

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

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

```
Retorno = EPSON_Obter_AtoCOTEPE_SeparadoEX("c:\\Epson\\Ato_COTEPE.txt", 0);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Obter_AtoCOTEPE_SeparadoEX("c:\\Epson\\Ato_COTEPE.txt", 0)
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno:= EPSON_Obter_AtoCOTEPE_SeparadoEX('c:\\Epson\\Ato_COTEPE.txt', 0);
```

**4.4.57 EPSON\_Obter\_Arquivos\_Binarios**

Esta função gera os arquivos de Binário da MF e MFD da máquina.

O tempo de execução desta operação pode variar entre alguns minutos e algumas horas, dependendo da porcentagem de utilização da MFD, e da velocidade de processamento do computador utilizado.

**Sintaxe:**

```
function EPSON_Obter_Arquivos_Binarios(pszInicio:PChar; pszFinal:PChar; dwTipoEntrada:Integer;  
                                       pszArquivoMF:PChar; pszArquivoMFD:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszInicio	PCHAR	8	String contendo Data, COO ou CRZ inicial.
pszFinal	PCHAR	8	String contendo Data, COO ou CRZ Final.
dwTipoEntrada	INTEGER	-	Inteiro indicando o formato de range de entrada, onde: <b>0</b> - Faixa de Data. <b>1</b> - Faixa de CRZ. <b>2</b> - Faixa de COO. <b>3</b> - Total.
pszArquivoMF	PCHAR	1024	Nome do arquivo da MF com o caminho completo.
pszArquivoMFD	PCHAR	1024	Nome do arquivo da MFD com o caminho completo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
Retorno = EPSON_Obter_Arquivos_Binarios("08012001", "08012012", 0,  
"C:\\EPSON\\MF.bin", "C:\\EPSON\\MFD.bin");
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Obter_Arquivos_Binarios("08012001", "08012012", 0,  
"C:\\EPSON\\MF.bin", "C:\\EPSON\\MFD.bin")
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Obter_Arquivos_Binarios('08012001', '08012012', 0,  
'C:\\EPSON\\MF.bin', 'C:\\EPSON\\MFD.bin');
```



**4.4.58 EPSON\_Obter\_Arquivo\_Binario\_MF**

Esta função extrai os dados da memória fiscal para um arquivo Binário .

**Sintaxe:**

```
function EPSON_Obter_Arquivo_Binario_MF(pszNomeArquivo:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNomeArquivo	PCHAR	1024	Nome do arquivo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Documentos fechados.

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

```
Retorno = EPSON_Obter_Arquivo_Binario_MF( " c:\\Temp\\MF.bin");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Obter_Arquivo_Binario_MF( "c:\\Temp\\MF.bin" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Obter_Arquivo_Binario_MF( ' c:\\Temp\\MF.bin' );
```

**4.4.59 EPSON\_Obter\_Arquivo\_Binario\_MFD**

Esta função gera um arquivo de Binário da MFD da máquina.

O tempo de execução desta operação pode variar entre alguns minutos e algumas horas, dependendo da porcentagem de utilização da MFD, e da velocidade de processamento do computador utilizado.

**Sintaxe:**

```
function EPSON_Obter_Arquivo_Binario_MFD(pszNomeArquivoMFD:PChar):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNomeArquivoMFD	PCHAR	1024	Nome do arquivo que será gerado com path completo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando não disponível para ECF modelos TM-H6000 FB, TM-T81 FB e TM-T88 FB.

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

```
Retorno = EPSON_Obter_Arquivo_Binario_MFD("c:\\saida\\bin_mfd.bin");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Obter_Arquivo_Binario_MFD("c:\\saida\\bin_mfd.bin")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Obter_Arquivo_Binario_MFD( 'c:\\saida\\bin_mfd.bin' );
```

**4.4.60 EPSON\_Obter\_Versao\_SWBasicoEX**

Esta função retorna versão do software básico, data de sua instalação e hora de instalação.

**Sintaxe:**

```
function EPSON_Obter_Versao_SWBasicoEX ( pszVersaoSWBasico:PChar; pszDataInstalacao:PChar;  
                                          pszHoraInstalacao:PChar):Integer;StdCall; External  
                                          'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszVersaoSWBasico	PCHAR	9	Versão Software Básico
pszDataInstalacao	PCHAR	9	Data de Instalação
pszHoraInstalacao	PCHAR	7	Hora de Instalação

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szVersaoSWBasico[9];  
char szDataInstalacao [9];  
char szHoraInstalacao [7];  
Retorno = EPSON_Obter_Versao_SWBasicoEX ( szVersaoSWBasico,  
szDataInstalacao, szHoraInstalacao );
```

---

**Exemplo em C# :**

```
StringBuilder szVersaoSWBasico = new StringBuilder(9, 9);  
StringBuilder szDataInstalacao = new StringBuilder(9, 9);  
StringBuilder szHoraInstalacao = new StringBuilder(7, 7);  
Retorno = EPSON_Obter_Versao_SWBasicoEX ( szVersaoSWBasico,  
szDataInstalacao, szHoraInstalacao );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szVersaoSWBasico As String  
Dim szDataInstalacao As String  
Dim szHoraInstalacao As String  
szVersaoSWBasico = Space(9)  
szDataInstalacao = Space(9)  
szHoraInstalacao = Space(7)  
Retorno = EPSON_Obter_Versao_SWBasicoEX( szVersaoSWBasico,  
szDataInstalacao, szHoraInstalacao )
```

---

**Exemplo em Delphi 7:**

```
szVersaoSWBasico: array[0..9] of Char;  
szDataInstalacao: array[0..9] of Char;  
szHoraInstalacao: array[0..7] of Char;  
Retorno := EPSON_Obter_Versao_SWBasicoEX ( szVersaoSWBasico,  
szDataInstalacao, szHoraInstalacao );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szVersaoSWBasico := AnsiString(StringOfChar(' ', 8));  
szDataInstalacao := AnsiString(StringOfChar(' ', 8));  
szHoraInstalacao := AnsiString(StringOfChar(' ', 6));  
Retorno := EPSON_Obter_Versao_SWBasicoEX(PAnsiChar(szVersaoSWBasico),  
PAnsiChar(szDataInstalacao), PAnsiChar(szHoraInstalacao));
```

**4.4.61 EPSON\_Obter\_Codigo\_Nacional\_ECF**

Esta função retorna o código nacional de identificação do ECF e também o nome de arquivo exigido pelo PAFECF no formato CCCCCNNNNNNNNNNNNNNDDMMAAAA.txt

Sendo:

“CCCCC” o Código Nacional de Identificação de ECF.

“NNNNNNNNNNNNNNNN” os 14 (quatorze) últimos dígitos do número de fabricação do ECF.

“DDMMAAAA” data atual da impressora no formato dia/mês/ano.

**Sintaxe:**

```
function EPSON_Obter_Codigo_Nacional_ECF(pszCodigoNacionalECF:PChar; pszNomeArquivo:PChar):  
Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszCodigoNacionalECF	PCHAR	7	Código Nacional do ECF
pszNomeArquivo	PCHAR	33	Nome do Arquivo PAFECF

**Retornos:**

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

**Requisitos:**

Nenhum.

**Observação:**

Nome de arquivo somente valido quando exigida data atual da maquina no nome do arquivo.  
Para o **Simulador** o retorno do código será “000000”.

---

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

```
char szCodigoNacionalECF[7];  
char szNomeArquivo [33];  
Retorno = EPSON_Obter_Codigo_Nacional_ECF (szCodigoNacionalECF ,  
szNomeArquivo);
```

---

**Exemplo em C# :**

```
StringBuilder szCodigoNacionalECF = new StringBuilder(7, 7);  
StringBuilder szNomeArquivo = new StringBuilder(33, 33);  
Retorno = EPSON_Obter_Codigo_Nacional_ECF (szCodigoNacionalECF ,  
szNomeArquivo);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szCodigoNacionalECF As String  
Dim szNomeArquivo As String  
szCodigoNacionalECF = Space(7)  
szNomeArquivo = Space(33)  
Retorno = EPSON_Obter_Codigo_Nacional_ECF (szCodigoNacionalECF ,  
szNomeArquivo)
```

---

**Exemplo em Delphi 7:**

```
szCodigoNacionalECF: array[0..7] of Char;  
szNomeArquivo: array[0..33] of Char;  
Retorno := EPSON_Obter_Codigo_Nacional_ECF (szCodigoNacionalECF ,  
szNomeArquivo);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
SzCodigoNacionalECF := AnsiString(StringOfChar(' ', 6));  
szNomeArquivo := AnsiString(StringOfChar(' ', 32));  
Retorno := EPSON_Obter_Codigo_Nacional_ECF(PAnsiChar(szCodigoNacionalECF)  
, PAnsiChar(szNomeArquivo));
```

**4.4.62 EPSON\_Obter\_Numero\_Usuario**

Esta função retorna o numero do Usuário.

**Sintaxe:**

```
function EPSON_Obter_Numero_Usuario(pszNumeroUsuario:PChar): Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszNumeroUsuario	PCHAR	2	Número do Usuário.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
char pszNumeroUsuario[2];  
Retorno = EPSON_Obter_Numero_Usuario ( pszNumeroUsuario );
```

**Exemplo em C# :**

```
StringBuilder pszNumeroUsuario = new StringBuilder(2, 2);  
Retorno = EPSON_Obter_Numero_Usuario ( pszNumeroUsuario );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim pszNumeroUsuario As String  
pszNumeroUsuario = Space(2)  
Retorno = EPSON_Obter_Numero_Usuario ( pszNumeroUsuario )
```

**Exemplo em Delphi 7:**

```
pszNumeroUsuario: array[0..2] of Char;  
Retorno := EPSON_Obter_Numero_Usuario ( pszNumeroUsuario );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
pszNumeroUsuario := AnsiString(StringOfChar(' ', 2));  
Retorno := EPSON_Obter_Numero_Usuario( PAnsiChar(pszNumeroUsuario));
```

**4.4.63 EPSON\_Obter\_Arredonda\_Trunca\_Fabricacao**

Esta função verifica se a impressora está configurada para truncar ou arredondar o valor da venda do item, bem como se está configurada para vender o item como fabricação própria ou de terceiros.

**Sintaxe:**

```
function EPSON_Obter_Arredonda_Trunca_Fabricacao(pszOpcArredondaTrunca:PChar;  
                                                  pszOpcFabricacaoPropria:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszOpcArredondaTrunca	PCHAR	2	Flag que indica a opção padrão para o cálculo do valor do item. <b>S</b> – Arredonda. <b>N</b> – Trunca.
pszOpcFabricacaoPropria	PCHAR	2	Flag que indica a opção padrão para fabricação do item. <b>S</b> – Fabricação própria. <b>N</b> – Fabricação por Terceiros.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelos TM-H6000 FBIII, TM-T81 FBIII e TM-T88 FBIII.



---

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

```
char pszOpcArredontaTrunca[2], pszOpcFabricacaoPropria[2];  
Retorno = EPSON_Obter_Arredonda_Trunca_Fabricacao(pszOpcArredontaTrunca,  
pszOpcFabricacaoPropria);
```

---

**Exemplo em C# :**

```
StringBuilder pszOpcArredontaTrunca = new StringBuilder(2, 2);  
StringBuilder pszOpcFabricacaoPropria = new StringBuilder(2, 2);  
Retorno = EPSON_Obter_Arredonda_Trunca_Fabricacao(pszOpcArredontaTrunca,  
pszOpcFabricacaoPropria);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim pszOpcArredontaTrunca, pszOpcFabricacaoPropria As String  
pszOpcArredontaTrunca = Space(2)  
pszOpcFabricacaoPropria = Space(2)  
Retorno = EPSON_Obter_Arredonda_Trunca_Fabricacao(pszOpcArredontaTrunca,  
pszOpcFabricacaoPropria)
```

---

**Exemplo em Delphi 7:**

```
pszOpcArredontaTrunca: array[0..2] of Char;  
pszOpcFabricacaoPropria: array[0..2] of Char;  
Retorno := EPSON_Obter_Arredonda_Trunca_Fabricacao(pszOpcArredontaTrunca,  
pszOpcFabricacaoPropria);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
PszOpcArredontaTrunca := AnsiString(StringOfChar(' ', 1));  
pszOpcFabricacaoPropria := AnsiString(StringOfChar(' ', 1));  
Retorno :=  
EPSON_Obter_Arredonda_Trunca_Fabricacao(PAnsiChar(pszOpcArredontaTrunca),  
PAnsiChar(pszOpcFabricacaoPropria));
```

**4.4.64 EPSON\_Obter\_Log\_Comandos**

Esta função retorna o log dos últimos 10 eventos (comando ou bloqueio) ocorridos na impressora.

**Sintaxe:**

```
function EPSON_Obter_Log_Comandos( pszTabelaLog:PChar; pszTamanhoByteLog:PChar): Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTabelaLog	PCHAR	271 bytes	Log dos eventos da impressora.
pszTamanhoByteLog	PCHAR	4 bytes	Quantidade de bytes do Log.

Os dados na variável pszTabelaLog serão retornados no seguinte formato:

Byte 0 - 7	: Data de execução do evento no formato DDMMAAAA
Byte 8 - 11	: Hora de execução do evento no formato HHMM
Byte 12	: Tipo do bloqueio ( para os comandos este valor será preenchido com '0')
Byte 13 - 16	: Nome do bloqueio ( para os comandos este valor será preenchido com 'NONE')
Byte 17 - 21	: Número do bloqueio ( para os comandos este valor será preenchido com '00000')
Byte 22 - 25	: Número do comando ( para os bloqueios este valor será preenchido com '0000')
Byte 26	: Tipo do evento: <b>A</b> : Comando automático <b>B</b> : Bloqueio <b>C</b> : Comando executado <b>D</b> : Comando de abertura do dia <b>E</b> : Comando automático de entrada de intervenção <b>S</b> : Comando automático de saída de intervenção

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
char szTabelaLog[271], szTamanhoByteLog[4];  
Retorno = EPSON_Obter_Log_Comandos(szTabelaLog, szTamanhoByteLog);
```

---

**Exemplo em C# :**

```
StringBuilder szTabelaLog = new StringBuilder(271, 271);  
StringBuilder szTamanhoByteLog = new StringBuilder(4, 4);  
Retorno = EPSON_Obter_Log_Comandos(szTabelaLog, szTamanhoByteLog);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTabelaLog, szTamanhoByteLog As String  
szTabelaLog = Space(271)  
szTamanhoByteLog = Space(4)  
Retorno = EPSON_Obter_Log_Comandos(szTabelaLog, szTamanhoByteLog)
```

---

**Exemplo em Delphi 7:**

```
szTabelaLog: array[0..271] of Char;  
szTamanhoByteLog: array[0..4] of Char;  
Retorno := EPSON_Obter_Log_Comandos(szTabelaLog, szTamanhoByteLog);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
SzTabelaLog := AnsiString(StringOfChar(' ', 270));  
szTamanhoByteLog := AnsiString(StringOfChar(' ', 3));  
Retorno := EPSON_Obter_Log_Comandos(PAnsiChar(szTabelaLog),  
PAnsiChar(szTamanhoByteLog));
```

**4.4.65 EPSON\_Obter\_Estado\_ReducacaoZ\_Automatica**

Esta função verifica se a impressão da Redução Z automática está habilitada.

