



Manual de Programação FIT

Revisão 2.01F

CONTATOS

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

CEP 01201-901

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

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

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

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

Apoio ao Desenvolvedor: 0800 707 98 90

www.elgin.com.br

Tabela de Conteúdo

Part I Manual de Programação da FIT	8
1 Introdução	8
2 Firmware da impressora fiscal	8
Modos de Operação	8
Modo de intervenção Técnica	8
Modo Normal de Operação	9
Limites do ECF	9
Tabelas do ECF	10
Inicializações Necessárias para o uso do ECF	11
Dados de Identificação do Proprietário	12
Dados Clichê	12
Situações Tributárias	13
Meios de Pagamento	14
Totalizadores Não Fiscais	15
Gerenciais	16
Departamentos	17
Criptografia do GT	17
Símbolos Fiscais	17
Tipos de Documentos	17
Cupom Fiscal (CF)	18
Comprovante Não Fiscal (CNF)	19
Comprovante de Crédito ou Débito (CCD)	19
Relatório Gerencial	20
Relatórios Fiscais	21
Outros Documentos	22
Cheque e Autenticação	23
Recursos	24
Horário de Verão	24
Configurações	24
Mecanismo	24
Leitura do Texto Impresso	25
Modos e Comandos de Impressão	25
Estados do Software Básico	28
Indicadores	28
Leitura de Registradores	28
Memória de Fita Detalhe (MFD)	29
Perda da Memória RAM	30
Relógio	30
Diagnóstico de Erros	30
Falta de Energia	31
3 Protocolo de Comunicação	31
Fluxo de Comandos	31
Estrutura dos Comandos	32
Numeração dos Comandos	33
Identificação do Comando Igual a Zero	34
Comandos e Parâmetros	35

Respostas.....	35
Comandos.....	36
Protocolo Logger I	36
Tradutor de Protocolo Logger I.....	36
Programação do Protocolo.....	36
Uso de Recursos FiscNET através do Protocolo Logger I.....	37
Biblioteca	38
Considerações Finais	38
4 O Firmware da Impressora Fiscal	38
5 Tipos de Dados	39
6 Tipo de parâmetro	40
7 Registradores Protocolo FiscNET	40
Aliquota00ICMSReducao	40
Aliquota00Reducao	41
Aliquota01ICMSReducao	41
Aliquota01Reducao	41
Aliquota02ICMSReducao	41
Aliquota02Reducao	42
Aliquota03ICMSReducao	42
Aliquota04Reducao	42
Aliquota05ICMSReducao	42
Aliquota05Reducao	43
Aliquota06ICMSReducao	43
Aliquota06Reducao	43
Aliquota07ICMSReducao	44
Aliquota07Reducao	44
Aliquota08ICMSReducao	44
Aliquota08Reducao	44
Aliquota09ICMSReducao	45
Aliquota09Reducao	45
Aliquota10ICMSReducao	45
Aliquota10Reducao	45
Aliquota11ICMSReducao	46
Aliquota11Reducao	46
Aliquota12ICMSReducao	46
Aliquota12Reducao	46
Aliquota13ICMSReducao	47
Aliquota13Reducao	47
Aliquota14ICMSReducao	47
Aliquota14Reducao	47
Aliquota15ICMSReducao	48
Aliquota15Reducao	48
AliquotaDisponivel	48
Arredondamento	48
BaudRate	49
CancelamentosICMSReducao	49
CancelamentosISSQNReducao	49
CCF	49
CDC	50
CER	50
CFC	50
CFD	50
Cliche	51

CMC7Campo1	51
CMC7Campo2	51
CMC7Campo3	51
CMC7Documento	52
CMC7Memoria	52
CNPJ	52
CON	52
ContadorDocUltimoltemVendido	53
ContadorProprietarios	53
ContadorReinicioReducao	53
ContadorVersoes	53
ContrasteBobina	53
ContrastePagina	54
COO	54
COOInicioDia	54
COOReducao	54
CRO	55
CRORestantes	55
CRZ	55
CRZRestantes	55
DadosUltimaReducaoZ	56
Data	56
DataAbertura	56
DataReducao	56
DataUltimoDoc	57
DescontosReducao	57
DiaAberto	57
DiaFechado	57
DocumentoAberto	57
ECF	58
EnderecoSoftwareBasico	58
EspacamentoDocumentos	58
EspacamentoLinhas	58
EstadoFiscal	59
EstadoGeralECF	59
GNF	59
GRG	59
GT	60
GTInicial	60
HabilitaCupomAdicional	60
Hora	60
HoraReducao	60
HorarioVerao	61
HoraUltimoDoc	61
IE	61
IM	61
Indicadores	62
IntervencaoTecnica	62
Loja	62
Marca	62
MecanismoBobina	63
MecanismoBobinaPadrao	63
MecanismoPagina	63
MecanismosBobina	63

MFDDesconectada	64
MFDdesconectada	64
MFD Esgotada	64
MFD Livre	64
Modelo	64
NCN	65
NFC	65
NomeEstadoFiscal	65
NomeMoedaPlural	65
NomeMoedaSingular	66
NumeroMeiosPagamentoCupomAtual	66
NumeroSerieECF	66
NumeroSerieMFD	66
NumLinhasImpressas	67
OperacoesNaoFiscaisReducao	67
PercentualMFDUtilizada	67
ProtocoloSerial	67
RAMInvalida	68
RelogioInconsistente	68
SemPapel	68
SensorAutenticacao	68
SensorCabecote	68
SensorCheque	69
SensorGaveta	69
SensorPoucoPapel	69
SimboloMoeda	69
SoftwareBasico	70
TamanhoSoftwareBasico	70
TempoAcionamentoPadrao	70
TempoEmitindoDocFiscal	70
TempoOperacional	70
TextoLivre	71
TipoDocEmEmissao	71
TipoECF	71
TipoUltimoDocEmitido	71
TotalAliquota00Reducao	72
TotalAliquota01Reducao	72
TotalAliquota02Reducao	72
TotalAliquota03Reducao	72
TotalAliquota04Reducao	73
TotalAliquota05Reducao	73
TotalAliquota06Reducao	73
TotalAliquota07Reducao	73
TotalAliquota08Reducao	74
TotalAliquota09Reducao	74
TotalAliquota10Reducao	74
TotalAliquota11Reducao	74
TotalAliquota12Reducao	75
TotalAliquota13Reducao	75
TotalAliquota14Reducao	75
TotalAliquota15Reducao	75
TotalAliquotaF1Reducao	76
TotalAliquotaFS1Reducao	76
TotalAliquotal1Reducao	76

TotalAliquotas1Reducao	76
TotalAliquotaN1Reducao	77
TotalAliquotaNS1Reducao	77
TotalDiaAcrescimos	77
TotalDiaAcrescimosItens	77
TotalDiaAcrescimosNaoFiscais	78
TotalDiaCancelamentoItens	78
TotalDiaCancelamentosICMS	78
TotalDiaCancelamentosISSQN	78
TotalDiaCancelamentosNaoFiscais	79
TotalDiaDescontos	79
TotalDiaDescontosItens	79
TotalDiaDescontosNaoFiscais	79
TotalDiaDinheiro	79
TotalDiaImpostoAliquota	80
TotalDiaImpostoICMS	80
TotalDiaImpostoISSQN	80
TotalDiasencaoICMS	80
TotalDiasencaoISSQN	81
TotalDiaISSQN	81
TotalDiaMeioPagamento	81
TotalDiaNaoFiscal	81
TotalDiaNaoTributadoICMS	81
TotalDiaNaoTributadoISSQN	82
TotalDiaOperacoesNaoFiscais	82
TotalDiaQuantidadeVendasDepartamento	82
TotalDiaSubstituicaoTributariaICMS	82
TotalDiaSubstituicaoTributariaISSQN	83
TotalDiaTrocoTipoAcessoR	83
TotalDiaValorAliquota	83
TotalDiaValorEmCaixa	83
TotalDiaValorRecebido	84
TotalDiaValorTributadoICMS	84
TotalDiaValorTributadoISSQN	84
TotalDiaValorVendasDepartamento	84
TotalDiaVendaBruta	85
TotalDiaVendaLiquida	85
TotalDocAcrescimos	85
TotalDocAliquota	85
TotalDocBruto	85
TotalDocCancelamentoICMS	86
TotalDocCancelamentoISSQN	86
TotalDocCancelamentosNaoFiscais	86
TotalDocDescontos	86
TotalDocDinheiro	87
TotalDocIsencaoICMS	87
TotalDocIsencaoISSQN	87
TotalDocISSQN	87
TotalDocLiquido	87
TotalDocMeioPagamento	88
TotalDocNaoFiscal	88
TotalDocNaoTributadoICMS	88
TotalDocNaoTributadoISSQN	88
TotalDocSubstituicaoTributariaICMS	89

TotalDocSubstituicaoTributariaISSQN	89
TotalDocTransacoesVenda	89
TotalDocTroco	89
TotalDocValorPago	90
TradutorIndiceMeioPagamentoDinheiro	90
UltimoCmdNumero	90
UltimoErroCircunstancia	90
UltimoErroCodigo	91
UltimoErroNome	91
VendaBrutaReducao	91
VersaoSW	91
8 Retornos do Protocolo FiscNET	92
9 Comandos do Protocolo FiscNET	96
AbreCreditoDebito	97
AbreCupomFiscal	97
AbreCupomNaoFiscal	98
AbreGaveta	98
AbreGerencial	98
AcertaHorarioVerao	99
AcresceltemFiscal	99
AcresceltemNaoFiscal	100
AcresceSubtotal	100
AvancaPapel	101
CancelaCupom	101
CancelaltemFiscal	101
CancelaltemNaoFiscal	101
ChancelaCheque	102
CortaPapel	102
CruzaCheque	102
DefineAliquota	103
DefineDadosCabecalho	103
DefineDepartamento	104
DefineGerencial	104
DefineMeioPagamento	105
DefineNaoFiscal	106
DefineProprietario	106
EmiteltemNaoFiscal	107
EmitelituraFitaDetalhe	107
EmitelituraMF	108
EmitelituraX	108
EmitelituraZ	109
EmitelituraViaCreditoDebito	109
EncerraDocumento	109
EscreveData	110
EscreveHora	110
EscreveIndicador	110
EscreveInteiro	111
EscreveMoeda	111
EscreveTexto	111
EstornaCreditoDebito	112
EstornaMeioPagamento	112
ExcluiAliquota	113
ExcluiDepartamento	113

ExcluiGerencial	113
ExcluiMeioPagamento	114
ExcluiNaoFiscal	114
ExpulsaDocumento	114
IdentificaConsumidor	114
ImprimeAutenticacao	115
ImprimeCheque	115
ImprimeCupomAdicional	117
ImprimeDadosDepartamento	117
ImprimeTexto	118
IniciaFitaDetalhe	118
LeAliquota	118
LeData	119
LeDepartamento	119
LeGerencial	120
LeHora	120
LeImpressao	120
LeIndicador	121
LeInteiro	121
LeMeioPagamento	122
LeMoeda	122
LeNaoFiscal	122
LeTexto	123
PagaCupom	123
ProgramaRelogio	124
ReimprimeViaCreditoDebito	124
VendeItem	124
10 Retorno dos Registradores	125
Tipos de Documentos	126
Estado Geral ECF	126
Dados Última Redução Z	127
11 Tabela de estados do Software Básico	127
12 Indicadores do Software Básico	127
13 Restrições do Tradutor de Protocolo	129
14 Recomendações de Uso da MFD	129

1 Manual de Programação da FIT

1.1 Introdução

Este manual de programação abrange um conjunto de equipamentos que possuem internamente o mesmo software básico de controle.

Este conjunto de equipamentos fazem parte da série Logger II. A "geração" Logger II de equipamentos fiscais possuem, como característica principal, a implementação da Fita Detalhe Eletrônica e abrange o máximo de funcionalidade dos equipamentos pré-existentes da série Logger I, sendo que há alterações inevitáveis devido as mudanças na legislação.

1.2 Firmware da impressora fiscal

1.2.1 Modos de Operação

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

1.2.1.1 Modo de intervenção Técnica

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

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

Automaticamente, ao ligar o ECF sem o *jumper* para entrar em MIT, o *software* básico faz as seguintes verificações:

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

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

Alguns dos principais comandos do protocolo que podem ser executados em MIT são:

Comando	Operação
ProgramaRelogio	Programa o relógio do ECF (não aceita data/hora anterior ao último documento emitido)
IniciaFitaDetalhe	Usado para iniciar Memória de Fita Detalhe adicional (segunda ou posterior)
EmiteLeituraX	Emite um relatório fiscal chamado Leitura X
EmiteLeituraMF	Emite um relatório fiscal chamado Leitura da Memória Fiscal
EmiteLeituraFitaDetalhe	Permite selecionar um período de emissão de documentos através do intervalo de COO e imprimí-los.

1.2.1.2 Modo Normal de Operação

O Modo Normal de Operação é também chamado de Modo Fiscal. Neste modo, as operações normais de emissão de cupons são permitidas, tais como: venda de itens, cupons não fiscais, relatórios gerenciais, emissão de vias de crédito/débito, cancelamentos de documentos fiscais e não-fiscais, etc. O Modo Normal de Operação e Modo de Intervenção Técnica são exclusivos, ou seja, o ECF estará no Modo Normal de Operação sempre que não estiver em MIT.

1.2.2 Limites do ECF

Os limites dos valores armazenados pelo ECF são definidos através de seus tipos de dados (ver definição dos tipos de dados no tópico [Tipos de Dados](#)).

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

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

Identificação da Variável	Até a Versão 03.02.01		A partir da Versão 03.03.00	
	Dígitos	Formato e Limite	Dígitos	Formato
Preço por Item	10	99.999.999,00	8	99.999.999,99
Capacidade Máxima por Registro de Item	11	999.999.999,99	11	999.999.999,99
Quantidade por item	8	99.999,999	7	9.999,999
Totalizadores parciais	12	9.999.999.999,99	13	99.999.999.999,99
GT	20	999.999.999.999.999,99	18	9.999.999.999.999,99
Venda Bruta	13	99.999.999.999,99	14	999.999.999.999,99
Venda Líquida	13	99.999.999.999,99	14	999.999.999.999,99
Número de Cupons Cancelados (CFC)	4	9999	4	9999
Contador de Ordem de Operação (COO)	6	999999	6	999999
Geral de Operação Não-Fiscal (GNF)	6	999999	6	999999
Contador de Cupom Fiscal (CCF)	6	999999	6	999999
Comprovante de Crédito/Débito (CDC)	6	999999	4	9999
Geral de Relatórios Gerenciais (GRG)	6	999999	6	999999
Contador de Reduções Z (CRZ)	6	999999	4	9999
Geral de Oper. Não-Fiscal Canc. (NFC)	6	999999	4	9999
Contador de Fita-Detalhe (CFD)	6	999999	6	999999
Contador de Reinício de Operação (CRO)	6	999999	3	999
Itens por Cupom Fiscal	3	999	3	999

1.2.3 Tabelas do ECF

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

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

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

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

1.2.4 Inicializações Necessárias para o uso do ECF

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

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

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

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

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

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

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

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

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

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

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

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

NOTA

A impressora fiscal não necessita a carga da tabela de criptografia para ser inicializada. A tabela é

gerada automaticamente a partir do software básico quando ocorre a carga do cabeçalho.

1.2.4.1 Dados de Identificação do Proprietário

Os dados de identificação do proprietário são informações necessárias para iniciar as operações fiscais do ECF. São dados de identificação:

Atributo	Descrição
CNPJ	CNPJ do estabelecimento usuário do ECF
InscricaoEstadual	Inscrição Estadual do estabelecimento usuário do ECF
InscricaoMunicipal	Inscrição Municipal do estabelecimento usuário do ECF
Senha	Senha para habilitação do uso do ECF

O único campo opcional é a Inscrição Municipal. Os demais são obrigatórios.

Cada novo registro de identificação do proprietário gravado incrementará o contador de proprietários do ECF. O **número máximo** de identificações aceitos pelo ECF são **20** (vinte). A gravação de um registro de identificação contendo um CNPJ idêntico ao último registrado no ECF não será considerada troca de proprietário.

A troca de proprietário ocasiona o zeramento de uma série de contadores e totalizadores do ECF, tais como: COO, GNF, GRG, GT, entre outros.

A senha deve ser solicitada à revenda.

IMPORTANTE

A definição de proprietário somente pode ocorrer com o dia fechado e no estado de intervenção técnica.

O comando relacionado a identificação do proprietário é:

Comando	Operação
DefineProprietario	Define um novo proprietário ou redefine um já existente

Há [registradores](#) que retornam valores associados a identificação de proprietário. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeTexto, LeInteiro, etc (ver tópico [Tipos de Dados](#)).

1.2.4.2 Dados Clichê

Também chamado de cabeçalho, os dados do clichê são impressos no início de cada cupom. Definem os dados do clichê:

Atributo	Descrição
ECF	Número seqüencial do ECF no estabelecimento entre 1 e 999.
Loja	Número de identificação do estabelecimento entre 1 e 999.
TextoCabeçalho	Dados do clichê, podendo conter: Razão Social, Nome Fantasia e Endereço, distribuídos em até 3 linhas.

IMPORTANTE

A definição dos dados do clichê somente pode ocorrer com o dia fechado e no estado de intervenção técnica.

O comando relacionado definição do clichê é:

Comando	Operação
DefineDadosCabeçalho	Define os dados do cabeçalho ou redefine os já existentes

Há [registradores](#) que retornam valores associados aos dados do clichê. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeTexto, LeInteiro, etc

(ver tópico [Tipos de Dados](#)).

1.2.4.3 Situações Tributárias

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

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

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

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

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

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

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

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

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

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

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

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

IMPORTANTE

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

Os principais comandos relacionados a tabela de alíquotas são:

Comando	Operação
DefineAliquota	Define uma nova alíquota ou redefine uma já existente
ExcluiAliquota	Exclui uma alíquota previamente definida
LeAliquota	Lê os atributos de uma alíquota definida

Há vários [registradores](#) que retornam valores associados a alíquotas, tais como seus percentuais definidos em cada redução realizada. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: `LeMoeda`, `LeTexto`, `LeInteiro`, etc (ver tópico [Tipos de Dados](#)).

1.2.4.4 Meios de Pagamento

Um meio de pagamento é composto pelas seguintes informações:

Atributo	Descrição
CodMeioPagamentoProgram	Código único que identifica um meio de pagamento
NomeMeioPagamento	Nome que identifica o meio de pagamento
PermiteVinculado	Indica se o meio de pagamento permite a emissão de documentos vinculados (<i>Yes</i>) ou não permite (<i>No</i>)
DescricaoMeioPagamento	Descrição opcional para o meio de pagamento

Pode ser identificado pelo seu código ou pelo seu nome.

O indicador de Crédito/Débito (campo `PermiteVinculado`), sendo configurado como ativo, habilitará a emissão de crédito/débito após cada pagamento com este código. Ao ser registrado um pagamento que possua este indicador ativo, um contador específico de controle será incrementado em uma unidade no ECF, indicando a não emissão de documento de crédito/débito (contador **NCN**). Este contador somente será decrementado após a ocorrência de uma das opções abaixo:

- a emissão do respectivo documento de crédito/débito.
- estorno de um meio de pagamento que habilite CCD para um meio de pagamento que não habilite CDC.

A descrição de um meio de pagamento é um texto informativo opcional que o descreve a critério do usuário, sem finalidade específica para o ECF, podendo ser consultado com o comando `LeMeioPagamento`.