**Sintaxe:**

```
function EPSON_Obter_Estado_ReducacaoZ_Automatica(Var bRZAutomatica:Boolean):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pRZAutomatica	BOOLEAN	-	<b>True</b> - Redução Z automática habilitada. <b>False</b> - Redução Z automática desabilitada.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelos TM-H6000 FBIII, TM-T81 FBIII e TM-T88 FBIII.

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

```
bool bRZAutomatica;  
Retorno = EPSON_Obter_Estado_ReducacaoZ_Automatica( &bRZAutomatica );
```

**Exemplo em C# :**

```
Boolean bRZAutomatica=false;  
Retorno = EPSON_Obter_Estado_ReducacaoZ_Automatica( ref bRZAutomatica );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim bRZAutomatica as Boolean  
Retorno = EPSON_Obter_Estado_ReducacaoZ_Automatica( bRZAutomatica )
```

**Exemplo em Delphi 7:**

```
bRZAutomatica: Boolean;  
Retorno := EPSON_Obter_Estado_ReducacaoZ_Automatica( @bRZAutomatica );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
bRZAutomatica: LongBool;  
Retorno := EPSON_Obter_Estado_ReducacaoZ_Automatica( bRZAutomatica );
```

**4.4.66 EPSON\_Obter\_Valores\_Cupom**

Este comando lê os valores acumulador no cupom fiscal em andamento.

**Sintaxe:**

```
function EPSON_Obter_Valores_Cupom(pszDados:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszDados	PCHAR	280	Dados recebidos da impressora.

Os dados serão retornados no seguinte formato:

Byte 0-17	:	Grande Total (GT)
Byte 18-31	:	Venda Bruta (VB)
Byte 32-44	:	Cancelamento ICMS
Byte 45-57	:	Desconto ICMS
Byte 58-70	:	Total ISSQN
Byte 71-83	:	Cancelamento ISSQN
Byte 84-96	:	Desconto ISSQN
Byte 97-109	:	Total ICMS
Byte 110-122	:	Acréscimo ICMS
Byte 123-135	:	Acréscimo ISSQN
Byte 136-148	:	Total Não Fiscal
Byte 149-161	:	Cancelamento Não Fiscal
Byte 162-174	:	Desconto Não Fiscal
Byte 175-187	:	Acréscimo Não Fiscal
Byte 188-200	:	Troco
Byte 201-213	:	Substituição Tributária ICMS
Byte 214-226	:	Isento ICMS
Byte 227-239	:	Não Incidência ICMS
Byte 240-252	:	Substituição Tributária ISSQN
Byte 253-265	:	Isento ISSQN
Byte 266-278	:	Não Incidência ISSQN

Todos os campos têm tamanho fixo.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

Nenhum.

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

```
char szDados[280];  
Retorno = EPSON_Obter_Valores_Cupom ( szDados );
```

**Exemplo em C# :**

```
StringBuilder szDados = new StringBuilder(280, 280);  
Retorno = EPSON_Obter_Valores_Cupom( szDados );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDados As String  
szDados = Space(280)  
Retorno = EPSON_Obter_Valores_Cupom( szDados )
```

**Exemplo em Delphi 7:**

```
szDados: array[0..280] of Char;  
Retorno := EPSON_Obter_Valores_Cupom( szDados );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDados := AnsiString(StringOfChar(' ', 280));  
Retorno := EPSON_Obter_Valores_Cupom(PAnsiChar(szDados));
```

## 4.4.67 EPSON\_ESC\_ECF\_Obter\_Dados

Este comando lê os valores dos contadores, totalizadores e demais parâmetros do ECF, seguindo o protocolo ESC-ECF.

### Sintaxe:

```
function EPSON_ESC_ECF_Obter_Dados(szGrupo:PChar; szIndice:PChar; szDados:PChar):Integer;StdCall;
External 'InterfaceEpson.dll';
```

### Entradas:

Variável	Tipo	Tam.	Descrição
szGrupo	PCHAR	2 (N)	Número do grupo que será utilizado.
szIndice	PCHAR	2 (N)	Índice da informação que será recuperada.

OS grupos e índices disponíveis são mostrados na tabela abaixo:

Grupo		Índice		Resposta
Número	Descrição	Número	Descrição	
1	Contadores Fixos	0	Todo o grupo	índice valor  . . . índice valor
		1	COO	índice valor
		2	GNF	índice valor
		3	CRO	índice valor
		4	CRZ	índice valor
		5	CCF	índice valor
		7	CFD	índice valor
		8	CCD	índice valor
		9	GRG	índice valor
		10	NFC	índice valor
		11	CFC	índice valor
		14	NCN	índice valor
		15	RZ restantes	índice valor
2	Contador de Relatório Gerencial	0	Todo o grupo	índice valor  . . . índice valor
		1	RG 01	índice valor
		...	...	...
		30	RG 30	índice valor
3	Contador e Totalizador de Operação Não-Fiscal	0	Todo o grupo	índice contador valor  . . . índice contador valor
		1	NF 01	índice contador valor
		...	...	...
		30	NF 30	índice contador valor
4	Totalizadores Gerais	0	Todo o grupo	índice valor  . . . índice valor
		1	GT	índice valor
		2	VB	índice valor
		3	Can-T	índice valor
		4	DT	índice valor
		5	Can-S	índice valor
		6	DS	índice valor
		7	Venda Líquida	índice valor
		8	AT	índice valor

		9	AS	índice valor
5	Totalizadores ICMS/ISS	0	Todo o grupo	índice tipo% valor  . . . índice tipo% valor
		1	T01	índice tipo% valor
		...	...	...
		30	T30	índice tipo% valor
		31	S01	índice tipo% valor
		...	...	...
		60	S30	índice tipo% valor
6	Totalizadores F, I, N	0	Todo o grupo	tipo valor ..... tipo valor
		1	F1, I1, N1, FS1, IS1, NS1	tipo valor ..... tipo valor
		2	F2, I2, N2, FS2, IS2, NS2	tipo valor ..... tipo valor
		3	F3, I3, N3, FS3, IS3, NS3	tipo valor ..... tipo valor
7	Totalizadores de Meios de Pagamento	0	Todo o grupo	Índice valor  . . . Índice valor
		1	PAG 01	Índice valor
		...	...	...
		20	PAG 20	Índice valor
		21	Troco	Índice valor
8	Status do Movimento	-	-	ddmmaaaa status COO Gti  data do movimento, Status, COO inicial e GT inicial status = 0 - Não houve movimento status = 1 - Com movimento aberto status = 2 - Redução Z Pendente
9	Relógio	-	-	ddmmaaaahhmmssf  ddmmaaaa – data do relógio hhmmss – horário do relógio f – flag de verão (V= horário de verão)
10	Tempo Emitindo Doc. Fiscal/Tempo Operacional	-	-	hhmmss hhmmss
11	Tabela de Alíquotas	0	Todas as cadastradas	Índice tipo alíquota  . . . Índice tipo alíquota
		1	% T 01	Índice tipo alíquota
		...	...	...
		30	% T 30	Índice tipo alíquota
		31	% S 01	Índice tipo alíquota
		...	...	...
		60	% S 30	Índice tipo alíquota
12	Tabela de Operações Não-Fiscais	0	Todas as cadastradas	Índice nome  . . . Índice nome
		1	NF 01	Índice nome
		...	...	...
		30	NF 30	Índice nome
13	Tabela de Relatórios Gerenciais	0	Todos os cadastrados	Índice nome  . . . Índice nome
		1	RG 01	Índice nome
		...	...	...
		30	RG 30	Índice nome
14	Tabela de Meios de Pagamento	0	Todos os cadastrados	Índice nome CCD  . . . Índice nome CCD
		1	NomePg 01	Índice nome CCD
		...	...	...
		20	NomePg 20	Índice nome CCD
15	Parâmetros	0	Todo o grupo	Todos os campos separados por pipe “ ”
		1	Marca	Marca do ECF



		2	Modelo	Modelo do ECF
		3	Tipo ECF	Tipo do ECF
		4	NFab	Número de fabricação
		5	NOS	Número do ECF na Loja
		8	CNPJ	CNPJ do usuário
		9	IE	I.E. do usuário
		10	IM	I.M. do usuário
		11	Moeda	Moeda
		12	Decimais Valor	Casas decimais do valor unitário
		13	Decimais Quant.	Casas decimais da quantidade
		14	Versão SB	Versão atual software básico ( XX.XX.XX )
		15	Razão Social	Razão Social do usuário
		16	Nome Fantasia	Nome Fantasia do usuário
		17	Endereço	Endereço do usuário
		18	Criptografia GT	Criptografia do GT
		20	Versão Esc-ECF	"XX.XX"
		21	UF	Sigla da UF
		22	MUN	Sigla do Município
		23	Modo	0 = ECF para comércio
		24	nPrest	Número de Prestadores Cadastrados
		25	nPrestAtv	Número de Prestadores Habilitados
16	Status	1	Gaveta	0 - Fechada 1 - Aberta
		2	Bobina de Papel	0 - Ok 1 - Pouco Papel 2 - Sem Papel
		3	Tampa	0 - OK 1 - Aberta
		4	Modo de Funcionamento	0 - MO 1 - MIT 2 - MIL 3 - MBT 4 - MBD 5 - MNI 6 - Qualquer modo, em situação de erro
		5	Contexto	0 - Repouso 10 - CF Aberto 11 - CF Sub-totalizado 12 - CF Em pagamento 13 - CF Pago e Não finalizado 20 - CNF Aberto 21 - CNF Sub-totalizado 22 - CNF em Pagamento 23 - CNF Pago e Não Finalizado 30 - CCD Aberto 31 - Estorno de CCD Aberto 32 - Relatório Gerencial
17	Dados da RZ gravados na MF	N	CRZ	CRZ
				Data Movimento (ddmmaaaa)
				Data da gravação (ddmmaaaahhmmssf)
				CRO
				COO inicial ( RZ anterior + 1 )
				COO da RZ
				Número do Usuário

				VB
				Desconto ICMS
				Acréscimo ICMS
				Cancelamentos ICMS
				Desconto ISSQN
				Acréscimo ISSQN
				Cancelamento ISSQN
				Total acumulador de Não Fiscais
				Quantidade de acumuladores fiscais parciais
				Tipo acumulador  Alíquota  Base de cálculo  ( se repete para cada acumulador fiscal )
				Versão do Software básico ( XX.XX.XX )
				CNPJ
				I.E.
				I.M.
19	Configuração de Acesso Remoto	1	Montagem de modem GPRS	0 = ECF sem modem 1 = ECF com modem
		2	Atendimento de chamada	0 = Não atende 1 = Atende
20	Chave Pública N = Módulo D = expoente	0	Todas	N   D   N   D
		1	Chave Software Básico	N   D
		2	Chave Arquivos Eletrônicos	N   D
		3	Chave Documentos emitidos	N   D
		4	Chave autenticação fisco	N   D   (pode ser uma lista com várias chaves)
		5	Chave MIL	N   D
21	Parâmetros Adicionais	0	Todo o grupo	Todos os campos separados por pipe " "
		1	COO restante	Número de documentos que ainda podem ser emitidos 999999999 se não houver limitação
		2	URL	Endereço eletrônico programado para transmissão eletrônica de arquivos

**Saídas:**

Variável	Tipo	Tam.	Descrição
szDados	PCHAR	-	Informação recuperada.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

---

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

```
char szContadores[1024];  
Retorno = EPSON_ESC_ECF_Obter_Dados( "1", "1", szContadores);
```

---

**Exemplo em C#:**

```
StringBuilder szContadores = new StringBuilder(1024, 1024);  
Retorno = EPSON_ESC_ECF_Obter_Dados( "1", "1", szContadores);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szContadores As String  
szContadores = Space(1024)  
Retorno = EPSON_ESC_ECF_Obter_Dados( "1", "1", szContadores)
```

---

**Exemplo em Delphi 7:**

```
szContadores: array[0..1024] of Char;  
Retorno := EPSON_ESC_ECF_Obter_Dados( '1', '1', szContadores);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szContadores := AnsiString(StringOfChar(' ', 1024));  
Retorno := EPSON_ESC_ECF_Obter_Versao_Comandos('1', '1',  
PAnsiChar(szContadores));
```

**4.4.68 EPSON\_ESC\_ECF\_Obter\_Binario\_SB**

Esta função gera um arquivo binário contendo os dados do software básico do ECF, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Obter_Binario_SB(szFileName:PChar):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szFileName	PCHAR	1024 (A)	Nome do arquivo binário que será gerado (Path completo).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Obter_Binario_SB("SBasico.bin");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Obter_Binario_SB("SBasico.bin")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Obter_Binario_SB('SBasico.bin');
```

**4.4.69 EPSON\_ESC\_ECF\_Obter\_Binario\_MF**

Esta função gera um arquivo binário contendo os dados da Memória Fiscal do ECF, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Obter_Binario_MF(szModo:PChar; szInicial:PChar; szFinal:PChar;  
                                         szFileName:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szModo	PCHAR	1 (N)	Modo de geração do arquivo binário. 1 – Por Data. 2 – Por CRZ. 3 – Total.
szInicial	PCHAR	8 (O,N)	Valor Inicial (Data ou CRZ).
szFinal	PCHAR	8 (O,N)	Valor Final (Data ou CRZ).
szFileName	PCHAR	1024 (A)	Nome do arquivo binário que será gerado (Path completo).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Obter_Binario_MF("2", "1", "10", "MF.bin");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Obter_Binario_MF("2", "1", "10", "MF.bin")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Obter_Binario_MF('2', '1', '10', 'MF.bin');
```

**4.4.70 EPSON\_ESC\_ECF\_Obter\_Binario\_MFD**

Esta função gera um arquivo binário contendo os dados da Memória Fita Detalhe do ECF, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Obter_Binario_MFD(szModo:PChar; szInicial:PChar; szFinal:PChar;  
                                          szFileName:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szModo	PCHAR	1 (N)	Modo de geração do arquivo binário. 1 – Por Data. 2 – Por CRZ. 3 – Por COO. 4 – Total.
szInicial	PCHAR	8 (O,N)	Valor Inicial (Data ou CRZ ou COO).
szFinal	PCHAR	8 (O,N)	Valor Final (Data ou CRZ ou COO).
szFileName	PCHAR	1024 (A)	Nome do arquivo binário que será gerado (Path completo).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Obter_Binario_MFD("2", "1", "10", "MFD.bin");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Obter_Binario_MFD("2", "1", "10", "MFD.bin")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Obter_Binario_MFD('2', '1', '10', 'MFD.bin');
```

**4.4.71 EPSON\_ESC\_ECF\_Obter\_Binario\_TDM**

Esta função gera um arquivo binário contendo os dados de todas as memórias do ECF (Memória Fiscal, Memória de Fita detalhe e Memória de Trabalho), seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Obter_Binario_TDM(szModo:PChar; szInicial:PChar; szFinal:PChar;  
                                          szFileName:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szModo	PCHAR	1 (N)	Modo de geração do arquivo binário. 1 – Por Data. 2 – Por CRZ. 3 – Total.
szInicial	PCHAR	8 (O,N)	Valor Inicial (Data ou CRZ).
szFinal	PCHAR	8 (O,N)	Valor Final (Data ou CRZ).
szFileName	PCHAR	1024 (A)	Nome do arquivo binário que será gerado (Path completo).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Obter_Binario_TDM("2", "1", "10", "TDM.bin");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Obter_Binario_TDM("2", "1", "10", "TDM.bin")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Obter_Binario_TDM('2', '1', '10', 'TDM.bin');
```

**4.4.72 EPSON\_ESC\_ECF\_Obter\_Versao\_Comandos**

Esta função verifica a versão da Especificação de Comandos implementada no ECF, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Obter_Versao_Comandos( szNumeroECF:PChar; szVersao:PChar;  
                                              szIdentificacao:PChar):Integer; StdCall; External  
                                              'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szNumeroECF	PCHAR	3 (O,N)	Número do ECF.