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

- . **15** (quinze) **meios de pagamento programáveis** (codificados de 0 a 14)

- . O meio de pagamento -2 é **pré-configurado** pelo software básico como **DINHEIRO**. Este meio de pagamento não permite a emissão de cupom de crédito/débito.

A operação de pagamento em um cupom exige a identificação do meio de pagamento, além do valor a ser pago. O valor de cada operação de pagamento efetuada é acumulado em um totalizador específico na memória do ECF. Estes acumuladores são listados nos Relatórios Fiscais X e Z, na seção *Meios de Pagamento*. Após a emissão de um fechamento - Redução Z, os acumuladores dos meios de pagamento são zerados para início do novo dia fiscal.

Os valores lançados nos meios de pagamento de um cupom serão sempre acumulados nos respectivos totalizadores de meios de pagamento. Caso a soma dos valores pagos for maior que o valor total do cupom, a diferença será lançada em um totalizador de **TROCO**, além disso não será aceito mais registro de meio de pagamento no caso da soma dos meios de pagamento igualar ou exceder o total do documento. Portanto, o valor em caixa pode ser obtido somando-se todos os valores acumulados nos meios de pagamento definidos menos o totalizador de troco.

Exemplo:

Valor Total Cupom: R\$ 100,00

Pago meio pagamento 00: R\$ 90,00 (R\$ 90,00 acumulados Meio de Pagam. 00)

Pago meio pagamento 03: R\$ 15,00 (R\$ 15,00 acumulados Meio de Pagam. 03)

Troco: R\$ 5,00 (R\$ 5,00 serão acumulados no totalizador *TROCO*)

IMPORTANTE

Um meio de pagamento somente pode ser redefinido ou excluído após o fechamento diário (Redução Z) e no estado de intervenção técnica. São permitidos no máximo 31 operações de pagamento para cada cupom (fiscal ou não fiscal). Igualando ou excedendo o total do documento, não será aceito mais registros de meios de pagamento para o documento atual.

Os principais comandos relacionados a meios de pagamento são:

Comando	Operação
DefineMeioPagamento	Define um novo meio de pagamento ou redefine um já existente
ExcluiMeioPagamento	Exclui um meio de pagamento previamente definido
LeMeioPagamento	Lê os atributos de um meio de pagamento definido

Há vários [registradores](#) que retornam valores associados aos meios de pagamento. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeMoeda, LeTexto, LeInteiro, etc (ver tópico [Tipos de Dados](#)).

1.2.4.5 Totalizadores Não Fiscais

Um totalizador não fiscal é composto pelas seguintes informações:

Atributo	Descrição
CodNaoFiscal	Código único que identifica um totalizador não fiscal
NomeNaoFiscal	Nome que identifica o totalizador não fiscal
TipoNaoFiscal	Indica se o tipo do totalizador não fiscal refere-se a um valor de entrada no caixa (<i>Yes</i>) – suprimento ou pagamento - ou refere-se a um valor de saída do caixa (<i>No</i>) – sangria ou retirada
DescricaoNaoFiscal	Descrição opcional para o totalizador não fiscal

Pode ser identificado pelo seu código ou pelo seu nome. A descrição do não fiscal é um texto informativo opcional que o descreve sem finalidade para o ECF, podendo ser consultado pelo comando LeNaoFiscal.

A impressora fiscal gerencia **15** (quinze) **totalizadores não fiscais programáveis** (codificados de 0 a 14).

Para cada código não fiscal definido há um contador específico de operação não fiscal correspondente chamado de **CON**, o qual é incrementado a cada emissão de um item não fiscal. Nos Relatórios Fiscais de Leitura X e Redução Z, na seção *Totalizadores Não Fiscais*, são impressos o índice, nome, CON e valor acumulado de cada não fiscal. Após a emissão de um fechamento - Redução Z, os contadores e os acumuladores dos não fiscais são zerados para início do novo dia fiscal.

Os cupons não fiscais permitem o lançamento de valores nos totalizadores não fiscais definidos. É possível classificar o totalizador não fiscal em dois tipos:

- não fiscais de entrada ou
- não fiscais de saída.

Um totalizador não fiscal é dito de entrada, quando o valor acumulado neste refere-se a uma transação onde o valor indica uma entrada de caixa, tal como um suprimento ou pagamento de conta. Ao contrário, um totalizador não fiscal é caracterizado como totalizador de saída quando o valor associado ao mesmo representa uma saída de valor do caixa, tal como uma sangria. O ECF não aceita lançamentos de ambos os tipos de totalizadores em um mesmo cupom não fiscal, entretanto é permitido o registro de vários itens não-fiscais dentro do mesmo documento.

Se o cupom não fiscal for iniciado com um lançamento para um determinado tipo de totalizador, somente totalizadores daquele tipo poderão ser usados no cupom. O tipo de cupom não fiscal em uso pode ser lido através do registrador *TipoDocEmEmissao*. Cupons não fiscais que registram totalizadores de saída não aceitam comandos de pagamento antes da finalização do cupom.

As operações de desconto/acréscimo em item/subtotal e propaganda também são permitidas em cupons não fiscais de entrada. É opcional a impressão dos dados de identificação do comprador no início dos cupons não fiscais, independente de seu tipo.

IMPORTANTE

Um totalizador não fiscal somente pode ser redefinido ou excluído após o fechamento diário

(Redução Z) e no estado de intervenção técnica.

Os principais comandos relacionados a totalizadores não fiscais são:

Comando	Operação
DefineNaoFiscal	Define um novo totalizador não fiscal ou redefine um já existente
ExcluiNaoFiscal	Exclui um totalizador não fiscal previamente definido
LeNaoFiscal	Lê os atributos de um totalizador não fiscal definido

Há vários [registradores](#) que retornam valores associados aos totalizadores não fiscais. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeMoeda, LeTexto, LeInteiro, etc (ver tópico [Tipos de Dados](#)).

TÓPICOS RELACIONADOS:

- [Comprovante Não Fiscal](#)

1.2.4.6 Gerenciais

Um relatório gerencial é um relatório no qual informações livres podem ser enviadas pelo software aplicativo ao ECF. Para que seja possível a emissão de relatórios gerenciais, é necessário primeiramente defini-los.

Um relatório gerencial é definido pelos seguintes atributos:

Atributo	Descrição
CodGerencial	Código único que identifica um relatório gerencial
NomeGerencial	Nome que identifica o relatório gerencial
DescricaoGerencial	Descrição opcional para o relatório gerencial

Pode ser identificado pelo seu código ou pelo seu nome. A descrição de um gerencial é um texto informativo opcional que o descreve, sem função para o ECF mas que pode ser consultado com o comando LeGerencial.

A impressora fiscal gerencia **20** (vinte) **relatórios gerenciais programáveis** (codificados de 0 a 19). Apenas os gerenciais carregados podem ser emitidos. Não existe gerencial pré-definido pelo ECF. Para cada gerencial definido há um contador específico correspondente chamado de **CER** que é incrementado a cada impressão de um gerencial. Nos Relatórios Fiscais de Leitura X e Redução Z, na seção *Relatório Gerencial*, são impressos o índice, o nome do gerencial e o CER correspondente. Após a emissão de um fechamento - Redução Z, os contadores dos gerenciais são zerados para início do novo dia fiscal.

IMPORTANTE

Um gerencial somente pode ser redefinido ou excluído após o fechamento diário (Redução Z) e no estado de intervenção técnica.

Os principais comandos relacionados a relatórios gerenciais são:

Comando	Operação
DefineGerencial	Define um relatório gerencial ou redefine um já existente
ExcluiGerencial	Exclui um relatório gerencial previamente definido
LeGerencial	Lê os atributos de um relatório gerencial definido

Há vários [registradores](#) que retornam valores associados aos relatórios gerenciais. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeMoeda, LeTexto, LeInteiro, etc (ver tópico [Tipos de Dados](#)).

1.2.4.7 Departamentos

Um departamento é composto pelas seguintes informações:

Atributo	Descrição
CodDepartamento	Código único que identifica um departamento
NomeDepartamento	Nome que identifica o departamento
DescricaoDepartamento	Descrição opcional para o departamento
QuantidadeVendas	Quantidade vendida no departamento (zerada após RZ)
ValorVendas	Valor vendido no departamento (zerado após RZ)

Pode ser identificado pelo seu código ou pelo seu nome. A descrição de um departamento é um texto informativo opcional que o descreve podendo ser consultado pelo comando LeDepartamento..

A impressora fiscal gerencia **19** (dezenove) **departamentos programáveis** (codificados de 1 a 19) e **1** (um) **departamento pré-configurado** como **Combustível** codificado/acessado com o código 0.

Os campos quantidade e valor vendidos somente serão acumulados pelo ECF quando o respectivo departamento for indicado no comando de venda de item. Esta característica permite criar categorias de produtos através de totalizadores controlados automaticamente pelo ECF.

Os departamentos podem ser impressos em Relatórios Gerenciais ou seus dados lidos através de registradores. Após a emissão de um fechamento - Redução Z, os valores acumulados nesses campos são zerados.

IMPORTANTE

Um departamento somente pode ser alterado ou excluído após o fechamento diário (Redução Z) e no estado de intervenção técnica.

Os principais comandos relacionados a departamentos são:

Comando	Operação
DefineDepartamento	Define um departamento ou redefine um já existente
ExcluiDepartamento	Exclui um departamento previamente definido
LeDepartamento	Lê os atributos de um departamento definido
ImprimeDadosDepartamento	Imprime os dados de departamento em um relatório gerencial

Há vários [registradores](#) que retornam valores associados aos departamentos. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeMoeda, LeTexto, LeInteiro, etc (ver [Tipos de Dados](#)).

1.2.5 Criptografia do GT

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

1.2.6 Símbolos Fiscais

Todo valor impresso pela impressora fiscal que é armazenado no GT (Grande Total) é identificado por um símbolo característico.