**Saídas:**

Variável	Tipo	Tam.	Descrição
szVersao	PCHAR	5 (N)	Versão implementada.
szIdentificacao	PCHAR	3 (A)	Código do fabricante.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szVersao[5], szIdentificacao[3];  
Retorno = EPSON_ESC_ECF_Obter_Versao_Comandos( "", szVersao,  
szIdentificacao);
```

**Exemplo em C#:**

```
StringBuilder szVersao = new StringBuilder(5, 5);  
StringBuilder szIdentificacao = new StringBuilder(3, 3);  
Retorno = EPSON_ESC_ECF_Obter_Versao_Comandos( "", szVersao,  
szIdentificacao);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szVersao As String  
Dim szIdentificacao As String  
szVersao = Space(5)  
szIdentificacao = Space(3)  
Retorno = EPSON_ESC_ECF_Obter_Versao_Comandos( "", szVersao,  
szIdentificacao)
```



---

**Exemplo em Delphi 7:**

```
szVersao: array[0..5] of Char;  
szIdentificacao: array[0..3] of Char;  
Retorno := EPSON_ESC_ECF_Obter_Versao_Comandos( '', szVersao,  
szIdentificacao);
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szVersao := AnsiString(StringOfChar(' ', 5));  
szIdentificacao := AnsiString(StringOfChar(' ', 3));  
Retorno := EPSON_ESC_ECF_Obter_Versao_Comandos(' ', PAnsiChar(szVersao),  
PAnsiChar(szIdentificacao));
```

**4.4.73 EPSON\_ESC\_ECF\_Obter\_XML\_Cupom\_Fiscal**

Esta função efetua a leitura do XML de um ou mais cupons fiscais eletrônicos.

**Sintaxe:**

```
function EPSON_ESC_ECF_Obter_XML_Cupom_Fiscal(szModo:PChar; szInicial:PChar; szFinal:PChar;  
szFileName:PChar):Integer; StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szModo	PCHAR	1 (N)	Modo de geração do arquivo binário. 1 – Por Data de movimento. 2 – Por COO. 3 – Por CCF. 4 – Por chave de busca.
szInicial	PCHAR	8 (O,N)	Valor Inicial (Data, COO, CCF ou Chave de busca).
szFinal	PCHAR	8 (O,N)	Valor Final (Data, COO, CCF ou Chave de busca).
szFileName	PCHAR	1024 (A)	Nome do arquivo binário que será gerado (Path completo).

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Obter_XML_Cupom_Fiscal("3", "1", "10",  
"Cupons.xml");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Obter_XML_Cupom_Fiscal("3", "1", "10",  
"Cupons.xml")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Obter_XML_Cupom_Fiscal('3', '1', '10',  
'Cupons.xml');
```

**4.4.74 EPSON\_ESC\_ECF\_Obter\_Tabela\_NaoFiscais**

Esta função efetua a leitura da tabela dos totalizadores não fiscais cadastrados no ECF, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Obter_Tabela_NaoFiscais(pszTabelaNaoFiscais:PChar):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszTabelaNaoFiscais	PCHAR	1080 (A)	Buffer contendo a tabela dos totalizadores não fiscais.

Os dados serão retornados no seguinte formato:

Byte 0-1	:	Número da posição.
Byte 2-5	:	Valor do contador específico do totalizador não fiscal
Byte 6-19	:	Valor acumulado
Byte 20-34	:	Descrição
Byte 35	:	Tipo do totalizador (E – Entrada, S – Saída)
...		

O buffer retornado na variável pszTabelaNaoFiscais, é composto por 30 sequências de 36 caracteres cada. Cada sequência corresponderá a uma das 30 operações não fiscais possíveis de serem cadastradas no ECF, totalizando 1080 bytes a serem retornados.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szTabelaNaoFiscais[1081];  
Retorno = EPSON_ESC_ECF_Obter_Tabela_NaoFiscais( szTabelaNaoFiscais );
```

**Exemplo em C# :**

```
StringBuilder szTabelaNaoFiscais = new StringBuilder(1081, 1081);  
Retorno = EPSON_ESC_ECF_Obter_Tabela_NaoFiscais( szTabelaNaoFiscais );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szTabelaNaoFiscais As String
szTabelaNaoFiscais = Space(1081)
Retorno = EPSON_ESC_ECF_Obter_Tabela_NaoFiscais( szTabelaNaoFiscais )
```

---

**Exemplo em Delphi 7:**

```
szTabelaNaoFiscais: array[0..1081] of Char;
Retorno := EPSON_ESC_ECF_Obter_Tabela_NaoFiscais( szTabelaNaoFiscais );
```

---

**Exemplo em Delphi 2010 / Delphi XE:**

```
szTabelaNaoFiscais := AnsiString(StringOfChar(' ', 1081));
Retorno := EPSON_ESC_ECF_Obter_Tabela_NaoFiscais( PAnsiChar(
szTabelaNaoFiscais ));
```

**4.4.75 EPSON\_ESC\_ECF\_Obter\_Chave\_Busca\_CFE**

Esta função efetua a leitura da chave de busca do CF-e, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Obter_Chave_Busca_CFE (szCOO:PChar, pszChave:PChar):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCOO	PCHAR	9 (A)	COO do cupom que será buscado.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszChave	PCHAR	45 (A)	Chave de busca do CF-e.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

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

```
char szChave[45];  
Retorno = EPSON_ESC_ECF_Obter_Chave_Busca_CFE( "10", szChave );
```

**Exemplo em C# :**

```
StringBuilder szChave = new StringBuilder(45, 45);  
Retorno = EPSON_ESC_ECF_Obter_Chave_Busca_CFE( "10", szChave );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szChave As String  
szChave = Space(45)  
Retorno = EPSON_ESC_ECF_Obter_Chave_Busca_CFE( "10", szChave )
```

**Exemplo em Delphi 7:**

```
szChave: array[0..45] of Char;  
Retorno := EPSON_ESC_ECF_Obter_Chave_Busca_CFE( '10', szChave );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szChave := AnsiString(StringOfChar(' ', 45));  
Retorno := EPSON_ESC_ECF_Obter_Chave_Busca_CFE( '10', PAnsiChar(szChave));
```

## 4.5 Grupo de Configuração da Impressora

Estas funções são usadas para configurar a impressora:

### 4.5.1 EPSON\_Config\_Aliquota

Esta função adiciona alíquotas ISS e ICMS.

Atenção: as alíquotas somente poderão ser apagadas em intervenção técnica.

#### Sintaxe:

```
function EPSON_Config_Aliquota(pszValorAliquota:PChar; bTipoAliquota:Boolean):Integer;StdCall; External  
'InterfaceEpson.dll';
```

#### Entradas:

Variável	Tipo	Tam.	Descrição
pszValorAliquota	PCHAR	4 (N)	Taxa. Ex.: Para adicionar uma taxa de 5,20%, o argumento é "520"
bTipoAliquota	BOOLEAN	-	<b>True</b> - para adicionar alíquota ISS. <b>False</b> - para adicionar alíquota ICMS.

#### Saídas:

Nenhum.

#### Retornos:

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

#### Requisitos:

Os documentos devem estar fechados e o ECF com papel. Caso o ECF não possua inscrição municipal cadastrada, não será possível incluir alíquotas ISS.

---

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

```
Retorno = EPSON_Config_Aliquota ( "520", false );
```

---

**Exemplo em C#:**

```
Retorno = EPSON_Config_Aliquota ( "520", false );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Aliquota ( "520", false )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Aliquota ( '520', false );
```

## 4.5.2 EPSON\_Config\_Relatorio\_Gerencial

Esta função adiciona relatório gerencial.

Atenção: relatórios gerenciais só podem ser removidos em intervenção técnica.

### Sintaxe:

```
function EPSON_Config_Relatorio_Gerencial( pszDescricaoRelatorio:Pchar ):Integer;StdCall; External  
'InterfaceEpson.dll';
```

### Entradas:

Variável	Tipo	Tam.	Descrição
pszDescricaoRelatorio	PCHAR	15 (RT)	Descrição do relatório gerencial.

### Saídas:

Nenhum.

### Retornos:

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

### Requisitos:

Os documentos devem estar fechados e o ECF com papel.

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



```
Retorno = EPSON_Config_Relatorio_Gerencial ( "Relatório 1" );
```

### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Config_Relatorio_Gerencial ( "Relatório 1" )
```

### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Config_Relatorio_Gerencial ( 'Relatorio 1' );
```



### 4.5.3 EPSON\_Config\_Forma\_Pagamento

Esta função adiciona um meio de pagamento. Veja também o comando EPSON\_Config\_Forma\_PagamentoEX.  
Atenção: formas de pagamento somente podem ser removidas em intervenção técnica.

#### Sintaxe:

```
function EPSON_Config_Forma_Pagamento(bTipoVinculado:Boolean; pszNumeroMeio:PChar;  
                                       pszDescricao:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

#### Entradas:

Variável	Tipo	Tam.	Descrição
bTipoVinculo	BOOLEAN	-	<b>True</b> – Pag. vinculado. <b>False</b> – Pag. não-vinculado.
pszNumeroMeio	PCHAR	2 (N)	Número do meio de pagamento.
pszDescricao	PCHAR	15 (RT)	Descrição do meio de pagamento.

#### Saídas:

Nenhum.

#### Retornos:

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

#### Requisitos:

Os documentos devem estar fechados e o ECF com papel.

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



```
Retorno = EPSON_Config_Forma_Pagamento ( false, "3", "Ticket" );
```

#### Exemplo em C# :



```
Retorno = EPSON_Config_Forma_Pagamento ( false, "3", "Ticket" );
```

#### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Config_Forma_Pagamento ( false, "3", "Ticket" )
```

#### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Config_Forma_Pagamento ( false, '3', 'Ticket' );
```

#### 4.5.4 EPSON\_Config\_Forma\_PagamentoEX

Esta função adiciona um meio de pagamento na primeira posição vaga na tabela de meios de pagamento.

Atenção: formas de pagamento somente podem ser removidas em intervenção técnica.

##### Sintaxe:

```
function EPSON_Config_Forma_PagamentoEX( bTipoVinculo:Boolean; pszDescricaoPagamento:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

##### Entradas:

Variável	Tipo	Tam.	Descrição
bTipoVinculo	BOOLEAN	-	<b>True</b> – Pag. vinculado. <b>False</b> – Pag. não-vinculado.
pszDescricaoPagamento	PCHAR	15 (RT)	Descrição do meio de pagamento.

##### Saídas:

Nenhum.

##### Retornos:

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

##### Requisitos:

Os documentos devem estar fechados e o ECF com papel.

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



```
Retorno = EPSON_Config_Forma_PagamentoEX( false, "Ticket" );
```

##### Exemplo em C# :



```
Retorno = EPSON_Config_Forma_PagamentoEX( false, "Ticket" );
```

##### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Config_Forma_PagamentoEX( false, "Ticket" )
```

##### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Config_Forma_PagamentoEX( false, 'Ticket' );
```

### 4.5.5 EPSON\_Config\_Totalizador\_NaoFiscal

Adiciona um novo Totalizador Não-Fiscal.

Atenção: totalizadores adicionados no ECF somente poderão ser apagados em intervenção técnica.

#### Sintaxe:

```
function EPSON_Config_Totalizador_NaoFiscal(pszDescricao:PChar; Var  
                                           pdwNumeroTotalizador:Integer):Integer; StdCall; External  
                                           'InterfaceEpson.dll';
```

#### Entradas:

Variável	Tipo	Tam.	Descrição
pszDescricaoTotalizador	PCHAR	15	Descrição do totalizador.

#### Saídas:

Variável	Tipo	Tam.	Descrição
pdwNumeroTotalizador	INTEGER	-	Número do totalizador.

#### Retornos:

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

#### Requisitos:

Os documentos devem estar fechados e o ECF com papel.

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



```
int dwNumeroTotalizador;  
Retorno=EPSON_Config_Totalizador_NaoFiscal("Tot 1",&dwNumeroTotalizador );
```

#### Exemplo em C# :



```
Int16 dwNumeroTotalizador;  
Retorno=EPSON_Config_Totalizador_NaoFiscal("Tot 1",ref  
dwNumeroTotalizador);
```

#### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Dim dwNumeroTotalizador As Integer  
Retorno=EPSON_Config_Totalizador_NaoFiscal( "Tot 1", dwNumeroTotalizador )
```

#### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
dwNumeroTotalizador: Integer;  
Retorno:=EPSON_Config_Totalizador_NaoFiscal("Tot 1",dwNumeroTotalizador );
```

#### 4.5.6 EPSON\_Config\_Horario\_Verao

Esta função habilita ou desabilita o horário de verão.

**Atenção,** é aconselhado realizar entrada e saída de horário de verão através do comando de Redução Z EPSON\_RelatorioFiscal\_RZ. Utilize a função EPSON\_Config\_Horario\_Verao apenas se isto não for possível.

---

**Sintaxe:**

```
function EPSON_Config_Horario_Verao():Integer;StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel. A Redução Z deve ser o último documento impresso. É aconselhado realizar entrada e saída de horário de verão através do comando de Redução Z EPSON\_RelatorioFiscal\_RZ. Utilize a função EPSON\_Config\_Horario\_Verao apenas se isto não for possível.

---

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

```
Retorno = EPSON_Config_Horario_Verao ( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Horario_Verao ( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Horario_Verao ( );
```

#### 4.5.7 EPSON\_Config\_Espaco\_Entre\_Documentos

Esta função configura o número de linhas entre um documento e outro.

##### Sintaxe:

```
function EPSON_Config_Espaco_Entre_Documentos(pszNumeroLinhas:PChar):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

##### Entradas:

Variável	Tipo	Tam.	Descrição
pszNumeroLinhas	PCHAR	3	Número de linhas.

##### Saídas:

Nenhum.

##### Retornos:

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

##### Requisitos:

Os documentos devem estar fechados.

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



```
Retorno = EPSON_Config_Espaco_Entre_Documentos ( "5" );
```

##### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Config_Espaco_Entre_Documentos ( "5" )
```

##### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Config_Espaco_Entre_Documentos ( '5' );
```

**4.5.8 EPSON\_Config\_Espaco\_Entre\_Linhas**

Esta função configura o espaço entre linhas.

**Sintaxe:**

```
function EPSON_Config_Espaco_Entre_Linhas(pszEspaco:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszEspaco	PCHAR	3	Espaço entre linhas. (Obs.: os valores variam de 0 à 255, sendo que, o espaçamento é de aproximadamente 0,06mm por unidade.)

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados.

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

```
Retorno = EPSON_Config_Espaco_Entre_Linhas ( "5" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Espaco_Entre_Linhas ( "5" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Espaco_Entre_Linhas ( '5' );
```

### 4.5.9 EPSON\_Config\_Habilita\_Logotipo

Esta função habilita ou desabilita a impressão do logotipo no cabeçalho dos documentos impressos

**Sintaxe:**

```
function EPSON_Config_Habilita_Logotipo(bStatusHabilitaLogo:Boolean):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bStatusHabilitaLogo	BOOLEAN	-	<b>True</b> - Imprimir o bitmap do logotipo. <b>False</b> - Não imprimir o bitmap do logotipo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.

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

```
Retorno = EPSON_Config_Habilita_Logotipo ( true );
```

**Exemplo em C#:**

```
Retorno = EPSON_Config_Habilita_Logotipo ( true );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Habilita_Logotipo ( true )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Habilita_Logotipo ( true );
```

**4.5.10 EPSON\_Config\_Logotipo**

Esta função configura o logotipo do cabeçalho do cupom.

**Sintaxe:**

```
function EPSON_Config_Logotipo( pszDadosLogotipo:PChar; dwTamDadosLogotipo:Integer;  
                               pszNumeroLinha:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszDadosLogotipo	PCHAR	2880 (H)	Dados do Logotipo
dwTamDadosLogotipo	INTEGER	-	Tamanho do campo de dados
pszNumeroLinha	PCHAR	1 (N)	Número da linha

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.

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

```
char szDadosLogotipo[]=  
"00000000000000000000000000000000fe0001fc000110000009000000e0000000e00007c00018800  
00f80ff00e0880000880fe0701be0001100ff0b80d90ee0880000d80fc07708200010  
207e0820c30c40810000c11fe07e09200011203e1920c30000810fe0891f006f01000  
00b00ff0ee0ff0000880fe0d801007700600007c0070c807c0e80c401e07c00200704  
40001e60ff1120e019203e08e00700007c0000fb00006f0000010fe00f1820f41020c  
408203e0fc0030000000fa0000fe0001920001120ff00209900009100008100000004  
40ff0e60881120881120f80de02000006207c0f30ce09910208918204e0fe00000003  
c0000c30fe08111008119007e0000000000021800ff1fe07008000e18007f0000480f  
c00000e0000fc0001e000001e0ff03e0811e00810060c703c03e1c80000e80ff01e09  
90020890900911fe0000100001900000fe0000000000fe0001fe00019200011200000  
20000000000000000000000000000000";  
Retorno = EPSON_Config_Logotipo ( szDadosLogotipo, 720, "6" );
```



**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Dim szDadosLogotipo As String
szDadosLogotipo="00000000000000000000fe0001fc0001100000900000e000000e000
07c0001880000f80ff00e0880000880fe0701be0001100ff0b80d90ee0880000d80
fc07708200010207e0820c30c40810000c11fe07e09200011203e1920c30000810f
e0891f006f0100000b00ff0ee0ff0000880fe0d801007700600007c0070c807c0e8
0c401e07c0020070440001e60ff1120e019203e08e00700007c0000fb00006f0000
010fe00f1820f41020c408203e0fc0030000000fa0000fe0001920001120ff00209
90000910000810000000440ff0e60881120881120f80de02000006207c0f30ce099
10208918204e0fe00000003c0000c30fe08111008119007e0000000000021800ff1
fe07008000e18007f0000480fc00000e0000fc0001e000001e0ff03e0811e008100
60c703c03e1c80000e80ff01e0990020890900911fe0000100001900000fe000000
0000fe0001fe000192000112000002000000000000000000000000000000000000"

Retorno = EPSON_Config_Logotipo ( szDadosLogotipo, 720, "1" )
```

**Exemplo em Delphi 7:**

```
szDadosLogotipo: String;
szDadosLogotipo :=
  "000000000000000000000000fe0001fc0001100000900000e000000e00007c000188
  0000f80ff00e0880000880fe0701be0001100ff0b80d90ee0880000d80fc0770820
  0010207e0820c30c40810000c11fe07e09200011203e1920c30000810fe0891f006
  f0100000b00ff0ee0ff0000880fe0d801007700600007c0070c807c0e80c401e07c
  0020070440001e60ff1120e019203e08e00700007c0000fb00006f0000010fe00f1
  820f41020c408203e0fc0030000000fa0000fe0001920001120ff00209900009100
  008100000000440ff0e60881120881120f80de02000006207c0f30ce099102089182
  04e0fe00000003c0000c30fe08111008119007e0000000000021800ff1fe0700800
  0e18007f0000480fc00000e0000fc0001e000001e0ff03e0811e00810060c703c03
  e1c80000e80ff01e0990020890900911fe0000100001900000fe0000000000fe000
  1fe0001920001120000020000000000000000000000000000000000000000000000";

Retorno := EPSON_Config_Logotipo ( szDadosLogotipo, 720, '1' );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
szDadosLogotipo: Ansistring;
szDadosLogotipo :=
  "000000000000000000000000fe0001fc0001100000900000e000000e00007c000188
  0000f80ff00e0880000880fe0701be0001100ff0b80d90ee0880000d80fc0770820
  0010207e0820c30c40810000c11fe07e09200011203e1920c30000810fe0891f006
  f0100000b00ff0ee0ff0000880fe0d801007700600007c0070c807c0e80c401e07c
  0020070440001e60ff1120e019203e08e00700007c0000fb00006f0000010fe00f1
  820f41020c408203e0fc0030000000fa0000fe0001920001120ff00209900009100
  008100000000440ff0e60881120881120f80de02000006207c0f30ce099102089182
  04e0fe00000003c0000c30fe08111008119007e0000000000021800ff1fe0700800
  0e18007f0000480fc00000e0000fc0001e000001e0ff03e0811e00810060c703c03
  e1c80000e80ff01e0990020890900911fe0000100001900000fe0000000000fe000
  1fe0001920001120000020000000000000000000000000000000000000000000000";

Retorno := EPSON_Config_Logotipo (PAnsiChar(szDadosLogotipo), 720, '1' );
```

**4.5.11 EPSON\_Config\_Operador**

Esta função configura o nome do operador.

**Sintaxe:**

```
function EPSON_Config_Operador(pszNomeOperador:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNomeOperador	PCHAR	20 (O,A)	Nome do operador.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.

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

```
Retorno = EPSON_Config_Operador ( "NomeOperador" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Operador ( "NomeOperador" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Operador ( 'NomeOperador' );
```

**4.5.12 EPSON\_Config\_OperadorEX**

Esta função configura o nome do operador, possibilitando ou não a impressão de um relatório gerencial.

**Sintaxe:**

```
function EPSON_Config_OperadorEX(pszNomeOperador:PChar; dwImpressaoRelatorio:Integer):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNomeOperador	PCHAR	20	Nome do operador.
dwImpressaoRelatorio	INTEGER	-	0 - Imprimir o relatório gerencial. 1- Não imprimir o relatório gerencial.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.

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

```
Retorno = EPSON_Config_OperadorEX ( "NomeOperador", 0 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_OperadorEX ( "NomeOperador", 0 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_OperadorEX ( 'NomeOperador', 0 );
```

**4.5.13 EPSON\_Config\_Corte\_Papel**

Esta função configura o corte do papel.

**Sintaxe:**

```
function EPSON_Config_Corte_Papel(bStatusCortePapel:Boolean):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bStatusCortePapel	BOOLEAN	-	<b>True</b> - Habilita o corte de papel. <b>False</b> - Desabilita o corte de papel.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
Retorno = EPSON_Config_Corte_Papel ( true );
```

**Exemplo em C#:**

```
Retorno = EPSON_Config_Corte_Papel ( true );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Corte_Papel ( true )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Corte_Papel ( true );
```

**4.5.14 EPSON\_Config\_Serial\_Impressora**

Esta função configura a velocidade de comunicação da impressora

**Sintaxe:**

```
function EPSON_Config_Serial_Impressora(pszVelocidadeComunicacao:PChar):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszVelocidadeComunicacao	PCHAR	6	String contendo a velocidade de comunicação desejada para a impressora, podendo ser: 9600, 19200, 38400, 57600 ou 115200 bps.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel. Depois de executada a função, desligue e ligue a impressora para que a nova configuração passe a ser utilizada.

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

```
Retorno = EPSON_Config_Serial_Impressora ( "9600" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Serial_Impressora ( "9600" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Serial_Impressora ( '9600' );
```

**4.5.15 EPSON\_Config\_Dados\_Sintegra**

Esta função deverá ser executada para configurar os dados utilizados pela função EPSON\_Obter\_Dados\_MF\_MFD na geração dos registros sintegra 10 e 11.

**Sintaxe:**

```
function EPSON_Config_Dados_Sintegra(pszRazaoSocial:PChar; pszLogradouro:PChar; pszNumero:PChar;  
    pszComplemento:PChar; pszBairro:PChar; pszMunicipio:PChar;  
    pszCEP:PChar; pszUF:PChar; pszFax:PChar; pszFone:PChar;  
    pszNomeContato:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszRazaoSocial	PCHAR	35	Razão social do estabelecimento.
pszLogradouro	PCHAR	34	Logradouro do estabelecimento.
pszNumero	PCHAR	5	Número do endereço do estabelecimento.
pszComplemento	PCHAR	22	Caso existam informações adicionais de endereço deverão ser informadas neste campo.
pszBairro	PCHAR	15	Bairro em que se localiza o estabelecimento.
pszMunicipio	PCHAR	30	Município em que se localiza o estabelecimento.
pszCEP	PCHAR	8	CEP do estabelecimento.
pszUF	PCHAR	2	Estado em que se localiza o estabelecimento.
pszFax	PCHAR	10	Número do fax do estabelecimento.
pszFone	PCHAR	12	Número de telefone do estabelecimento.
pszNomeContato	PCHAR	28	Nome da pessoa responsável à prestar maiores informações.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
Retorno = EPSON_Config_Dados_Sintegra ("Epson Paulista Ltda.", "Av.  
Tucunaré", "720", "", "Tamboré", "Barueri", "06460-020", "SP",  
"1141966250", "1141966250", "Contato");
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Dados_Sintegra ("Epson Paulista Ltda.", "Av.  
Tucunaré", "720", " ", "Tamboré", "Barueri", "06460-020", "SP",  
"1141966250", "1141966250", "Contato")
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Dados_Sintegra ('Epson Paulista Ltda.', 'Av.  
Tucunaré', '720', ' ', 'Tamboré', 'Barueri', '06460-020', 'SP',  
'1141966250', '1141966250', 'Contato');
```

**4.5.16 EPSON\_Config\_Dados\_SPED**

Esta função deverá ser executada para configurar os dados utilizados pela função EPSON\_Obter\_Dados\_MF\_MFD na geração dos registros do SPED Fiscal.

**Sintaxe:**

```
function EPSON_Config_Dados_SPED(CFOP:PChar; AliqPis:PChar; AliqCofins:PChar; CodObsLancFisc:PChar;  
UF:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
CFOP	PCHAR	5	Número CFOP dos produtos comercializados. Códigos mais comuns: 5101 - Venda de produção do estabelecimento 5102 - Venda de mercadoria adquirida ou recebida de terceiros
AliqPis	PCHAR	6	Valor percentual da alíquota PIS, com valor máximo de 99,99
AliqCofins	PCHAR	6	Valor percentual da alíquota COFINS, com valor máximo de 99,99
CodObsLancFisc	PCHAR	7	Número do código da observação do lançamento fiscal
UF	PCHAR	3	Sigla da Unidade Federada com duas letras

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
Retorno = EPSON_Config_Dados_SPED ("5102", "5,00", "7,00", "", "SP");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Dados_SPED ("5102", "5,00", "7,00", "", "SP")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Dados_SPED ('5102', '5,00', '7,00', '', 'SP');
```



**4.5.17 EPSON\_Config\_Habilita\_CAT52\_Auto**

Esta função habilita e desabilita a Geração automática do arquivo da CAT52.

**Sintaxe:**

```
function EPSON_Config_Habilita_CAT52_Auto(dwStatusHabilitaCAT52:Integer):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dw StatusHabilitaCAT52	INTEGER	-	1 - Habilita a geração automática da CAT 52. 0 - Desabilita a geração automática da CAT 52.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível para todas as impressoras fiscais.

**Observação:**

Para a geração correta do arquivo a função de Habilita CAT 52 deve ser ativada com a jornada fiscal fechada, caso contrario somente serão salvos os dados da Jornada fiscal seguinte.

No Sistema Operacional Windows uma vez habilitada ou desabilitada a função sua configuração ficará salva no Registro, não sendo necessário realizar sua habilitação ou desabilitação toda vez que a DLL for carregada.

No Sistema Operacional Linux será necessário habilitar a função de Habilita CAT 52 sempre que a Lib for carregada pelo aplicativo.

Ao se habilitar à função EPSON\_Config\_Habilita\_CAT52\_Auto a função EPSON\_Config\_Habilita\_RJSSER16\_Auto será automaticamente desabilitada.

---

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

```
Retorno = EPSON_Config_Habilita_CAT52_Auto ( 1 );
```

---

**Exemplo em C#:**

```
Retorno = EPSON_Config_Habilita_CAT52_Auto( 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Habilita_CAT52_Auto( 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Habilita_CAT52_Auto( 1 );
```

**4.5.18 EPSON\_Config\_Habilita\_RJSSER16\_Auto**

Esta função habilita e desabilita a Geração automática do arquivo da MFD exigido pela portaria SSER16.

**Sintaxe:**

```
function EPSON_Config_Habilita_RJSSER16_Auto(dwStatusGeracaoArquivo:Integer):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwStatusGeracaoArquivo	INTEGER	-	1 - Habilita a geração automática do arquivo. 0 - Desabilita a geração automática do arquivo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível para todas as impressoras fiscais.

**Observação:**

Para a geração correta do arquivo a função deve ser habilitada com a jornada fiscal fechada, caso contrario somente serão salvos os dados da Jornada fiscal seguinte.

No Sistema Operacional Windows uma vez habilitada ou desabilitada a função sua configuração ficará salva no Registro, não sendo necessário realizar sua habilitação ou desabilitação toda vez que a DLL for carregada.

No Sistema Operacional Linux será necessário habilitar a função sempre que a Lib for carregada pelo aplicativo.

Ao se habilitar à função EPSON\_Config\_Habilita\_RJSSER16\_Auto a função EPSON\_Config\_Habilita\_CAT52\_Auto será automaticamente desabilitada.

---

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

```
Retorno = EPSON_Config_Habilita_RJSSER16_Auto( 1 );
```

---

**Exemplo em C#:**

```
Retorno = EPSON_Config_Habilita_RJSSER16_Auto( 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Habilita_RJSSER16_Auto( 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Habilita_RJSSER16_Auto( 1 );
```

**4.5.19 EPSON\_Config\_Habilita\_Mensagem\_Cupom\_Mania**

Esta função habilita e desabilita a impressão automática da mensagem contendo o total de crédito obtido no cupom. A impressão desta mensagem é exigida pela portaria SSER16.

**Sintaxe:**

```
function EPSON_Config_Habilita_Mensagem_Cupom_Mania(dwStatusImpressaoMensagem:Integer):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwStatusImpressaoMensagem	INTEGER	-	1 - Habilita a impressão automática da mensagem. 0 - Desabilita a impressão automática da mensagem.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível para todas as impressoras fiscais.

**Observação:**

No Sistema Operacional Windows uma vez habilitada ou desabilitada a função sua configuração ficará salva no Registro, não sendo necessário realizar sua habilitação ou desabilitação toda vez que a DLL for carregada.

No Sistema Operacional Linux será necessário habilitar a função sempre que a Lib for carregada pelo aplicativo.

---

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

```
Retorno = EPSON_Config_Habilita_Mensagem_Cupom_Mania( 1 );
```

---

**Exemplo em C#:**

```
Retorno = EPSON_Config_Habilita_Mensagem_Cupom_Mania( 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Habilita_Mensagem_Cupom_Mania( 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

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

**4.5.20 EPSON\_Config\_Habilita\_Mensagem\_Minas\_Legal**

Esta função habilita e desabilita a impressão automática da mensagem Minas Legal.

**Sintaxe:**

```
function EPSON_Config_Habilita_Mensagem_Minas_Legal(dwStatusImpressaoMensagem:Integer):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwStatusImpressaoMensagem	INTEGER	-	1 - Habilita a impressão automática da mensagem. 0 - Desabilita a impressão automática da mensagem.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível para todas as impressoras fiscais.

**Observação:**

No Sistema Operacional Windows uma vez habilitada ou desabilitada a função sua configuração ficará salva no Registro, não sendo necessário realizar sua habilitação ou desabilitação toda vez que a DLL for carregada.

No Sistema Operacional Linux será necessário habilitar a função sempre que a Lib for carregada pelo aplicativo.

---

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

```
Retorno = EPSON_Config_Habilita_Mensagem_Minas_Legal( 1 );
```

---

**Exemplo em C#:**

```
Retorno = EPSON_Config_Habilita_Mensagem_Minas_Legal( 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Habilita_Mensagem_Minas_Legal( 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

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



**4.5.21 EPSON\_Config\_Habilita\_PAFECF\_Auto**

Esta função habilita e desabilita Geração automática do arquivo do PAFECF.

**Sintaxe:**

```
function EPSON_Config_Habilita_PAFECF_Auto(dwStatusGeracaoArquivoPAFECF:Integer):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwStatusGeracaoArquivoPAFECF	INTEGER	-	1 – Habilita a geração automática do arquivo. 0 - Desabilita a geração automática do arquivo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível para todas as impressoras fiscais.

**Observação:**

Para a geração correta do arquivo de registros do PAF-ECF a função EPSON\_Config\_Habilita\_PAFECF\_Auto deve ser ativada com a jornada fiscal fechada, caso contrario somente serão salvos os dados da Jornada fiscal seguinte.

No Sistema Operacional Windows uma vez habilitada ou desabilitada a função sua configuração ficará salva no Registro, não sendo necessário realizar sua habilitação ou desabilitação toda vez que a DLL for carregada.

No Sistema Operacional Linux será necessário habilitar a função de Habilita PAFECF sempre que a Lib for carregada pelo aplicativo.