Símbolo de acumulação do total venda de item no GT

1.2.7 Tipos de Documentos

Os principais documentos emitidos pelo ECF são:

[Cupom Fiscal](#) (CF)

[Comprovante Não Fiscal \(CNF\)](#)
[Comprovante de Crédito/Débito \(CCD\)](#)
[Relatório Gerencial](#)
[Relatórios Fiscais](#)

Existe um contador gerenciado pelo ECF, chamado de Contador de Ordem de Operação (COO), que é impresso no cabeçalho de cada cupom emitido. Este contador é zerado apenas quando há [troca de proprietário](#).

1.2.7.1 Cupom Fiscal (CF)

O Cupom Fiscal (CF) é o documento onde são registradas as vendas. Existe um contador gerenciado pelo ECF, chamado de Contador de Cupons Fiscais (CCF), o qual é impresso no cabeçalho de cada CF. O COO também é incrementado a cada CF emitido.

As principais partes de um CF são:

- cabeçalho do cupom: onde são impressos os dados do clichê
- cabeçalho dos itens de venda, identificando as colunas impressas
- área de transações: itens vendidos e operações sobre itens
- totalização do cupom
- área de transações sobre o subtotal
- indicação dos meios de pagamento
- informações adicionais
- rodapé do cupom, podendo ser informado o operador

Pode ser impresso o Cupom Adicional logo após a emissão de um CF. O Cupom Adicional contém informações resumidas sobre o CF emitido, tais como: COO, CCF e valor totalizado no CF.

Um CF pode ser cancelado durante sua emissão ou imediatamente após a sua emissão. O ECF contabiliza o número de CF cancelado através do Contador de Cupom Fiscal Cancelado (CFC).

Os principais comandos de protocolo associados a um CF são:

Comando	Operação
AbreCupomFiscal	Imprime os cabeçalhos do cupom e dos itens
VendeItem	Emite uma venda de um item
AcresceItemFiscal	Emite uma transação de acréscimo ou desconto sobre um item vendido ou cancela esse tipo de transação
CancelaItemFiscal	Cancela um item previamente emitido
AcresceSubtotal	Emite uma transação de acréscimo ou desconto sobre o subtotal do cupom ou cancela esse tipo de transação
PagaCupom	Emite os meios de pagamento com os valores para pagamento do cupom
AbreGaveta	Comanda a abertura da gaveta de valores conectada na porta do ECF
ImprimeTexto	Opcionalmente imprime Mensagem Promocional
IdentificaConsumidor	Opcionalmente imprime dados para identificação do consumidor, quando não impresso na abertura do CF
EncerraDocumento	Imprime o rodapé do cupom, encerrando o documento
ImprimeCupomAdicional	Opcionalmente, permite a impressão de um cupom adicional ao CF, contendo as informações que referenciam o mesmo além do valor totalizado no CF
CancelaCupom	Cancela o último cupom emitido ou em emissão

Há vários [registradores](#) que retornam informações importantes referentes a CF, tais como totais líquido e bruto do cupom, itens emitidos, entre outras informações. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeMoeda, LeTexto, LeInteiro, etc (ver tópico [Tipos de Dados](#)).

NOTA

O cancelamento de um CF encerrado gera um cupom chamado de Cupom Fiscal Cancelamento. Neste caso, o COO, CCF e CFC são incrementados de uma unidade.

1.2.7.2 Comprovante Não Fiscal (CNF)

Um Comprovante Não Fiscal permite o acúmulo de valores em [totalizadores não fiscais](#). Um CNF pode ser usado apenas para registro de totalizadores não fiscais de entrada ou apenas para registro de totalizadores não fiscais de saída. Ambos os tipos de totalizadores não podem ser emitidos em um mesmo CNF.

O contador gerenciado pelo ECF para contabilizar CNF é chamado de Contador Geral Não Fiscal (**GNF**). Para cada transação de acúmulo de valor realizada para um totalizador não fiscal, há um contador específico chamado de Contador de Operação Não Fiscal (**CON**). O COO também é incrementado a cada emissão de um CNF.

As principais partes de um CNF são:

- cabeçalho do cupom: onde são impressos os dados do clichê
- área de transações: registro de valores nos totalizadores não fiscais e operações sobre itens (este último apenas em CNF de entrada)
- totalização do cupom (apenas em CNF de entrada)
- área de transações sobre o subtotal (apenas em CNF de entrada)
- indicação dos meios de pagamento (apenas em CNF de entrada)
- informações adicionais (apenas em CNF de entrada)
- rodapé do cupom, podendo ser informado o operador

Os principais comandos de protocolo associados a um CNF são:

Comando	Operação
AbreCupomNaoFiscal	Imprime o cabeçalho do cupom
EmiteItemNaoFiscal	Acumula um valor em um totalizador não fiscal
AcresceItemNaoFiscal	Emite uma transação de acréscimo ou desconto sobre um totalizador não fiscal ou cancela esse tipo de transação
CancelaItemNaoFiscal	Cancela uma transação (item) não fiscal previamente emitida
AcresceSubtotal	Emite uma transação de acréscimo ou desconto sobre o subtotal do cupom ou cancela esse tipo de transação
PagaCupom	Emite os meios de pagamento com os valores para pagamento do cupom
AbreGaveta	Comanda a abertura da gaveta de valores conectada na porta do ECF
EncerraDocumento	Imprime o rodapé do cupom, encerrando o documento
CancelaCupom	Cancela o último cupom emitido

Há vários [registradores](#) que retornam informações importantes referentes a um CNF, tais como totais líquido e bruto do cupom, itens emitidos, entre outras informações. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeMoeda, LeTexto, LeInteiro, etc (ver tópico [Tipos de Dados](#)).

1.2.7.3 Comprovante de Crédito ou Débito (CCD)

Este comprovante é de natureza não fiscal e é usado tipicamente para operações de TEF (Transferência Eletrônica de Fundos) em operações como impressão de comprovantes de Cartão de Crédito ou Cartão de Débito. Permite imprimir informações gerais sobre o [meio de pagamento](#) que habilitou sua emissão. O Comprovante de Crédito ou Débito (**CCD**) libera a impressão dos dados (texto livre) para o aplicativo durante **2 minutos**. Após este tempo, caso o comprovante não tenha sido encerrado pelo aplicativo, o ECF o encerrará automaticamente.

O contador gerenciado pelo ECF para contabilizar cada CCD emitido é chamado de Contador de Comprovante de Crédito ou Débito (**CDC**). Além deste contador, o GNF e o COO também são incrementados a cada emissão de um CCD.

Há também um contador específico controlado pelo ECF para gerenciar a quantidade de CCD habilitados para emissão (através de meio de pagamento), porém não impressos até o momento. Este contador é chamado de Comprovante de Crédito ou Débito Não Emitidos (**NCN**) e é listado em alguns relatórios fiscais (Leitura X e Redução Z). Este contador é zerado após a Redução Z (fechamento diário).

As principais partes de um CCD são:

- cabeçalho do cupom: onde são impressos os dados do clichê
- identificação dos dados do CCD impressos pelo software básico, contendo: o meio de pagamento e o COO que habilitaram a impressão deste documento, valores da compra e do pagamento do COO de origem e o número de parcelas
- área de impressão do texto livre a ser enviado pelo aplicativo, opcionalmente
- rodapé do cupom, podendo ser informado o operador

Um CCD pode ser estornado através de comando específico. Um cupom de estorno incrementa os contadores COO, CDC e GNF e possui as mesmas partes de um CCD.

Outras operações permitidas para um CCD são: reimpressão de via e emissão de nova via. Ambas não alteram os contadores COO, CDC e GNF e necessitam apenas um comando para impressão do documento. No caso do primeiro comando, a mensagem "REIMPRESSAO" é registrada no cupom. No segundo caso, apenas o número que identifica a via do CCD é incrementada com relação a última impressão.

Os principais comandos de protocolo associados a um CCD são:

Comando	Operação
AbreCreditoDebito	Imprime o cabeçalho do cupom e a identificação dos dados do CCD
ImprimeTexto	Imprime o texto enviado pelo aplicativo
ReimprimeViaCreditoDebito	Emite uma via idêntica ao último CCD impresso, informando que é uma Reimpressão no texto do comprovante
EmiteViaCreditoDebito	Emite uma nova via referente ao último CCD impresso, mantendo todos os dados da via original, acrescentando apenas a numeração da via
EncerraDocumento	Imprime o rodapé do cupom, encerrando o documento
EstornaCreditoDebito	Estorna um CCD previamente impresso

Há vários [registradores](#) que retornam informações importantes referentes a um CCD. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeMoeda, LeTexto, LeInteiro, etc (ver tópico [Tipos de Dados](#)).

NOTA

A emissão de uma nova via de CCD somente pode ocorrer seqüencialmente a emissão da via anterior ou sua reimpressão. É permitida apenas uma reimpressão de via de CCD, e esta somente pode ocorrer seqüencialmente a emissão da respectiva via. A emissão de um CCD decrementa o contador NCN de uma unidade. O estorno de um CCD acresce uma unidade a este contador. Sendo um CCD emitido imediatamente após um CF ou CNF, esses somente poderão ser cancelados se o CCD for estornado e o comando de cancelamento de cupom for emitido na seqüência do estorno de CCD.

1.2.7.4 Relatório Gerencial

Este comprovante é de natureza não fiscal e permite imprimir informações gerais de interesse do aplicativo. O ECF libera a impressão dos dados (texto livre) para o aplicativo durante **2 minutos**, conforme estabelecido em legislação. Após este tempo, caso o relatório não tenha sido encerrado pelo aplicativo, o ECF o encerrará automaticamente.

O contador gerenciado pelo ECF para contabilizar cada Relatório Gerencial emitido é chamado de Contador Geral de Relatório Gerencial (**GRG**). O COO e GNF também são incrementados a cada emissão deste documento.

Há também um contador específico controlado pelo ECF para cada tipo de relatório gerencial emitido. Este contador é chamado de Contador Específico de Relatório Gerencial (**CER**) e aparece nos relatórios fiscais de Leitura X e Redução Z.