---

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

```
Retorno = EPSON_Config_Habilita_PAFECF_Auto ( 1 );
```

---

**Exemplo em C#:**

```
Retorno = EPSON_Config_Habilita_PAFECF_Auto ( 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Habilita_PAFECF_Auto ( 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

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

**4.5.22 EPSON\_Config\_Dados\_PAFECF**

Esta função configura dados do Aplicativo da empresa desenvolvedora do Programa Aplicativo Fiscal para serem utilizados na geração do arquivo de registros do PAF-ECF.

**Sintaxe:**

```
function EPSON_Config_Dados_PAFECF( pszDeveloperCNPJ:PChar; pszIE:PChar; pszIM:PChar;  
                                     pszName:PChar; pszNomePAF:PChar; pszVersaoPAF:PChar;  
                                     pszPAFMD5:PChar; pszVersaoER:PChar):Integer;StdCall; External  
                                     'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszCNPJ	PCHAR	14 (A)	CNPJ do desenvolvedor.
pszIE	PCHAR	14 (A)	Inscrição Estadual do desenvolvedor.
pszIM	PCHAR	14 (A)	Inscrição Municipal do desenvolvedor.
pszNome	PCHAR	40 (A)	Denominação do desenvolvedor.
pszNomePAF	PCHAR	40 (A)	Nome do Comercial do PAF.
pszVersaoPAF	PCHAR	10 (A)	Versão Atual do PAF.
pszPAFMD5	PCHAR	32 (A)	Código MD-5 do principal arquivo executável.
pszVersaoER	PCHAR	4 (A)	Versão da Especificação de Requisitos.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
Retorno = EPSON_Config_Dados_PAFECF ("12345678901234", "123.456.789-01",  
"12345678901234", "Nome do Desenvolvedor", "Nome do PAF", "2223334445",  
"0123456789ABCDEF0123456789ABCDEF", "9955");
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Dados_PAFECF ("12345678901234", "123.456.789-01",  
"12345678901234", "Nome do Desenvolvedor", "Nome do PAF", "2223334445",  
"0123456789ABCDEF0123456789ABCDEF", "9955")
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Dados_PAFECF ('123456788901234', '123.456.789-01',  
'12345678901234', 'Nome do Desenvolvedor', 'Nome do PAF', '2223334445',  
'0123456789ABCDEF0123456789ABCDEF', '9955');
```

**4.5.23 EPSON\_Config\_Mensagem\_Aplicacao**

Esta função envia informações da aplicação para serem impressas no rodapé do cupom fiscal.

**Sintaxe:**

```
function EPSON_Config_Mensagem_Aplicacao(pszLinhaTexto01:PChar; pszLinhaTexto02:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszLinhaTexto01	PCHAR	42	Linha de texto 1.
pszLinhaTexto02	PCHAR	42	Linha de texto 2.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A Jornada fiscal deve estar aberta, em período de vendas, cupom fiscal aberto, o ECF com papel e o cupom com os devidos pagamentos já efetuados. Pode-se antes de executar esta função imprimir a mensagem de venda.

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

```
Retorno = EPSON_Config_Mensagem_Aplicacao( "Programa", "Ver 2.3" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Mensagem_Aplicacao( "Programa", "Ver 2.3" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Mensagem_Aplicacao( 'Programa', 'Ver 2.3' );
```

**4.5.24 EPSON\_Config\_Mensagem\_Aplicacao\_Auto**

Esta função configura mensagem da aplicação para ser impressa automaticamente ao fim de cada cupom fiscal e cupom não fiscal.

**Sintaxe:**

```
function EPSON_Config_Mensagem_Aplicacao_Auto(pszLinhaTexto01:PChar; pszLinhaTexto02:PChar):Integer;  
StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszLinhaTexto01	PCHAR	42	Linha de texto 1.
pszLinhaTexto02	PCHAR	42	Linha de texto 2.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

**Observação:**

Para desativar a impressão automática (apenas para modelos diferentes da TM-T800F) deve-se enviar o comando `EPSON_Config_Mensagem_Aplicacao_Auto("", "")`.

Para equipamentos modelo TM-T800F, uma vez configurada a mensagem da aplicação, não será possível desabilitá-la.

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

```
Retorno = EPSON_Config_Mensagem_Aplicacao_Auto( "Programa", "Ver 2.3" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Mensagem_Aplicacao_Auto( "Programa", "Ver 2.3" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Mensagem_Aplicacao_Auto( 'Programa', 'Ver 2.3' );
```

**4.5.25 EPSON\_Config\_Habilita\_Sintegra\_Auto**

Esta função habilita e desabilita a geração automática dos registros Sintegra 60M, 60A, 60I e 60D.

**Sintaxe:**

```
function EPSON_Config_Habilita_Sintegra_Auto(dwStatusHabilitaSintegra:Integer):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwStatusHabilitaSintegra	INTEGER	-	1 - Habilita geração automática do sintegra. 0 - Desabilita geração automática do sintegra.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível para todas as impressoras fiscais.

**Observação:**

Para geração correta do arquivo de registros Sintegra a função EPSON\_Config\_Habilita\_Sintegra\_Auto, deve ser ativada com a jornada fiscal fechada, caso contrario somente serão salvos os dados da jornada fiscal seguinte.

No sistema operacional Windows uma vez habilitada ou desabilitada a função, sua configuração ficará salva no registro, não sendo necessário realizar sua habilitação ou desabilitação toda vez que a DLL for carregada.

No sistema operacional Linux será necessário chamar a função de habilitar a geração do Sintegra sempre que a Lib for carregada pelo aplicativo.

---

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

```
Retorno = EPSON_Config_Habilita_Sintegra_Auto( 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Habilita_Sintegra_Auto ( 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

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



**4.5.26 EPSON\_Config\_Habilita\_EAD**

Esta função habilita e desabilita a assinatura digital EAD nos documentos ATO /COTEPE 17/04 e espelhos da MFD e MF.

**Sintaxe:**

```
function EPSON_Config_Habilita_EAD(bStatusHabilitaEAD:Boolean):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bStatusHabilitaEAD	BOOLEAN	-	<b>True</b> - Habilita EAD. <b>False</b> - Desabilita EAD.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
Retorno = EPSON_Config_Habilita_EAD( false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_Habilita_EAD( false )
```

**Exemplo em Delphi / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_Habilita_EAD( false );
```

**4.5.27 EPSON\_Config\_ReducaoZ\_Automatica**

Esta função configura a impressão automática da Redução Z.

**Sintaxe:**

```
function EPSON_Config_ReducaoZ_Automatica(bStatusHabilitaRZ:Boolean):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
bStatusHabilitaRZ	BOOLEAN	-	<b>True</b> - Habilita a impressão automática da Redução Z. <b>False</b> - Desabilita a impressão automática da Redução Z.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelos TM-H6000 FBIII, TM-T81 FBIII e TM-T88 FBIII.

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

```
Retorno = EPSON_Config_ReducaoZ_Automatica( true );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Config_ReducaoZ_Automatica( true )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Config_ReducaoZ_Automatica( true );
```

**4.5.28 EPSON\_ESC\_ECF\_Config\_Mensagem\_Aplicacao**

Esta função envia informações da aplicação para serem impressas no rodapé do documento fiscal.

**Sintaxe:**

```
function EPSON_ESC_ECF_Config_Mensagem_Aplicacao(szMensagem:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szMensagem	PCHAR	84 (A)	Mensagem que será impressa pelo ECF.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.  
Uma vez configurada a mensagem da aplicação, não será possível desabilitá-la.

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

```
Retorno = EPSON_ESC_ECF_Config_Mensagem_Aplicacao( "Mensagem do PDV" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Config_Mensagem_Aplicacao( "Mensagem do PDV" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Config_Mensagem_Aplicacao( 'Mensagem do PDV' );
```

**4.5.29 EPSON\_ESC\_ECF\_Config\_Horario\_Verao**

Esta função habilita ou desabilita o horário de verão, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Config_Horario_Verao(szModo:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szModo	PCHAR	1 (N)	<b>0</b> - Sai do horário de verão. <b>1</b> - Entra em horário de verão.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel. A Redução Z deve ser o último documento impresso.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Config_Horario_Verao( "0" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Config_Horario_Verao( "0" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Config_Horario_Verao( '0' );
```

**4.5.30 EPSON\_ESC\_ECF\_Config\_Aliquota**

Esta função adiciona alíquotas ISS e ICMS, seguindo o protocolo ESC-ECF.

**Atenção:** as alíquotas somente poderão ser apagadas em intervenção lógica.

**Sintaxe:**

```
function EPSON_ESC_ECF_Config_Aliquota(szIndice:PChar; szTipo:PChar; szValor:PChar):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice na tabela de alíquotas do ECF (1 a 30).
szTipo	PCHAR	1 (A)	Tipo da alíquota. T – Alíquota de ICMS. S – Alíquota de ISSQN.
szValor	PCHAR	4 (N)	Valor da alíquota.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel. Caso o ECF não possua inscrição municipal cadastrada, não será possível incluir alíquotas ISS.

Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Config_Aliquota("1", "T", "1800");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Config_Aliquota("1", "T", "1800")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Config_Aliquota('1', 'T', '1800');
```

**4.5.31 EPSON\_ESC\_ECF\_Config\_Forma\_Pagamento**

Esta função adiciona um meio de pagamento, seguindo o protocolo ESC-ECF.

**Atenção:** formas de pagamento somente podem ser removidas em intervenção lógica.

**Sintaxe:**

```
function EPSON_ESC_ECF_Config_Forma_Pagamento(szIndice:PChar; szNome:PChar;  
                                                szCCD:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice na tabela de formas de pagamento do ECF (2 a 20).
szNome	PCHAR	15 (A)	Nome da forma de pagamento com no mínimo 4 caracteres.
szCCD	PCHAR	1 (N)	Tipo da forma de pagamento. <b>0</b> – Sem CCD. <b>1</b> – Com CCD.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Config_Forma_Pagamento("2", "Cheque", "0" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Config_Forma_Pagamento("2", "Cheque", "0" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Config_Forma_Pagamento('2', 'Cheque', '0' );
```

**4.5.32 EPSON\_ESC\_ECF\_Config\_Totalizador\_NaoFiscal**

Adiciona um novo Totalizador Não-Fiscal, seguindo o protocolo ESC-ECF.

**Atenção:** totalizadores adicionados no ECF somente poderão ser apagados em intervenção lógica.

**Sintaxe:**

```
function EPSON_ESC_ECF_Config_Totalizador_NaoFiscal(szIndice:PChar; szNome:PChar;  
                                                    szTipo:PChar):Integer;StdCall; External  
            'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice na tabela de totalizadores não fiscal do ECF (3 a 30).
szNome	PCHAR	15 (A)	Nome do totalizador não fiscal com no mínimo 4 caracteres.
szTipo	PCHAR	1 (A)	Tipo do totalizador não fiscal. <b>E</b> – Entrada de valor. <b>S</b> – Saída de valor.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Config_Totalizador_NaoFiscal( "3","Boleto","E" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Config_Totalizador_NaoFiscal( "3","Boleto","E" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Config_Totalizador_NaoFiscal( '3','Boleto','E' );
```

**4.5.33 EPSON\_ESC\_ECF\_Config\_Relatorio\_Gerencial**

Esta função adiciona relatório gerencial, seguindo o protocolo ESC-ECF.

**Atenção:** relatórios gerenciais só podem ser removidos em intervenção lógica.

**Sintaxe:**

```
function EPSON_ESC_ECF_Config_Relatorio_Gerencial(szIndice:PChar; szNome:PChar):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szIndice	PCHAR	2 (N)	Índice na tabela de relatórios gerenciais do ECF (2 a 30).
szNome	PCHAR	15 (A)	Nome do relatório gerencial com no mínimo 4 caracteres.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Config_Relatorio_Gerencial( "2", "Config" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Config_Relatorio_Gerencial( "2", "Config" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Config_Relatorio_Gerencial( '2', 'Config' );
```



**4.5.34 EPSON\_ESC\_ECF\_Config\_Loja**

Esta função configura a identificação da loja no ECF, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Config_Loja(szLoja:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szLoja	PCHAR	4 (A)	Identificação da Loja.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Os documentos devem estar fechados e o ECF com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Config_Loja( "0001" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Config_Loja( "0001" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Config_Loja( '0001' );
```

**4.5.35 EPSON\_ESC\_ECF\_Config\_Dados\_Consumidor\_Rodape**

Esta função configura os dados do cliente que serão impressos no final do documento, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Config_Dados_Consumidor_Rodape(szCPFCNPJ:PChar; szNome:PChar;  
szEndereco:PChar):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szCPFCNPJ	PCHAR	14 (O,N)	Número CPF ou CNPJ do cliente.
szNome	PCHAR	30 (O, A)	Razão Social / Nome do cliente.
szEndereco	PCHAR	79 (O, A)	Endereço do cliente.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

O ECF deve estar ligado e com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Config_Dados_Consumidor_Rodape( "11122233344",  
"Jose da Silva", "");
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Config_Dados_Consumidor_Rodape( "11122233344",  
"Jose da Silva", "")
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Config_Dados_Consumidor_Rodape( '11122233344',  
'Jose da Silva', '');
```

**4.5.36 EPSON\_ESC\_ECF\_Config\_Operador**

Esta função configura o nome do operador, seguindo o protocolo ESC-ECF.

**Sintaxe:**

```
function EPSON_ESC_ECF_Config_Operador(szDados:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
szDados	PCHAR	20 (A)	Nome do operador.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

O ECF deve estar ligado e com papel.  
Comando disponível apenas para ECF modelo TM-T800F.

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

```
Retorno = EPSON_ESC_ECF_Config_Operador( "Operador 01" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Config_Operador( "Operador 01" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Config_Operador( 'Operador 01' );
```

## 4.6 Grupo do Cheque

Estas funções são utilizadas para manipulação do cheque:

### 4.6.1 EPSON\_Cheque\_Configurar\_Moeda

Esta função configura o nome da moeda a ser impressa no cheque.

#### Sintaxe:

```
function EPSON_Cheque_Configurar_Moeda( pszTextoMoedaSingular:PChar;  
    pszTextoMoedaPlural:PChar):Integer; StdCall; External 'InterfaceEpson.dll';
```

#### Entradas:

Variável	Tipo	Tam.	Descrição
pszTextoMoedaSingular	PCHAR	20 (RT)	Texto da moeda no singular.
pszTextoMoedaPlural	PCHAR	20 (RT)	Texto da moeda no plural.

#### Saídas:

Nenhum.

#### Retornos:

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

#### Requisitos:

Nenhum.

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



```
Retorno = EPSON_Cheque_Configurar_Moeda ( "real", "reais" );
```

#### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Cheque_Configurar_Moeda ( "real", "reais" )
```

#### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Cheque_Configurar_Moeda ( 'real', 'reais' );
```

**4.6.2 EPSON\_Cheque\_Configurar\_Parametros1**

Esta função configura a posição de impressão do valor do cheque e as duas linhas de descrição.

**Sintaxe:**

```
function EPSON_Cheque_Configurar_Parametros1(pszNumeroRegistro:PChar;  
                                              pszPosicaoValorHorizontal:PChar;  
                                              pszPosicaoValorVertical:PChar;  
                                              pszPosicaoHorizontalDescricao1:PChar;  
                                              pszPosicaoVerticalDescricao1:PChar;  
                                              pszPosicaoHorizontalDescricao2:PChar;  
                                              pszPosicaoVerticalDescricao2:PChar):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroRegistro	PCHAR	2	Número do registro
pszPosicaoValorHorizontal	PCHAR	3	Posicionamento horizontal do valor
pszPosicaoValorVertical	PCHAR	3	Posicionamento vertical do valor
psz PosicaoHorizontalDescricao1	PCHAR	3	Posicionamento horizontal da linha de descrição 1
psz PosicaoVerticalDescricao1	PCHAR	3	Posicionamento vertical da linha de descrição 1
psz PosicaoHorizontalDescricao2	PCHAR	3	Posicionamento horizontal da linha de descrição 2
psz PosicaoVerticalDescricao2	PCHAR	3	Posicionamento vertical da linha de descrição 2

Obs.: Veja modelo abaixo:

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

---

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

```
Retorno = EPSON_Cheque_Configurar_Parametros1 ( "01", "240", "143", "800",  
"182", "800", "205" );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_Configurar_Parametros1 ( "01", "240", "143", "800",  
"182", "800", "205" )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_Configurar_Parametros1 ( '01', '240', '143',  
'800', '182', '800', '205' );
```

**4.6.3 EPSON\_Cheque\_Configurar\_Parametros2**

Esta função configura a posição de impressão do nome, cidade, data e linhas adicionais.

**Sintaxe:**

```
function EPSON_Cheque_Configurar_Parametros2(pszNumeroRegistro:PChar;  
                                             pszPosicaoHorizontalPara:PChar;  
                                             pszPosicaoVerticalPara:PChar;  
                                             pszPosicaoHorizontalCidade:PChar;  
                                             pszPosicaoVerticalCidade:PChar; pszOffsetDia:PChar;  
                                             pszOffsetMes:PChar; pszOffsetAno:PChar;  
                                             pszPosicaoHorizontalLinhaAdicional:PChar;  
                                             pszPosicaoVerticalLinhaAdicional:PChar):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroRegistro	PCHAR	2	Número do Registro.
pszPosicaoHorizontalPara	PCHAR	3	Posição horizontal do campo Para
pszPosicaoVerticalPara	PCHAR	3	Posição vertical do campo Para
pszPosicaoHorizontalCidade	PCHAR	3	Posição horizontal do campo cidade
pszPosicaoVerticalCidade	PCHAR	3	Posição vertical do campo cidade
pszOffsetDia	PCHAR	3	Offset para o campo dia
pszOffsetMes	PCHAR	3	Offset para o campo mês
pszOffsetAno	PCHAR	3	Offset para o campo ano
pszPosicaoHorizontalLinhaAdicional	PCHAR	3	Posição horizontal do campo de linhas adicionais
pszPosicaoVerticalLinhaAdicional	PCHAR	3	Posição vertical do campo de linhas adicionais

Obs.: Veja modelo abaixo:

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

**Modelo que demonstra a área de impressão de cheques (Parâmetros 1 e 2):**

Observe os intervalos dos valores de X (32 à 883) e Y (0 à 510), vale lembrar que o desenho não está em escala.

```
Retorno := EPSON_Cheque_Configurar_Parametros2 ( '01', '800', '276',  
'700', '335', '33', '41', '55', '700', '460' );
```



#### 4.6.4 EPSON\_Cheque\_Imprimir

Esta função imprime o texto nos campos do cheque.

**Sintaxe:**

```
function EPSON_Cheque_Imprimir(pszNumeroRegistro:PChar; pszValorCheque:PChar;  
                               dwNumeroCasasDecimais:Integer; TextoCampoPara:PChar;  
                               pszTextoCampoCidade:PChar; pszTextoCampoDados:PChar;  
                               pszTextoAdicional:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroRegistro	PCHAR	2	Número do registro
pszValorCheque	PCHAR	13	Valor do cheque
dwNumeroCasasDecimais	INTEGER	-	Número de casas decimais utilizadas
pszTextoCampoPara	PCHAR	40	Texto do campo Para
pszTextoCampoCidade	PCHAR	30	Texto do campo Cidade
pszTextoCampoDados	PCHAR	40	Texto do campo Data formato(ddmmaaaa).
pszTextoAdicional	PCHAR	40	Texto adicional

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A impressora deve estar em modo normal de operação e com os documentos fechados.

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

```
Retorno = EPSON_Cheque_Imprimir ( "01", "50000", 2, "", "São Paulo",  
                                   "10032005", "" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_Imprimir ( "01", "50000", 2, "", "São Paulo",  
                                   "10032005", "" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_Imprimir ( '01', '50000', 2, '', 'São Paulo',  
                                   '10032005', '' );
```

#### 4.6.5 EPSON\_Cheque\_ImprimirEX

Esta função imprime o texto nos campos do cheque, utilizando os dados do arquivo poscheque.dat para configuração dos parâmetros de impressão.

**Sintaxe:**

```
function EPSON_Cheque_ImprimirEX(pszNumeroRegistro:PChar; pszValorCheque:PChar;  
dwNumeroCasasDecimais:Integer; pszTextoCampoPara:PChar;  
pszTextoCampoCidade:PChar; pszTextoCampoDados:PChar;  
pszTextoAdicional:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNumeroRegistro	PCHAR	3	Número do banco (deve constar no poscheque.dat)
pszValorCheque	PCHAR	13	Valor do cheque
dwNumeroCasasDecimais	INTEGER	-	Número de casas decimais utilizadas
pszTextoCampoPara	PCHAR	40	Texto do campo Para
pszTextoCampoCidade	PCHAR	30	Texto do campo Cidade
pszTextoCampoDados	PCHAR	40	Texto do campo Data formato(ddmmaaaa).
pszTextoAdicional	PCHAR	40	Texto adicional

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

A impressora deve estar em modo normal de operação e com os documentos fechados.  
Presença do arquivo poscheque.dat no mesmo diretório da dll.

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

```
Retorno = EPSON_Cheque_ImprimirEX ( "745", "50000", 2, "", "São Paulo",  
"10032005", "" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_ImprimirEX ( "745", "50000", 2, "", "São Paulo",  
"10032005", "" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_ImprimirEX ( '745', '50000', 2, '', 'São Paulo',  
'10032005', '' );
```

#### 4.6.6 EPSON\_Cheque\_Preparar\_Endosso

Esta função prepara o cheque para impressão de endosso. Exclusivamente para o modelo TM-H6000, caso o mecanismo de endosso não esteja presente, o endosso será impresso através do mecanismo de cheques.

---

**Sintaxe:**

```
function EPSON_Cheque_Preparar_Endosso():Integer;StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

**Requisitos:**

Modo normal de operação.

---

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

```
Retorno = EPSON_Cheque_Preparar_Endosso ( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_Preparar_Endosso ( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_Preparar_Endosso ( );
```

**4.6.7 EPSON\_Cheque\_Endosso\_Estacao**

Esta função prepara a estação de cheques para impressão de endosso.

**Sintaxe:**

```
function EPSON_Cheque_Endosso_Estacao(pszPosicaoEixoX:PChar;pszPosicaoEixoY:PChar;  
dwOrientacao:Integer): Integer;StdCall;External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszPosicaoEixoX	PCHAR	3	Posição no Eixo X.
pszPosicaoEixoY	PCHAR	3	Posição no Eixo Y.
dwOrientacao	INTEGER	-	0 - Vertical. 1 - Horizontal.

**Saídas:**

Nenhum.

**Retornos:**

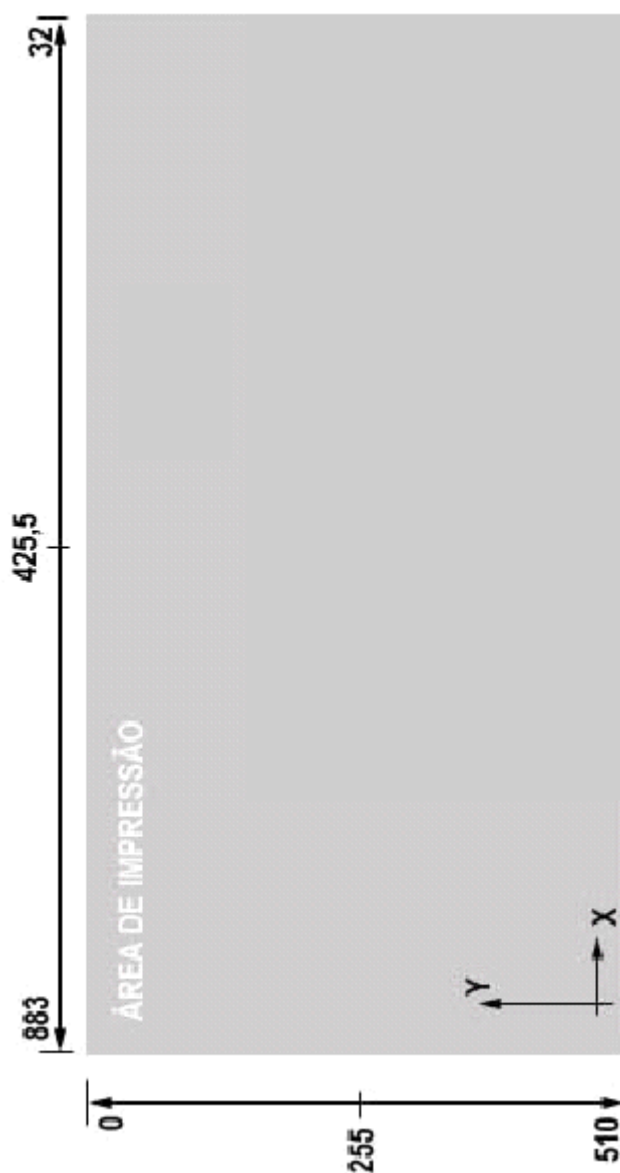
FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

**Requisitos:**

Modo normal de operação.

**Modelo que demonstra a área de impressão (Parâmetros 1 e 2):**

Observe os intervalos dos valores de X (32 à 883) e Y (0 à 510), vale lembrar que o desenho não está em escala.



---

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

```
Retorno = EPSON_Cheque_Endosso_Estacao("50", "0", 1);
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_Endosso_Estacao("50", "0", 1)
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_Endosso_Estacao('50', '0', 1);
```

#### 4.6.8 EPSON\_Cheque\_Imprimir\_Endosso

Esta função imprime uma linha de texto no cheque.

**Sintaxe:**

```
function EPSON_Cheque_Imprimir_Endosso(pszLinhaTexto:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszLinhaTexto	PCHAR	40 (O,RT)	Texto a ser impresso.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Modo normal de operação e durante a execução do comando EPSON\_Cheque\_Preparar\_Endosso

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

```
Retorno = EPSON_Cheque_Imprimir_Endosso ( "Texto" );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_Imprimir_Endosso ( "Texto" )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_Imprimir_Endosso ( 'Texto' );
```

#### 4.6.9 EPSON\_Cheque\_Ejetar\_Endosso

Esta função ejeta o cheque da impressora.

---

**Sintaxe:**

```
function EPSON_Cheque_Ejetar_Endosso():Integer;StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

**Requisitos:**

Modo normal de operação e enquanto o papel estiver inserido na maquina.

---

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

```
Retorno = EPSON_Cheque_Ejetar_Endosso ( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_Ejetar_Endosso ( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_Ejetar_Endosso ( );
```

**4.6.10 EPSON\_Cheque\_Cancelar\_Impressao**

Esta função cancela a impressão de cheque ou endosso.

**Sintaxe:**

```
function EPSON_Cheque_Cancelar_Impressao():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Modo normal de operação e durante a impressão do cheque ou endosso.

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

```
Retorno = EPSON_Cheque_Cancelar_Impressao ( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_Cancelar_Impressao ( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_Cancelar_Impressao ( );
```



**4.6.11 EPSON\_Cheque\_Endosso\_EstacaoEX**

Esta função imprime o texto de endosso utilizando a unidade de Impressão de cheques.

**Sintaxe:**

```
function EPSON_Cheque_Endosso_EstacaoEX(pszPosicaoEixoX:PChar;pszPosicaoEixoY:PChar;  
dwOrientacao: Integer;pszTexto:PChar):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszPosicaoEixoX	PCHAR	3	Posição no Eixo X.
pszPosicaoEixoY	PCHAR	3	Posição no Eixo Y.
dwOrientacao	INTEGER	-	<b>0</b> – Vertical. <b>1</b> – Horizontal.
pszTexto	PCHAR	40	Texto

**Saídas:**

Nenhum.

**Retornos:**

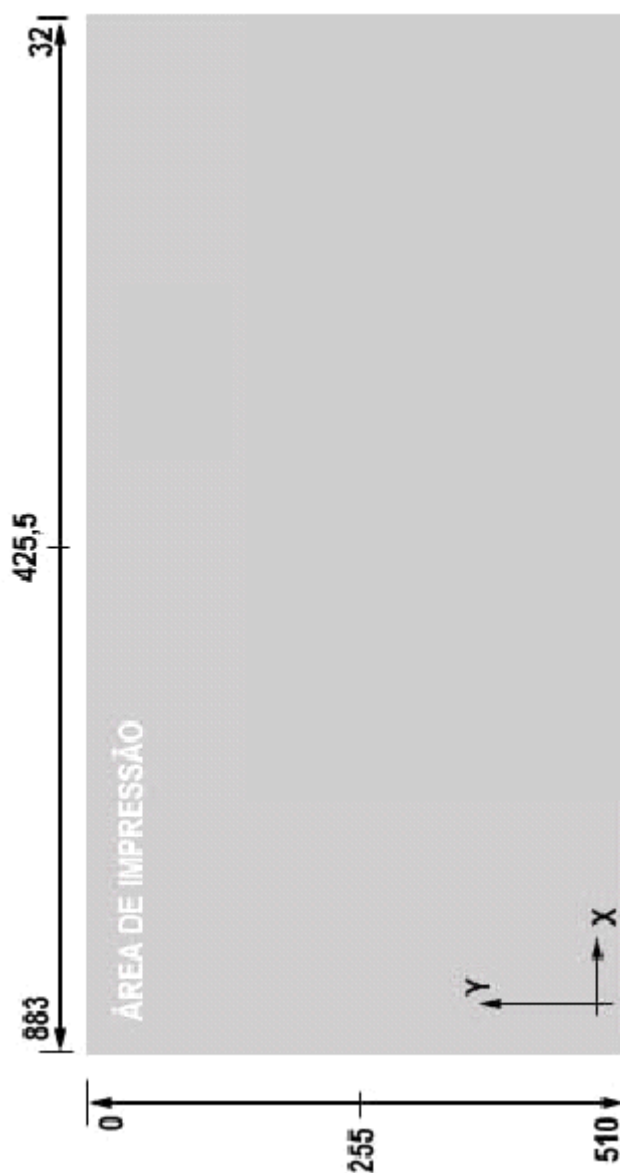
FUNC_SUCESSO	Operação realizada com sucesso.
FUNC_ERRO	Erro durante a execução.

**Requisitos:**

Modo normal de operação.

**Modelo que demonstra a área de impressão (Parâmetros 1 e 2):**

Observe os intervalos dos valores de X (32 à 883) e Y (0 à 510), vale lembrar que o desenho não está em escala.



---

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

```
Retorno = EPSON_Cheque_Endosso_EstacaoEX ("50", "0", 1, "Texto");
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_Endosso_EstacaoEX ("50", "0", 1, "Texto");
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_Endosso_EstacaoEX ('50', '0', 1, 'Texto');
```

**4.6.12 EPSON\_Cheque\_Endosso\_EstacaoCFG**

Esta função imprime o texto de endosso utilizando a unidade de Impressão de cheques.

**Sintaxe:**

```
function EPSON_Cheque_Endosso_EstacaoCFG(pszPosicaoEixoX:PChar; pszPosicaoEixoY:PChar;  
dwOrientacao:Integer; pszTexto:PChar;  
bStatusEjeta:Boolean):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszPosicaoEixoX	PCHAR	3	Posição no Eixo X.
pszPosicaoEixoY	PCHAR	3	Posição no Eixo Y.
dwOrientacao	INTEGER	-	0 – Vertical. 1 – Horizontal.
pszTexto	PCHAR	40	Texto
bStatusEjeta	BOOLEAN	-	<b>True</b> - Ejeta papel para cima. <b>False</b> - Ejeta papel para baixo.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Modo normal de operação.

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

```
Retorno = EPSON_Cheque_Endosso_EstacaoCFG("50", "0", 1, "Texto", true);
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Cheque_Endosso_EstacaoCFG("50", "0", 1, "Texto", true)
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Cheque_Endosso_EstacaoCFG('50', '0', 1, 'Texto', true);
```

**4.6.13 EPSON\_Cheque\_Ler\_MICR**

Esta função lê o MICR do cheque.

**Sintaxe:**

```
function EPSON_Cheque_Ler_MICR(pszValorMICRLido:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Variável	Tipo	Tam.	Descrição
pszValorMICRLido	PCHAR	257	Valor do MICR lido com no máximo 256 bytes.