As principais partes de um Relatório Gerencial são:

- cabeçalho do cupom: onde são impressos os dados do clichê
- identificação do Relatório Gerencial (nome)
- área de impressão do texto livre a ser enviado pelo aplicativo
- rodapé do cupom, podendo ser informado o operador

Os principais comandos de protocolo associados a um Relatório Gerencial são:

Comando	Operação
AbreGerencial	Imprime o cabeçalho do cupom e a identificação do relatório
ImprimeTexto	Imprime o texto enviado pelo aplicativo
EncerraDocumento	Imprime o rodapé do cupom, encerrando o documento

Há vários [registradores](#) que retornam informações importantes referentes a um Relatório Gerencial. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeMoeda, LeTexto, LeInteiro, etc (ver tópico [Registradores do Protocolo FiscNet](#)).

NOTA

É obrigatória a impressão do texto "NÃO E DOCUMENTO FISCAL" pelo *software* básico a cada 10 (dez) linhas de texto livre impressas.

1.2.7.5 Relatórios Fiscais

São documentos com informações para controle fiscal. Os Relatórios Fiscais são:

Leitura X

Redução Z

Leitura da Memória Fiscal

Uma Leitura X (LX) imprime uma imagem dos principais valores fiscais controlados pelo ECF no momento da impressão. O COO é incrementado de uma unidade ao ser emitido este documento. A LX é emitida a partir de um único comando enviado ao ECF.

As principais partes de uma LX são:

- cabeçalho do cupom: onde são impressos os dados do clichê
- seção de Contadores: valores de cada contador gerenciado pelo ECF
- seção de Totalizadores Fiscais: valores sobre venda bruta, líquida, descontos, etc.
- seção de ICMS: informações sobre as situações tributárias de ICMS definidas
- seção de ISSQN: informações sobre as situações tributárias de ISSQN definidas
- seção de Não Tributados: situação dos acumuladores Não Tributados (ICMS e ISSQN)
- seção de Totalizadores Não Fiscais: código, nome, COM e valor acumulado de cada totalizador não fiscal definido, descontos, acréscimos e cancelamentos não fiscais
- seção de Relatório Gerencial: código, nome e CER de cada relatório gerencial definido.
- seção de Meios de Pagamento: código, nome e valor acumulado de cada meio de pagamento definido
- informações gerais, contendo NCN, controle de tempos e números de séries
- rodapé do cupom

A Redução Z (RZ) imprime um documento contendo todos os dados de uma LX, porém com algumas

funções adicionais, a saber:

- Grava as informações de controle diário na Memória Fiscal
- Zera as informações de controle diário armazenadas em RAM (contadores, acumuladores, etc)
- Inicia um novo dia fiscal

A diferença do texto registrado entre uma LX e RZ está, basicamente:

- no título do cupom (ao invés de Leitura X, Redução Z),
- na impressão adicional da data de movimento reduzida
- na impressão do "bitmap" contendo informações sobre os cupons emitidos no dia

Ao emitir uma RZ, o COO é incrementado de uma unidade. O Contador de Reduções Z (**CRZ**) também é incrementado de uma unidade. A RZ também é emitida a partir de um único comando enviado ao ECF.

Após emitir uma RZ, não será possível gerar movimento (emitir CF, CNF, etc) com relação ao dia fiscal encerrado.

A Leitura da Memória Fiscal (LMF) informa os valores armazenados na Memória Fiscal (MF), tanto relacionados a inicializações quanto a reduções efetuadas.

As principais partes de uma LMF são:

- informação sobre os contadores CRZ, CRO (Contador de Reinício de Operação, isto é, número de Intervenções Técnicas realizadas) e CFD (Contador de Fita Detalhe impressas)
- registros de Reinício de Operação: CRO, data e hora da ocorrência
- registros de Impressão de Fita Detalhe: data, hora, COO inicial e COO final da impressão
- registro de proprietários (usuários do ECF): número do usuário, CRO inicial, data e hora do registro, inscrições e GT (Grande Total)
- informações sobre Reduções diárias: CRZ, CRO, COO, data, hora, venda bruta, descontos, cancelamentos e operações não fiscais.
- informações gerais: número de reduções restantes, versão *software* básico inicial e alterações (caso haja), número de série da MFD e codificações do GT
- rodapé do cupom

Os principais comandos de protocolo associados aos Relatórios Fiscais são:

Comando	Operação
EmiteLeituraX	Emite a Leitura X impressa ou via serial
EmiteLeituraMF	Emite a Leitura da Memória Fiscal impressa ou via serial
EmiteReducaoZ	Imprime a Redução Z

Há vários [registradores](#) que retornam informações importantes referentes aos dados impressos nos Relatórios Fiscais, tais como contadores, totalizadores, data de movimento, flags de dia aberto ou fechado, etc. Estes podem ser consultados a qualquer momento através dos comandos de protocolo associados a seu tipo: LeMoeda, LeTexto, LeInteiro, LeIndicador, etc (ver tópico [Tipos de Dados](#)).

1.2.7.6 Outros Documentos

Tratando-se de documentos impressos, há outros ainda não citados nos tópicos acima. Um deles é o cupom de **Estorno de Meio de Pagamento**.

A operação de estorno de meio de pagamento permite a transferência de valores acumulados entre dois meios de pagamento, debitando do meio de pagamento de origem e creditando no meio de pagamento destino. Os parâmetros de uma operação de estorno de meio de pagamento são: identificação do meio de pagamento origem, identificação do meio de pagamento destino e valor da transferência. Abaixo são listadas as condições de uso e importantes considerações a respeito deste tipo de operação:

- cada cupom de estorno permite uma transferência de valores entre 2 (dois) meios de pagamento

- as operações de estorno somente podem ser realizadas imediatamente após a emissão de um cupom em que houve pelo menos um meio de pagamento
- podem ser emitidos vários cupons de estornos de meios de pagamento sequencialmente
- os valores e o meios de pagamento de origem que poderão ser envolvidos em operações de estorno devem referir-se apenas ao último cupom emitido com meios de pagamento
- se o meio de pagamento destino permitir a emissão de um CCD, a emissão deste último será habilitado e o NCN será incrementado de uma unidade
- se o meio de pagamento origem permite a emissão de CDC, e por outro lado, se o meio de pagamento destino não permite a emissão de CDC, o contador NCN será decrementado de uma unidade.
- após a emissão de um cupom de estorno de meio de pagamento, o cupom que originou os meios de pagamento não poderá ser cancelado
- não será possível emitir qualquer CCD habilitado pelo cupom em que houve estorno de meio de pagamento sem que seja na seqüência do estorno

O outro tipo de impressão é a **Leitura da Memória de Fita Detalhe**. Esta impressão representa a segunda via armazenada na MFD (Memória de Fita Detalhe) no ECF. Esta leitura pode ser tanto impressa quanto serial. Quando realizada de forma impressa, exige que o ECF esteja em **MIT** (Modo de Intervenção Técnica).

Os comandos de protocolo referente aos relatórios citados neste tópico são:

Comando	Operação
EstornaMeioPagamento	Emitir um cupom de estorno de meio de pagamento
EmiteLeituraFitaDetalhe	Emitir a Leitura da Memória de Fita Detalhe (segunda via) impressa ou via serial

1.2.7.7 Cheque e Autenticação

Alguns modelos de ECF possuem um mecanismo específico para a impressão de cheque e autenticação.

A autenticação é realizada pela mesma estação (ou mecanismo) de cheque e pode ser impressa para cada valor emitido no ECF. Em função de exigência da legislação, o *software* básico limita em 5 (cinco) autenticações para cada valor impresso. A autenticação é impressa em 2 (duas) linhas, sendo possível que o *software* aplicativo envie um texto adicional a ser impresso. A posição das linhas de autenticação no documento é pré-definida e não pode ser alterada.

O posicionamento de impressão dos campos no cheque é configurável, permitindo o ajuste para todos os modelos existentes. Estas posições (ou coordenadas) são definidas em unidades de décimos de milímetros, permitindo um ajuste fino.

Basicamente, todos os modelos permitem a impressão de cheques sem maiores recursos adicionais. Alguns permitem os recursos de chancela e cruzamento de cheque. Outros permitem a leitura do código CMC7.

Os comandos de protocolo referente a autenticação e impressão de cheque são:

Comando	Operação
ImprimeAutenticacao	Emita uma autenticação
ImprimeCheque	Emita um cheque
ChancelaCheque	Chancela o cheque
CruzaCheque	Cruza o cheque
ExpulsaDocumento	Retira o documento do mecanismo de cheque
LeTexto	Com um dos parâmetros: 'CMC7Memoria', 'CMC7Documento', 'CMC7Campo1', 'CMC7Campo2', 'CMC7Campo3' permite ler o CMC7 do documento. A seqüência é: 'CMC7Documento' para leitura direta do documento e as demais para leitura do campo lido separadamente ou o campo por inteiro.

1.2.8 Recursos

Além da emissão de documentos, neste tópico serão listados alguns outros recursos disponibilizados pelo *software* básico.

1.2.8.1 Horário de Verão

É permitido o ajuste do horário de verão (entrada ou saída) sem que haja necessidade de entrar em [MIT](#). As condições são:

- o dia deve estar fechado
- o comando é aceito somente uma vez por dia

Não será possível realizar a entrada e a saída do horário de verão no mesmo dia fiscal.

Quando o ECF estiver em horário de verão, a hora sempre será impressa seguida do caracter 'V'. Ao sair do horário de verão, a hora será impressa sem este símbolo.

O comando de protocolo referente ao horário de verão é:

Comando	Operação
AcertaHorarioVerao	Comanda a entrada ou a saída do horário de verão

1.2.8.2 Configurações

O *software* básico permite algumas configurações para facilitar seu uso.