**Retornos:**

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


**Requisitos:**

Nenhum.


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

```
 char szValorMICRLido[257];  
Retorno = EPSON_Cheque_Ler_MICR ( szValorMICRLido );
```


**Exemplo em C# :**

```
 StringBuilder szValorMICRLido = new StringBuilder(257, 257);  
Retorno = EPSON_Cheque_Ler_MICR ( szValorMICRLido );
```


**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
 Dim szValorMICRLido As String  
szValorMICRLido = Space(257)  
Retorno = EPSON_Cheque_Ler_MICR ( szValorMICRLido )
```

**Exemplo em Delphi 7:**

```
 szValorMICRLido: array[0..257] of Char;  
Retorno := EPSON_Cheque_Ler_MICR ( szValorMICRLido );
```

**Exemplo em Delphi 2010 / Delphi XE:**

```
 szValorMICRLido := AnsiString(StringOfChar(' ', 257));  
Retorno := EPSON_Cheque_Ler_MICR ( PAnsiChar(szValorMICRLido));
```

## 4.7 Grupo da Impressora

Estas funções são utilizadas para comandar o mecanismo da impressora:

### 4.7.1 EPSON\_Impressora\_Abrir\_Gaveta

Esta função abre a gaveta de dinheiro.

---

**Sintaxe:**

```
function EPSON_Impressora_Abrir_Gaveta():Integer;StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

---

**Requisitos:**

Nenhum.

---

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

```
Retorno = EPSON_Impressora_Abrir_Gaveta ( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Impressora_Abrir_Gaveta ( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Impressora_Abrir_Gaveta ( );
```

### 4.7.2 EPSON\_Impressora\_Cortar\_Papel

Esta função corta o papel.

---

**Sintaxe:**

```
function EPSON_Impressora_Cortar_Papel():Integer;StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

---

**Requisitos:**

Documentos fechados.

---

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

```
Retorno = EPSON_Impressora_Cortar_Papel ( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Impressora_Cortar_Papel ( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Impressora_Cortar_Papel ( );
```

### 4.7.3 EPSON\_Impressora\_Avancar\_Papel

Esta função avança o papel.

**Sintaxe:**

```
function EPSON_Impressora_Avancar_Papel(dwNumeroLinhas:Integer):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwNumeroLinhas	INTEGER	-	Número de linhas para avanço de papel

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
Retorno = EPSON_Impressora_Avancar_Papel ( 5 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Impressora_Avancar_Papel ( 5 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Impressora_Avancar_Papel ( 5 );
```

#### 4.7.4 EPSON\_ESC\_ECF\_Impressora\_Abrir\_Gaveta

Esta função abre a gaveta de dinheiro, seguindo o protocolo ESC-ECF.

---

**Sintaxe:**

```
function EPSON_ESC_ECF_Impressora_Abrir_Gaveta():Integer;StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

---

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

---

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

```
Retorno = EPSON_ESC_ECF_Impressora_Abrir_Gaveta( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Impressora_Abrir_Gaveta( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Impressora_Abrir_Gaveta( );
```



#### 4.7.5 EPSON\_ESC\_ECF\_Impressora\_Cortar\_Papel

Esta função corta o papel, seguindo o protocolo ESC-ECF.

---

**Sintaxe:**

```
function EPSON_ESC_ECF_Impressora_Cortar_Papel():Integer;StdCall; External 'InterfaceEpson.dll';
```

---

**Entradas:**

Nenhum.

---

**Saídas:**

Nenhum.

---

**Retornos:**

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

---

---

**Requisitos:**

Comando disponível apenas para ECF modelo TM-T800F.

---

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

```
Retorno = EPSON_ESC_ECF_Impressora_Cortar_Papel( );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_ESC_ECF_Impressora_Cortar_Papel( )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_ESC_ECF_Impressora_Cortar_Papel( );
```

## 4.8 Grupo de Autenticação

Estas funções são utilizadas para impressão da autenticação.

### 4.8.1 EPSON\_Autenticar\_Imprimir

Esta função imprime a autenticação na impressora TM-H6000 com AUTENTICAÇÃO.

#### Sintaxe:

```
function EPSON_Autenticar_Imprimir(pszTextoLinha:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

#### Entradas:

Variável	Tipo	Tam.	Descrição
pszTextoLinha	PCHAR	50 (O,RT)	Linha de texto da autenticação

#### Saídas:

Nenhum.

#### Retornos:

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

#### Requisitos:

- Papel inserido na estação de autenticação;
- Serão permitidas apenas 5 impressões (1 impressão mais 4 reimpressões) da autenticação, conforme previsto em lei vigente.
- Comando não suportado na TM-T88 e TM-T81.

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



```
Retorno = EPSON_Autenticar_Imprimir ( "Texto" );
```

#### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Autenticar_Imprimir ( "Texto" )
```

#### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Autenticar_Imprimir ( 'Texto' );
```

## 4.8.2 EPSON\_Autenticar\_Reimprimir

Esta função reimprime a última autenticação impressa.

### Sintaxe:

```
function EPSON_Autenticar_Reimprimir():Integer; StdCall; External 'InterfaceEpson.dll';
```

### Entradas:

Nenhum.

### Saídas:

Nenhum.

### Retornos:

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

### Requisitos:

- Papel inserido na estação de autenticação;
- Serão permitidas apenas 4 reimpressões da autenticação, conforme previsto em lei vigente.
- Executado logo após o comando EPSON\_Autenticar\_Imprimir

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



```
Retorno = EPSON_Autenticar_Reimprimir ( );
```

### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Autenticar_Reimprimir ( )
```

### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Autenticar_Reimprimir ( );
```

## 4.9 Grupo do Display

Funções de comando do display externo ligado à impressora fiscal.

### 4.9.1 EPSON\_Display\_Enviar\_Texto

Esta função envia uma linha de texto para o display conectado à impressora fiscal

#### Sintaxe:

```
function EPSON_Display_Enviar_Texto(pszLinhaTexto:PChar):Integer;StdCall; External 'InterfaceEpson.dll';
```

#### Entradas:

Variável	Tipo	Tam.	Descrição
pszLinhaTexto	PCHAR	60 (A)	Linha de texto a ser exibida no display.

#### Saídas:

Nenhum.

#### Retornos:

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

#### Requisitos:

Modo normal de operação.

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



```
Retorno = EPSON_Display_Enviar_Texto ( "Linha de texto" );
```

#### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Display_Enviar_Texto ( "Linha de texto" )
```

#### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Display_Enviar_Texto ( 'Linha de texto' );
```

**4.9.2 EPSON\_Display\_Cursor**

Posiciona e configura as propriedades do cursor no display.

**Sintaxe:**

```
function EPSON_Display_Cursor(dwAcao:Integer; dwColuna:Integer; dwLinha:Integer):Integer;StdCall; External  
'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwAcao	INTEGER	-	Ação a ser executada:
			0 – ocultar cursor
			1 – mostrar cursor
			2 – mover o cursor para o início
			3 – mover o cursor para o final
			4 – mover o cursor para o canto esquerdo
			5 – mover o cursor para o canto direito
			6 – mover o cursor para cima
			7 – mover o cursor para baixo
			8 – mover o cursor para a esquerda
			9 – mover o cursor para a direita
			10 – mover o cursor para uma determinada posição
dwColuna	INTEGER	-	Coluna do display para onde se deseja mover o cursor
dwLinha	INTEGER	-	Linha do display para onde se deseja mover o cursor

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

Modo normal de operação.

---

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

```
Retorno = EPSON_Display_Cursor ( 0, 0, 0 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Display_Cursor ( 0, 0, 0 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Display_Cursor ( 0, 0, 0 );
```

### 4.9.3 EPSON\_Display\_Apagar\_Texto

Esta função apaga uma linha de texto do display.

**Sintaxe:**

```
function EPSON_Display_Apagar_Texto(dwLinhaTexto:Integer):Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwLinhaTexto	INTEGER	-	Linha de texto a apagar: <b>0</b> – apagar todas as linhas <b>1</b> – apagar linha onde o cursor se encontra

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Modo normal de operação.

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

```
Retorno = EPSON_Display_Apagar_Texto ( 0 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Display_Apagar_Texto ( 0 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Display_Apagar_Texto ( 0 );
```

**4.9.4 EPSON\_Display\_Configurar**

Esta função configura as preferências de visualização do display.

**Sintaxe:**

```
function EPSON_Display_Configurar(dwBrilho:Integer; dwLampejo:Integer; dwRolagem:Integer):Integer; StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwBrilho	INTEGER	-	Intensidade do brilho do display: -1 – a configuração atual será mantida 0 – 20% da intensidade 1 – 40% da intensidade 2 – 60% da intensidade 3 – 100% da intensidade
dwLampejo	INTEGER	-	Velocidade do lampejo do display: -1 – a configuração atual será mantida 0 – display apagado 1 – display sem lampejo 2 – lampejo lento 3 – lampejo médio 4 – lampejo rápido
dwRolagem	INTEGER	-	Modo de rolagem de texto: -1 – a configuração atual será mantida 0 – sobrescrever texto 1 – rolar texto horizontalmente 2 – rolar texto verticalmente

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

Modo normal de operação.



---

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

```
Retorno = EPSON_Display_Configurar ( 3, 1, 1 );
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Display_Configurar ( 3, 1, 1 )
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Display_Configurar ( 3, 1, 1 );
```

### 4.9.5 EPSON\_Display\_Inicializar

Esta função inicializa e executa teste de diagnóstico no display.

**Sintaxe:**

```
function EPSON_Display_Inicializar(dwAcaoInicializacaoDisplay:Integer):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
dwAcaoInicializacaoDisplay	INTEGER	-	Ação a ser executada: 1 – inicializar display. 2 – executar teste de autoverificação. 3 – inicializar e executar teste de autoverificação.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Modo normal de operação.

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

```
Retorno = EPSON_Display_Inicializar ( 1 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Display_Inicializar ( 1 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

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

## 4.10 Grupo de Funções de Sistema

Funções auxiliares de manipulação das janelas.

### 4.10.1 EPSON\_Sys\_Informar\_Handle\_Janela

Esta função configura o ponteiro para a janela da aplicação.

#### Sintaxe:

```
function EPSON_Sys_Informar_Handle_Janela(hHandle:THandle):Integer;StdCall; External 'InterfaceEpson.dll';
```

#### Entradas:

Variável	Tipo	Tam.	Descrição
hHandle	THANDLE		Ponteiro da janela da aplicação.

Obs. THANDLE é um ponteiro para uma janela do Windows.

#### Saídas:

Nenhum.

#### Retornos:

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

#### Requisitos:

Nenhum.

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



```
Retorno = EPSON_Sys_Informar_Handle_Janela ( Handle );
```

#### Exemplo em C# :



```
Retorno = EPSON_Sys_Informar_Handle_Janela ( Handle );
```

#### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Sys_Informar_Handle_Janela ( Handle )
```

#### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Sys_Informar_Handle_Janela ( Handle );
```

**4.10.2 EPSON\_Sys\_Atualizar\_Janela**

Esta função redesenha a janela da aplicação cujo ponteiro foi informado na função anterior.

**Sintaxe:**

```
function EPSON_Sys_Atualizar_Janela():Integer;StdCall; External 'InterfaceEpson.dll';
```

**Entradas:**

Nenhum.

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

Nenhum.

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

```
Retorno = EPSON_Sys_Atualizar_Janela ( );
```

**Exemplo em C# :**

```
Retorno = EPSON_Sys_Atualizar_Janela ( );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Sys_Atualizar_Janela ( )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Sys_Atualizar_Janela ( );
```

**4.10.3 EPSON\_Sys\_Aguardar\_Arquivo**

Esta função aguarda a condição de existência de um arquivo informado.

**Sintaxe:**

```
function EPSON_Sys_Aguardar_Arquivo(pszNomeArquivo:PChar; dwTimeout:Integer;  
                                   bBloqueiaEntradas:Boolean; bSincrono:Boolean):Integer;StdCall;  
External 'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszNomeArquivo	PCHAR	1024	Nome do arquivo
dwTimeout	, INTEGER	-	Espera máxima em segundos
bBloqueiaEntradas	BOOLEAN	-	<b>True</b> - Bloqueia a as entradas do mouse e teclado. <b>False</b> - Não bloqueia as entradas.
bSincrono	BOOLEAN	-	<b>True</b> - Trabalha com multithread. <b>False</b> - Não trabalha com multithread.

**Saídas:**

Nenhum.

**Retornos:**

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

**Requisitos:**

Nenhum.

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

```
Retorno = EPSON_Sys_Aguardar_Arquivo ( "arquivo.txt", 5, true, false );
```

**Exemplo em C# :**

```
Retorno = EPSON_Sys_Aguardar_Arquivo ( "c:\\Temp\\arquivo.txt", 5, true,  
false );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Sys_Aguardar_Arquivo ( "c:\\Temp\\arquivo.txt", 5, true,  
false )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := EPSON_Sys_Aguardar_Arquivo ( 'c:\\Temp\\arquivo.txt', 5, true,  
false );
```

#### 4.10.4 EPSON\_Sys\_Bloquear\_Entradas

Esta função bloqueia as entradas de teclado e mouse.

##### Sintaxe:

```
function EPSON_Sys_Bloquear_Entradas(bBloqueiaEntradas:Boolean):Integer;StdCall; External  
'InterfaceEpson.dll';
```

##### Entradas:

Variável	Tipo	Tam.	Descrição
bBloqueiaEntradas	BOOLEAN	-	<b>True</b> - Bloqueia a as entradas do mouse e teclado. <b>False</b> - Desbloqueia as entradas.

##### Saídas:

Nenhum.

##### Retornos:

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

##### Requisitos:

Nenhum.

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



```
Retorno = EPSON_Sys_Bloquear_Entradas ( true );
```

##### Exemplo em C# :



```
Retorno = EPSON_Sys_Bloquear_Entradas ( true );
```

##### Exemplo em Visual Basic 6.0 / Visual Basic.NET:



```
Retorno = EPSON_Sys_Bloquear_Entradas ( true )
```

##### Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:



```
Retorno := EPSON_Sys_Bloquear_Entradas ( true );
```

**4.10.5 EPSON\_Sys\_Log**

Este comando configura as propriedades de gravação de Log de operações e comunicação serial. É possível especificar o caminho onde os arquivos de log serão gravados. A biblioteca cria automaticamente os arquivos com extensão TXT separando cada dia de operação em um arquivo específico. Os arquivos de dias anteriores são automaticamente compactados com o formato ZIP.

Os modos de operação são: desabilitado, habilitado, e habilitado em caso de erro. Este último modo de operação grava o log apenas dos comandos que resultarem em erro de execução.

**Registro do Windows**

Também é possível habilitar ou desabilitar a gravação de Log através do registro do Windows. Bastando alterar o valor do registro "Log" contido na chave HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson de acordo com os seguintes parâmetros:

Ação do Log:

0 – Log desabilitado

1 – Log habilitado

2 – Log somente em caso de erro

O local de gravação do arquivo de Log é definido pela chave "Path" também encontrada no registro do Windows. O local padrão de gravação é C:\Epson\InterfaceEpson\Datafiles\

Obs.: Ao ser carregada a DLL a ação do log é determinada pelo registro, caso seja chamada à função EPSON\_Sys\_Log prevalecerá à ação determinada pela mesma.

Windows Vista e Windows7:

- Caso o controle de acesso do usuário (UAC) do Windows Vista esteja ativado e a aplicação não tenha privilegio de administrador os registros serão gravados na chave.

HKEY\_CURRENT\_USER\CLASSES\SOFTWARE\VIRTUALSTORE\MACHINE\SOFTWARE\EPSON\InterfaceEpson.

- Caso a aplicação tenha privilegio de administrador os registros serão gravados na chave padrão HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson.

- Caso o controle de acesso do usuário (UAC) do Windows Vista não esteja ativado e a aplicação não tenha privilegio de administrador não será possível gravar ou ler os dados do registro, portanto o log somente poderá ser ativado pela função EPSON\_Sys\_Log.

**Sintaxe:**