As configurações podem ser lidas ou definidas através de registradores. O acesso aos registradores deve ser feito através dos [comandos do protocolo LeXXXX](#) e [EscreveXXXX](#), onde **XXXX** indica o tipo do registrador a ser lido (ver o tópico [Tipos de Dados](#)). Abaixo está a lista de algumas das principais configurações:

- ContrasteBobina
- EspacamentoDocumentos
- EspacamentoLinhas
- SimboloMoeda
- NomeMoedaSingular
- NomeMoedaPlural
- TextoLivre

A tabela de registradores disponibilizadas pelo protocolo está no tópico [Registradores do Protocolo FiscNet](#).

1.2.8.3 Mecanismo

Os recursos de mecanismo dependem do tipo do mecanismo usado pelo modelo de ECF. É possível verificar o tipo de mecanismo usado pelo modelo de ECF através da leitura do registrador

Mecanismo Bobina. Os valores possíveis para este registrador estão listados no

[Retorno dos registradores.](#)

Os recursos básicos implementados na maioria dos mecanismos envolve, além do avanço de papel, a impressão de caracteres expandido, enfatizado e códigos de barras. Um recurso mais específico é o corte de papel.

Para os modelos de ECF que possuem mecanismo para impressão de cheques e autenticação, pode ser lido o registrador **Mecanismo Pagina.** Os valores possíveis para este registrador estão listados no [Retorno dos registradores.](#)

O *software* básico tratará adequadamente quando for recebido um comando para um recurso indisponível para o mecanismo em uso.

Os principais comandos do protocolo referentes aos mecanismos são:

Comando	Operação
AvancaPapel	Avança o papel no mecanismo impressão de documentos
CortaPapel	Corta o papel no mecanismo de impressão de documentos
ExpulsaDocumento	Expulsa o documento do mecanismo de impressão de cheques/autenticação

1.2.8.4 Leitura do Texto Impresso

Um recurso muito útil do *software* básico é a leitura da saída impressa. O texto impresso na bobina é disponibilizado ao *software* aplicativo através do comando `LeImpressao`.

Esta característica aplica-se muito bem a *softwares* aplicativos que montam a tela com os dados impressos na bobina do ECF, podendo reduzir o tempo gasto no desenvolvimento desta tarefa por parte do aplicativo.

O ECF armazena o texto impresso em uma área de memória com cerca de 16 Kbytes, sendo que cada comando `LeImpressao` pode retornar até 4000 caracteres. Caso não seja efetuada a leitura da saída impressa até o esgotamento dos 16 Kbytes de memória, haverá sobreposição dos dados no *buffer*, sendo retornados sempre os dados mais antigos impressos pelo ECF.

O Comando `LeImpressao` retornará sempre os dados impressos mais antigos e não lidos ainda pelo programa aplicativo até o limite de 4000 bytes. Quando não houverem mais dados impressos que não foram lidos pelo aplicativo, o comando `LeImpressao` retornará um texto vazio como retorno.

O comando do protocolo referente a leitura da saída impressa é:

Comando	Operação
LeImpressao	Lê a saída impressa

1.2.8.5 Modos e Comandos de Impressão

O *software* básico aceita alguns **comandos de impressão** baseados no protocolo EPSON® ESC/POS® que permitem realizar algumas formatações sobre o texto impresso. Algumas delas dizem respeito ao **modo de impressão**.

Os modos de impressão aceitos pelo *software* básico, estão:

- Normal
- Expandido
- Negrito
- Código de Barras

Os comandos de impressão somente são interpretados dentro de documentos que permitem a impressão de texto livre, tais como propagandas em cupons, textos de relatórios gerenciais e de CCD. Abaixo, a lista dos comandos de impressão do *software* básico:

- **Line Feed**: Passa o controle de impressão para a coluna 1 da linha seguinte.
ASCII: LF

Hexa: 0A
Decimal: 10
Protocolo Logger II: \x0A

- **Carriage Return:** Sem efeito.

ASCII: CR
Hexa: 0D
Decimal: 13
Protocolo Logger II: \x0D

- **Comandos ESC:**

Comando de formatação múltipla

ASCII: ESC ! n
Hexa: 1B 21 n
Decimal: 27 32 n
Protocolo Logger II: \x1B ! n

Onde n é um valor inteiro respeitando a seguinte tabela:

n	Formatação
0	Caracter normal
8	Negrito
32	Expandido

Exemplo: Para ativar o modo expandido, deve-se enviar a seguinte seqüência de caracteres antes do texto a ser impresso na Logger II:

\x1B!\x20Este texto será expandido\x1B!\x00

(Observe que $n = 32$ para ligar o modo negrito e 0 para desligá-lo; está representado em hexadecimal como \x20 e 0x00 respectivamente no comando)

Liga/desliga o modo negrito

ASCII: ESC E n
Hexa: 1B 45 n
Decimal: 27 69 n
Protocolo Logger II: \x1B E n

Onde n é um valor inteiro respeitando a seguinte tabela:

n	Significado
0	Desliga negrito
1	Liga negrito

Exemplo: Para ativar o modo negrito, deve-se enviar a seguinte seqüência de caracteres antes do texto a ser impresso na Logger II:

\x1BE\x1 texto negrito \x1BE\x0 texto não negrito

- **Comandos GS:**

Define posição de impressão do texto HRI nos próximos códigos de barras (texto HRI é o texto que acompanha o código de barras). Este comando tem efeito apenas para o ECF com mecanismo Epson® TM-H6000.

ASCII: GS H n
Hexa: 1D 48 n
Decimal: 29 72 n
Protocolo Logger II: \x1D H n

Onde n é um valor inteiro respeitando a seguinte tabela:

<i>n</i>	Significado
0	Não imprimir texto HRI
1	Imprimir texto HRI acima do código de barras
2	Imprimir texto HRI abaixo do código de barras
3	Imprimir texto HRI acima e abaixo do código de barras

Define o fonte do Código de Barras.

Este comando tem efeito apenas para o ECF com mecanismo Epson®TM-H6000.

ASCII:	GS	F	<i>n</i>
Hexa:	1D	66	<i>n</i>
Decimal:	29	102	<i>n</i>
Protocolo Logger II:	\x1D	f	<i>n</i>

Onde *n* é um valor inteiro respeitando a seguinte tabela:

<i>n</i>	Significado
0	Font A
1	Font B

Maiores informações acerca dos caracteres de controle devem ser obtidas no manual do protocolo ESC/POS® da Epson®.

Define a altura do Código de Barras.

ASCII:	GS	h	<i>n</i>
Hexa:	1D	68	<i>n</i>
Decimal:	29	104	<i>n</i>
Protocolo Logger II:	\x1D	h	<i>n</i>

Onde *n* é um valor inteiro entre 1 e 255 definindo a altura das barras.

Comando com o Código de Barras a ser impresso

ASCII:	GS	k	<i>m</i>	<i>d1..dk</i>	NUL
Hexa:	1D	6B	<i>m</i>	<i>d1..dk</i>	0
Decimal:	29	107	<i>m</i>	<i>d1..dk</i>	0
Protocolo Logger II:	\x1D	k	<i>m</i>	<i>d1..dk</i>	\x0

Onde *m* define o sistema de código de barras, conforme a tabela abaixo:

<i>m</i>	Sistema de Código de Barras
0	UPC-A
1	UPC-E
2	EAN13
3	EAN8
4	CODE39
5	ITF (interleaved 2 of 5)
6	CODABAR

Apenas a opção 5 (ITF) pode ser utilizada em todos os modelos de ECF da série Logger II, os demais podem ser usados apenas nos modelos baseados no mecanismo Epson® TM-H6000. O comando de impressão do código de barras deve ser o primeiro texto impresso na linha. Logo o caractere GS que inicia a sequência de impressão do código de barras deve ser o primeiro caractere da linha, caso contrário o comando será ignorado. Recomenda-se a finalização da linha de impressão após a impressão do código de barras com um caractere LF (Line Feed).

Maiores informações acerca dos caracteres de controle devem ser obtidas no manual do protocolo ESC/POS® da Epson®.

d1..dk são os dados a serem impressos.

Define a largura do Código de Barras.

ASCII:	GS	w	n
Hexa:	1D	77	n
Decimal:	29	119	n
Protocolo Logger II:	\x1D	w	n

Onde n é um valor inteiro que especifica a largura do código. Maiores informações acerca dos comandos e caracteres de controle devem ser obtidas no manual do protocolo ESC/POS® da Epson®.

1.2.9 Estados do Software Básico

O *software* básico define um conjunto de estados com o objetivo de controlar o fluxo de funcionamento das operações realizadas.

Os estados auxiliam no entendimento dos retornos informados durante a execução de uma operação. Um exemplo típico está no controle dos comandos que podem ser aceitos de acordo com a situação. Por exemplo, o *software* básico deve evitar a emissão de um relatório fiscal enquanto houver um cupom fiscal aberto.

É importante que o *software* aplicativo observe constantemente o estado atual do *software* básico para controle dos comandos que podem ser enviados. Uma tabela contendo todos os estados do *software* básico está listada no [Tabela de Estados do software básico](#).

O comando de protocolo associado à leitura de estado é o comando '**LeInteiro**'.

TÓPICOS RELACIONADOS:

- [Tabela de estados do software básico](#)

1.2.10 Indicadores

Os indicadores, também chamados de *flags*, informam se uma determinada condição está ativa ou não naquele momento. Ao ser comandado, o *software* básico confere o estado atual e se os indicadores para aquela operação permitem a sua execução.

O *software* aplicativo também deve levar em consideração os indicadores da impressora fiscal para controlar o fluxo de emissão de documentos. Um exemplo típico de indicador é a condição do ECF estar ou não em [MIT](#).

Uma tabela contendo todos os indicadores do *software* básico está listada no [Indicadores do Software Básico](#).

O comando de protocolo apropriado para leitura de estado é o '**LeIndicador**'

1.2.11 Leitura de Registradores

Registradores são informações armazenadas na memória do ECF. São muito úteis para o *software* aplicativo, podendo servir para fins tais como: conferência de subtotal do cupom atual, valores acumulados nos totalizadores fiscais e não fiscais, total de descontos ou acréscimos efetuados no dia ou no cupom atual, etc.

Um registrador é definido pelos seguintes atributos:

- Nome
- Tipo de Dado
- Tipo de Acesso
- Índice Mínimo
- Índice Máximo

Cada registrador é identificado unicamente pelo campo *Nome*. Os demais campos são informações complementares importantes para definir a forma como o mesmo será acessado.

Tipo de Dado especifica o tipo de dado de cada registrador. Cada tipo de registrador pode ser lido ou

escrito apenas pelo respectivo comando 'Le_' ou 'Escreve_', por exemplo, um registrador do tipo Moeda pode ser lido apenas através do comando LeMoeda e escrito apenas através do comando EscreveMoeda.

Tipo de Acesso identifica o meio de acesso ao registrador: escrita (W), leitura (R) ou ambos (RW). Por exemplo, o registrador que representa o símbolo da moeda possui o *Tipo de Acesso* RW. Ao escrever neste registrador, está sendo redefinido o símbolo da moeda.

Um registrador pode representar um vetor com n elementos. Neste caso, *Índice Mínimo* e *Índice Máximo* representam os limites inferior e superior do vetor. Por exemplo, pode-se ler qualquer totalizador de qualquer situação tributária através do registrador 'TotalAliquotaDia' indexando-o com o respectivo índice da seguinte forma: 'TotalAliquotaDia[0]' representa o total vendido na alíquota programável 0 (zero) e assim sucessivamente.

Para realizar a leitura de um registrador no ECF, deve-se, primeiramente, saber o tipo de dado do registrador desejado. Todos os tipos de dados reconhecidos pelo ECF estão listados no tópico [Tipos de Dados](#).

Após a identificação do tipo de dado, será necessário saber o nome do registrador. No tópico [Registadores](#) são listados todos os registradores disponibilizados pelo ECF, informando o tipo de cada um.

Finalmente, para ler um registrador, deve-se usar a função específica para leitura daquele tipo de dado: 'LeMoeda', 'LeInteiro', 'LeTexto', etc. De forma análoga a escrita: 'EscreveMoeda', 'EscreveInteiro', 'EscreveTexto', etc.

Para maiores informações a respeito dos comandos e sua sintaxe, ver o capítulo que trata do protocolo de comunicação entre o software aplicativo e o ECF, além do tópico

[Comandos do Protocolo FiscNET](#).

1.2.12 Memória de Fita Detalhe (MFD)

As impressoras fiscais da série Logger II, assim como a Logger I, caracterizam-se pela fita detalhe eletrônica, dispensando a fita detalhe em papel. Todos os documentos impressos, são armazenados na MFD, com exceção daqueles facultados pela a legislação.

Fisicamente, a fita detalhe eletrônica é representada por um conjunto de memórias (não voláteis) e um conjunto de *hardware* que protege contra o apagamento ou a alteração dessa memória. Também chamada de **Memórias Logger**, **Memória de Fita Detalhe (MFD)** ou ainda **Cartucho de Memória**. Um ECF Logger II pode ter até 3 (três) Cartuchos de Memória instalados, sendo que, obrigatoriamente na placa do primeiro cartucho, estará presente a Memória Fiscal (MF). As MFD são resinadas e não removíveis. São ligadas a placa fiscal através de um cabo tipo *flat*.

A capacidade de armazenamento de cada MFD, pode ser de 8Mb, 16Mb, 32Mb ou 64 Mb. A quantidade de memória necessária para cada estabelecimento dependerá basicamente do volume e do tipo de documentos emitidos. A quantidade de documentos armazenados em uma mesma quantidade de memória dependerá também do tipo de uso feito pelo usuário e pelo programa aplicativo. Os dados impressos e armazenados na MFD são compactados pelo *firmware* da impressora fiscal para que a durabilidade do cartucho seja estendida ao máximo. Uma compactação eficaz, entretanto, depende também de um uso racional do ECF por parte do usuário. As recomendações para um uso otimizado do algoritmo de compactação podem ser consultadas no tópico [Recomendações de Uso da MFD](#) deste documento.

O ECF bloqueia suas operações se não houver a presença de um cartucho instalado em condições adequadas de funcionamento. Também bloqueia quando houver o esgotamento da capacidade de armazenamento dos dados na memória de fita detalhe.

Um item de fundamental importância e exigido pela Legislação quanto a segurança dos dados armazenados na MFD, chamado de *Bitmap*, deve ter a capacidade de representar todos os

documentos impressos no através de símbolos impressos ao final da Redução Z. Os símbolos impressos neste documento fiscal podem ser lidos através de um scanner e de um software específico para possibilitar a consulta aos documentos emitidos no dia relativo ao fechamento. De acordo com o volume de documentos emitidos, estes símbolos poderão ser impressos em blocos de símbolos, de modo a facilitar a recuperação dos dados. Para a consulta destes documentos, deve-se:

A partir de um scanner, gera-se um arquivo no formato .bmp para cada bloco de símbolos impresso na redução Z;

Cada arquivo .bmp lido deve ser submetido a uma análise e conseqüente conversão, a partir do software específico de leitura do logger;

Após a conversão, os cupons são gerados e armazenados em arquivos e podem ser consultados.

É possível realizar a leitura da MFD impressa na bobina de papel ou via serial através de um equipamento conectado ao ECF. Quanto à leitura impressa, somente pode ocorrer em MIT, enquanto que via serial não há esta necessidade. Além disso, o número de impressões da MFD em cada Intervenção Técnica é limitada a duas.

Os dados armazenados na MFD apresentam uma cópia fiel de todos os documentos emitidos no ECF.

1.2.13 Perda da Memória RAM

Na memória RAM são armazenadas algumas informações importantes para agilizar o processo das operações realizadas pelo *software* básico. Tratando-se de uma memória volátil, protegida por bateria durante o desligamento do ECF, pode ocorrer a perda dos dados por um dos seguintes motivos:

- interrupção da alimentação através da bateria (acidental ou intencional via *jumper* de perda de RAM)
- por problemas físicos do componente
- por outros problemas quaisquer

O *software* básico, ao ser ligado, verifica a condição da memória RAM. Caso não esteja íntegra, é iniciado o processo de recuperação da mesma através dos dados armazenados na MFD. Este processo pode ser lento, dependendo do movimento registrado na MFD do ECF, pois todos os contadores e totalizadores fiscais serão recalculados baseados nestes dados.

As informações serão recuperadas e o ECF retomará sua condição de funcionamento a partir do ponto onde estava antes de ter havido a perda da memória RAM. Porém, algumas [configurações do ECF](#), armazenadas em registradores, devem ser reconfiguradas, pois após a recuperação da memória RAM esses dados retomam os seus valores de inicialização.

1.2.14 Relógio

O *software* básico possui o controle sobre o relógio do ECF. Há condições específicas para atualização do relógio, a saber:

- o relógio somente pode ser programado em MIT
- não é possível retroceder o relógio com relação ao horário do último documento emitido pelo ECF
- o horário de verão pode ser atualizado sem estar em MIT, porém o dia deve estar fechado, além disto, no caso de retroagir o relógio (saída de horário de verão), o software básico só permite sair do horário de verão caso o horário de término do último documento emitido seja no mínimo inferior a 1 (uma) hora em relação ao horário atual do ECF.
- quando o ECF estiver em horário de verão, a hora será impressa com o sufixo 'V'

1.2.15 Diagnóstico de Erros

Para atender a legislação fiscal em vigor, o *software* básico realiza, ao decorrer de sua execução, uma série de consistências internas. A validade das variáveis fiscais armazenadas em sua memória RAM, a

integridade da MF e da MFD, relógio e os comandos enviados pelo *software* aplicativo são sempre verificados. Qualquer falha em uma dessas atividades básicas é detectado pelo *software* básico do ECF. Os efeitos das detecções de problemas podem ser desde a impressão do problema no mecanismo impressor, um código de erro retornado ao aplicativo ou até mesmo o bloqueio completo de suas atividades.

No caso de bloqueio, é necessária a realização de uma intervenção técnica (MIT) para que ocorra o desbloqueio da máquina. No tópico [Retornos do Protocolo FiscNet](#) são relacionados os possíveis retornos de erro gerados pelo software básico com seu nome e descrição.

1.2.16 Falta de Energia

A impressora fiscal armazena todas as informações de contexto quando ocorre a queda de energia. As atividades interrompidas, quando da falta de energia, são restabelecidas automaticamente pelo software básico da impressora fiscal. Isto garante que qualquer comando recebido pela impressora fiscal será executado mesmo diante a falta de energia.

A falta de energia pode ocasionar, em determinadas circunstâncias, a reimpressão das últimas linhas impressas ou até mesmo o reinício da impressão do último documento emitido não completado.

O *software* básico informa a ocorrência de falta de energia, imprimindo uma mensagem que indica também a data e a hora. Somente não será impressa a mensagem quando não houver documento aberto no momento da falta de energia.

1.3 Protocolo de Comunicação

A partir da série Logger II a ZPM passa a utilizar o protocolo **FiscNET**. Todos os futuros equipamentos fiscais da ZPM serão compatíveis com este protocolo, embora alguns possam não possuir determinada característica específica ou implementar todos os comandos, nenhum equipamento agirá de maneira inesperada ao receber um comando não implementado. Isto assegura que o esforço de adequação dos aplicativos comerciais aos equipamentos futuros serão recompensados com uma estabilidade da interface de comunicação.