```
function EPSON_Sys_Log(pszCaminhoLog:PChar; dwAcaoLog:Integer):Integer;StdCall; External  
    'InterfaceEpson.dll';
```

**Entradas:**

Variável	Tipo	Tam.	Descrição
pszCaminhoLog	PCHAR	1024	Nome do diretório de gravação do Log.
dwAcaoLog	INTEGER	-	Ação do Log: 0 – Log desabilitado 1 – Log habilitado 2 – Log somente em caso de erro

**Saídas:**

Nenhum.

**Retornos:**

FUNC\_SUCESSO

Operação realizada com sucesso.

FUNC\_ERRO

Erro durante a execução.

**Requisitos:**

Nenhum.

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

```
Retorno = EPSON_Sys_Log ( "C:\\Epson\\InterfaceEpson\\Datafiles\\", 1 );
```

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = EPSON_Sys_Log ( "C:\\Epson\\InterfaceEpson\\Datafiles\\", 1 )
```

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

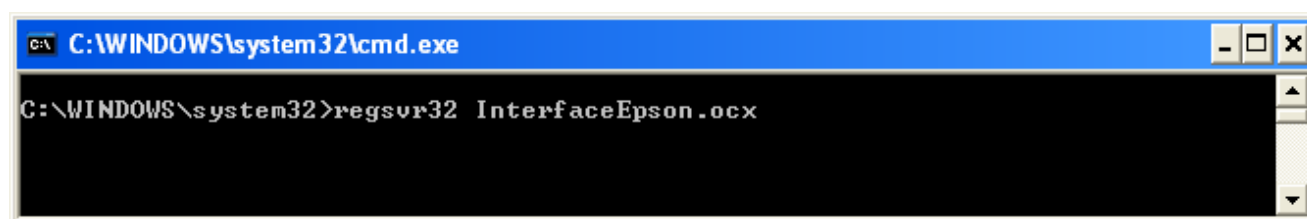
```
Retorno := EPSON_Sys_Log ( ' C:\\Epson\\InterfaceEpson\\Datafiles\\', 1 );
```



## 5 Interface OCX

Com o intuito de disponibilizar um número cada vez maior de ferramentas aos desenvolvedores, foi criada a interface Epson OCX, que facilita a utilização das Impressoras Fiscais Epson através de um browser Web. Trata-se de um Active X que disponibiliza todas as funções implementadas na interface dll, utilizando os mesmos nomes de funções e parâmetros.

Para utilizar a InterfaceEpson.ocx é necessário copia-lá preferencialmente no diretório de sistema e registrá-la no Windows utilizando a linha de comando abaixo:



Obs.:

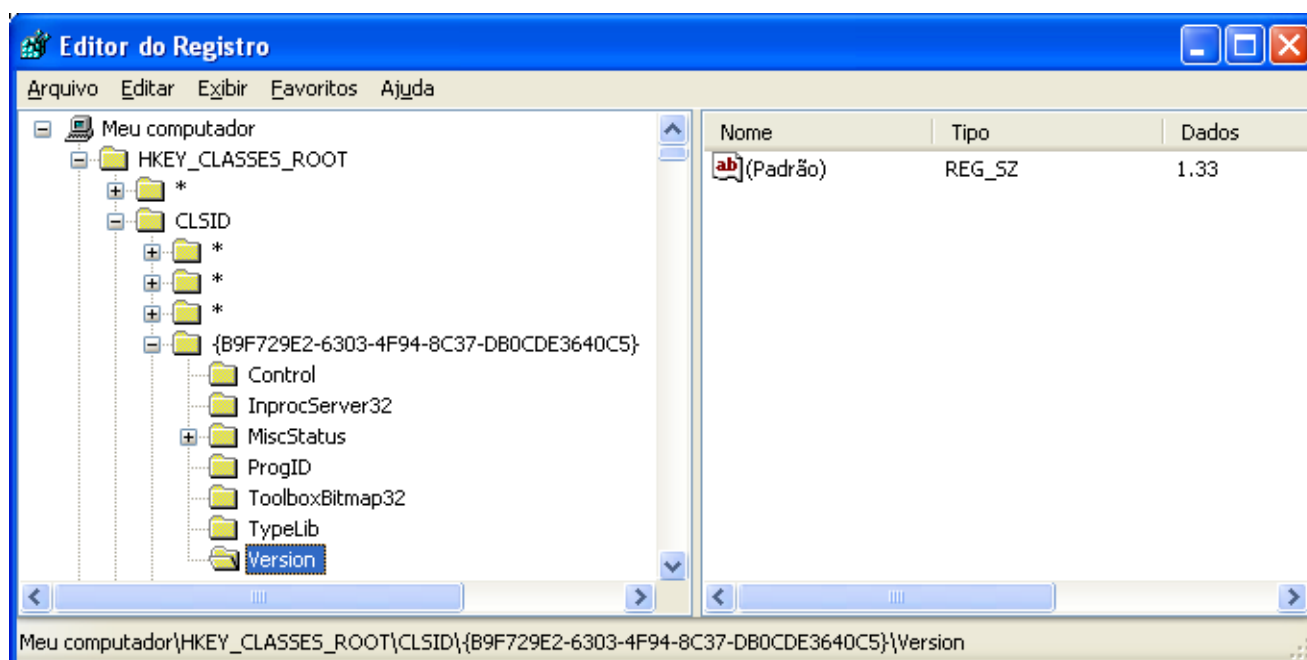
-Dependendo o sistema operacional utilizado é necessário privilegio de administrador para registrar a OCX.

-No Windows Vista caso controle de acesso do usuário (UAC) esteja ativado é necessário executar o prompt de comando como Administrador do sistema para registrar a OCX.

Após ser registrada, a ocx aparecerá no registro do Windows na seguinte pasta:

HKEY\_CLASSES\_ROOT\CLSID\{B9F729E2-6303-4F94-8C37-DB0CDE3640C5}

A figura abaixo ilustra quais informações são disponibilizadas ao acessar o registro através do editor de registros do Windows.



---

**Exemplo em HTML:**

```
<OBJECT id="InterfaceEpson" classid="clsid:b9f729e2-6303-4f94-8c37-  
db0cde3640c5" VIEWASTEXT></OBJECT>  
<script type="text/javascript">  
function Abrir_PortaEx()  
{  
    Retorno = InterfaceEpson.EPSON_Serial_Abrir_PortaEX();  
}  
</script>
```

---

**Exemplo em Visual Basic 6.0 / Visual Basic.NET:**

```
Retorno = InterfaceEpson.EPSON_Serial_Abrir_PortaEX()
```

---

**Exemplo em Delphi 7 / Delphi 2010 / Delphi XE:**

```
Retorno := InterfaceEpson.EPSON_Serial_Abrir_PortaEX();
```

## **6 SINTEGRA (Convênio ICMS 57/95)**

Existem 3 maneiras de gerar os registros SINTEGRA pela DLL InterfaceEpson:

### **1 - Utilizando o comando EPSON\_Obter\_Dados\_MF\_MFD**

Utilizando o comando EPSON\_Obter\_Dados\_MF\_MFD pode se gerar os registros SINTEGRA por intervalo de data e escolher os tipos de registros a serem gerados.

### **2 - Automaticamente ao emitir uma Redução Z**

Automaticamente ao emitir a Redução Z será gravado um arquivo texto, em disco, com o os registros 60M, 60A, 60I e 60D referente ao dia de movimento desta redução. Este arquivo tem a seguinte nomenclatura

Sintegra\_DDMMAAAA.txt (sendo DDMMAAAA a data de movimento).

Para a correta geração deste arquivo algumas medidas devem ser tomadas:

- A impressão de todos os documentos deve ser feita através da DLL
- O ECF não pode ser trocado de PDV
- O arquivo temp\_mfd.dat criado no diretório da InterfaceEpson não pode ser excluído
- O Path de gravação dos arquivos gerados será determinado pelo conteúdo do registro "Path" na chave\* HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson, caso o registro "path" não exista ou seu conteúdo contenha um endereço de path inválido será utilizado o path padrão "C:\EPSON\InterfaceEpson\DataFiles".
- Para habilitar a geração automática deve-se utilizar a função

EPSON\_Config\_Habilita\_Sintegra\_Auto

Obs.: Chave de registro no Windows Vista\*:

- Caso o controle de acesso do usuário (UAC) do Windows Vista esteja ativado e a aplicação não tenha privilegio de administrador os registros serão gravados na chave.

HKEY\_CURRENT\_USER\CLASSES\SOFTWARE\VIRTUALSTORE\MACHINE\SOFTWARE\EPSON\InterfaceEpson.

- Caso a aplicação tenha privilegio de administrador os registros serão gravados na chave padrão

HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson.

- Caso o controle de acesso do usuário (UAC) do Windows Vista não esteja ativado e a aplicação não tenha privilegio de administrador não será possível gravar ou ler os dados do registro, portanto será utilizado o path padrão "C:\EPSON\InterfaceEpson\DataFiles".

\*(testes realizados no Windows Vista Business)

## **7 Ato/COTEPE 17/04 (CAT 52)**

O Ato/COTEPE 17/04 para a CAT 52 pode ser gerado de três maneiras:

### **1 - Utilizando o comando EPSON\_Obter\_Dados\_MF\_MFD**

Utilizando o comando EPSON\_Obter\_Dados\_MF\_MFD pode-se gerar o Ato/COTEPE para qualquer período desejado, bastando informar nos parâmetros deste comando as datas ou CRZs de início e fim. Os dados serão lidos da Memória Fiscal e da Memória Fita-Detalhe e o arquivo será gerado em disco com o tipo de leitura e o nome desejado.

### **2 - Automaticamente ao emitir uma Redução Z**

Automaticamente ao emitir a Redução Z será gravado um arquivo texto, em disco, com o Ato/COTEPE referente ao dia de movimento desta redução. Este arquivo tem a nomenclatura definida pela CAT52. Ex.: EP001234.AA7.

Para a correta geração deste arquivo algumas medidas devem ser tomadas:

- A impressão de todos os documentos deve ser feita através da DLL
- O ECF não pode ser trocado de PDV
- O arquivo temp\_mfd.dat criado no diretório da InterfaceEpson não pode ser excluído
- O Path de gravação dos arquivos gerados será determinado pelo conteúdo do registro "Path" na chave\* HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson, caso o registro "path" não exista ou seu conteúdo contenha um endereço de path inválido será utilizado o path padrão "C:\EPSON\InterfaceEpson\DataFiles".
- Para habilitar a geração automática deve-se utilizar a função EPSON\_Config\_Habilita\_CAT52\_Auto

Obs.: Chave de registro no Windows Vista\*:

- Caso o controle de acesso do usuário (UAC) do Windows Vista esteja ativado e a aplicação não tenha privilégio de administrador os registros serão gravados na chave.


HKEY\_CURRENT\_USER\CLASSES\SOFTWARE\VIRTUALSTORE\MACHINE\SOFTWARE\EPSON\InterfaceEpson.

- Caso a aplicação tenha privilégio de administrador os registros serão gravados na chave padrão

HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson.

- Caso o controle de acesso do usuário (UAC) do Windows Vista não esteja ativado e a aplicação não tenha privilégio de administrador não será possível gravar ou ler os dados do registro, portanto será utilizado o path padrão "C:\EPSON\InterfaceEpson\DataFiles".

\*(testes realizados no Windows Vista Business)

 EXCEED YOUR VISION	Manual da Interface Epson		
	Ver.: 4.9.0	Data: 10/02/2017	Pg.: 405/406

## 8 Ato/COTEPE 06/08 (PAF-ECF)

Para facilitar a geração dos arquivos exigidos pelo Ato COTEPE 06/08 implementamos em nossa DLL funcionalidades como comandos que retornam informações do ECF e a geração do arquivo eletrônico de registros efetuados pelo ECF.

### 8.1 Comandos de auxílio o desenvolvimento do Programa Aplicativo Fiscal (PAF).

Segue a lista de comandos que foram implementados para facilitar o desenvolvimento do Programa Aplicativo Fiscal.

- EPSON\_Obter\_Codigo\_Nacional\_ECF  
Retorna o código nacional do ECF e nome a ser usado pelo arquivo gerado pelo PAF-ECF.
- EPSON\_Obter\_Versao\_SWBasico  
Retorna a versão do Software Básico e sua data de instalação.
- EPSON\_Configura\_Mensagem\_Aplicacao  
Imprime automaticamente os dados do PAF nos cupons fiscais e não fiscais.
- EPSON\_Fiscal\_Vender\_ItemEX  
Salva informação que indica se o produto é de fabricação própria ou de terceiros.
- EPSON\_Config\_Dados\_PAF-ECF  
Configura os dados do PAF-ECF para serem corretamente preenchidos no arquivo de registros efetuados pelo PAF-ECF

### 8.2 Geração do arquivo eletrônico de registros efetuados pelo PAF-ECF

(Ato COTEPE 06/08 Anexo VI requisito XXV)

O arquivo eletrônico dos registros efetuados pelo PAF-ECF pode ser gerado de três maneiras:

#### 1 - Utilizando o comando EPSON\_Obter\_Dados\_MF\_MFD

Utilizando o comando EPSON\_Obter\_Dados\_MF\_MFD pode-se gerar o arquivo de registros efetuados pelo PAF-ECF para qualquer período desejado, bastando informar nos parâmetros deste comando as datas ou CRZs de início e fim. Os dados serão lidos da Memória Fiscal e da Memória Fita-Detalhe e o arquivo será gerado em disco com o tipo de leitura e o nome desejado.

## 2 - Automaticamente ao emitir uma Redução Z

Automaticamente ao emitir a Redução Z será gravado um arquivo texto, em disco com os registros efetuados pelo PAF-ECF referente ao dia de movimento desta redução. Este arquivo tem a nomenclatura definida pelo Ato COTEPE 06/08 Ex.: 9999990000000000000109062009.txt

Para a correta geração deste arquivo algumas medidas devem ser tomadas:

- A impressão de todos os documentos deve ser feita através da DLL
- O ECF não pode ser trocado de PDV
- O arquivo temp\_mfd.dat criado no diretório da InterfaceEpson não pode ser excluído
- O Path de gravação dos arquivos gerados será determinado pelo conteúdo do registro "Path" na chave\* HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson, caso o registro "path" não exista ou seu conteúdo contenha um endereço de path inválido será utilizado o path padrão "C:\EPSON\InterfaceEpson\DataFiles".
- Para habilitar a geração automática deve-se utilizar a função EPSON\_Config\_Habilita\_PAFECF\_Auto
- Para correta geração do arquivo de registros do PAF-ECF é necessário utilizar antes da ReduçãoZ a função EPSON\_Config\_Dados\_PAFECF para configurar os dados referentes ao programa aplicativo fiscal.

Obs.: Chave de registro no Windows Vista\*:

- Caso o controle de acesso do usuário (UAC) do Windows Vista esteja ativado e a aplicação não tenha privilégio de administrador os registros serão gravados na chave.

HKEY\_CURRENT\_USER\CLASSES\SOFTWARE\VIRTUALSTORE\MACHINE\SOFTWARE\EPSON\InterfaceEpson.

- Caso a aplicação tenha privilégio de administrador os registros serão gravados na chave padrão

HKEY\_LOCAL\_MACHINE\SOFTWARE\EPSON\InterfaceEpson.

- Caso o controle de acesso do usuário (UAC) do Windows Vista não esteja ativado e a aplicação não tenha privilégio de administrador não será possível gravar ou ler os dados do registro, portanto será utilizado o path padrão "C:\EPSON\InterfaceEpson\DataFiles".

\*(testes realizados no Windows Vista Business)