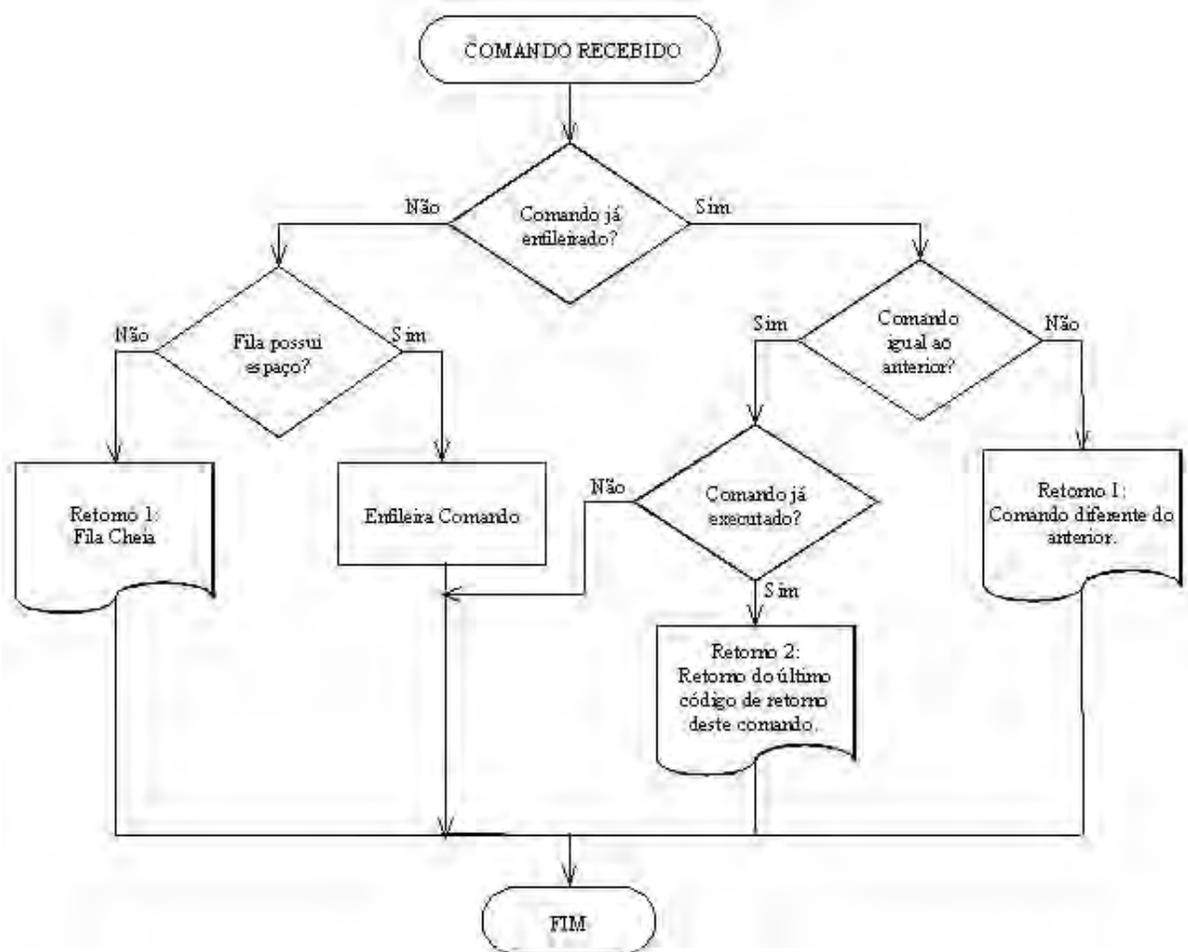
A flexibilização no protocolo FiscNET é obtida através do uso de comandos, parâmetros e códigos de retorno nomeados abrindo possibilidade para uma série de parâmetros e comandos opcionais.

1.3.1 Fluxo de Comandos

O protocolo FiscNET possui comandos de controle fiscal comandados sempre por um **mestre** (geralmente o computador) e obedecidos por um **escravo** (equipamento fiscal).

Os comandos são enfileirados para execução no equipamento fiscal e seu tempo de execução dependerá de diversos fatores como velocidade de impressão do mecanismo em uso, tempo de processamento do comando por parte do processador do equipamento fiscal e disponibilidade de recursos para sua execução como papel, por exemplo.

O conceito de fila de execução em equipamentos fiscais obriga à execução síncrona dos comandos, isto é, um comando precisa necessariamente ser concluído antes que um novo comando possa ser processado. Isto se faz necessário para equipamentos fiscais onde a seqüência dos comandos determina resultados diferentes. Observe no fluxograma a seguir o comportamento do equipamento fiscal frente ao recebimento de um novo comando.



A reação de um equipamento a um comando que não esteja implementado não causará uma reação inesperada do equipamento podendo, no máximo, retornar um código de erro de comando inexistente.

A configuração de comunicação para o protocolo FiscNET é a seguinte:

- Baudrate 115200 bps
- 8 data bits
- paridade par
- 1 stop bit.

1.3.1.1 Estrutura dos Comandos

A estrutura dos pacotes de dados do protocolo FiscNET é uma sequência de caracteres ASCII dividida em seções, separadas pelo caracter ponto e vírgula (";") onde cada seção possui a seguinte finalidade:

1ª seção	Número do identificação do comando. É um número gerado pelo mestre da rede entre ZERO e 255 e deve ser diferente a cada novo comando enviado ao ECF (pelo menos não deve ser repetido num intervalo de 50 comandos). Como sugestão pode ser implementado como um número de auto incremento dos comandos. A finalidade desta número é identificar a que comando os retornos enviados pelo ECF se referem. Opcionalmente este identificador de comando pode ser preenchido em ZERO, caso em que não serão aceitos comandos enfileirados para execução.
2ª seção	Quando enviado pelo mestre: Nome do comando. Mnemônico do comando conforme o tópico Comandos do Protocolo FiscNET implementados em cada modelo e versão. Quando enviado pelo escravo: Código de retorno. Mnemônico do código de retorno conforme o tópico Retorno dos Registradores do protocolo FiscNET.
3ª seção	Quando enviado pelo mestre: Parâmetros do comando. Lista de parâmetros nomeados do comando conforme tabela de comandos do protocolo FiscNET. Quando enviado pelo escravo: Parâmetros de retorno. Lista de parâmetros nomeados retornados pelo comando executado satisfatoriamente ou lista de informações pertinentes ao código de erro retornado.
4ª seção	Opcional: Tamanho do comando em bytes. Incluindo todos os espaços e caracteres especiais até o início desta seção. Se o comando recebido contiver esta seção o equipamento comandado também retornará esta seção na resposta. Esta seção serve como um verificador para o pacote.

O número de identificação do comando é a primeira seção do comando e seu uso serve para associar os diversos comandos recebidos pelo equipamento fiscal com seus respectivos códigos de retorno.

NOTA

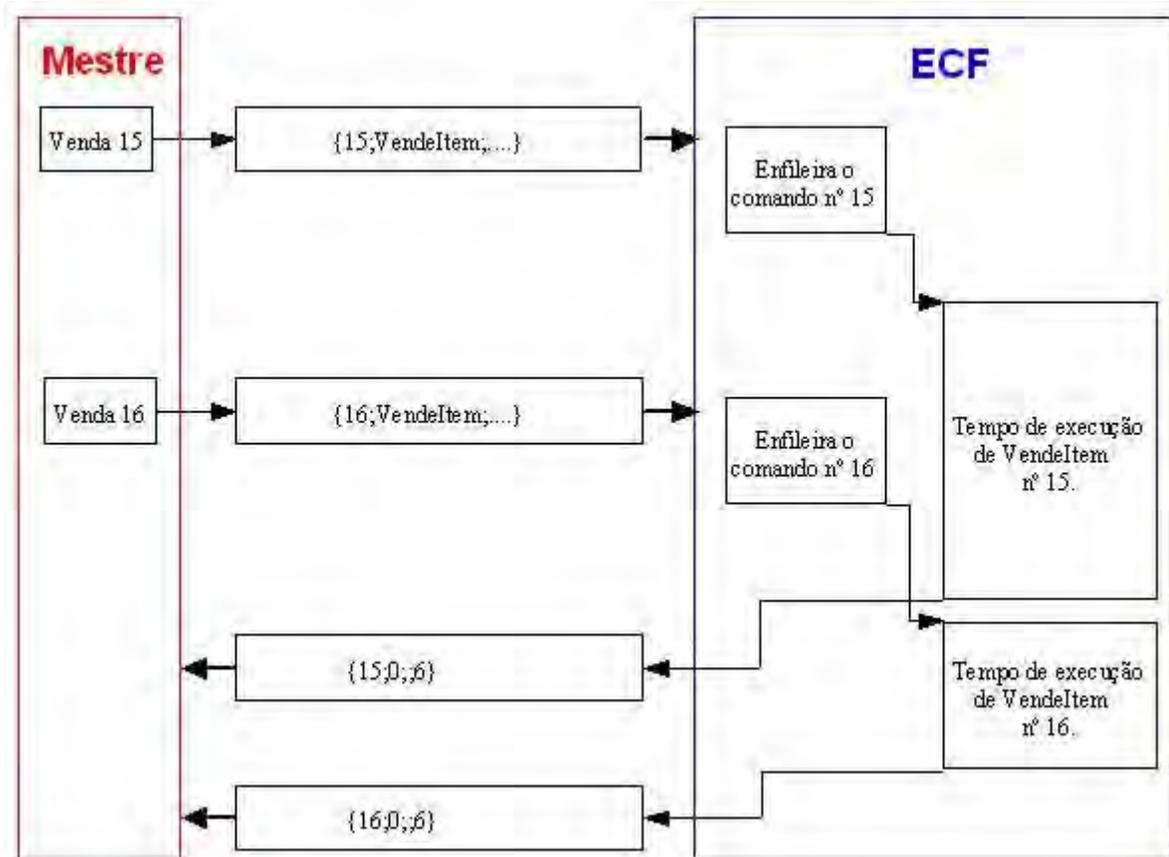
Comandos identificados com o número zero marcarão a fila de execução dos comandos como cheia, não permitindo novos comandos na fila até a liberação do respectivo comando. Isto é necessário pois se vários comandos com número zero forem recebidos torna-se impossível determinar a que comando determinado retorno está associado.

1.3.1.2 Numeração dos Comandos

A figura a seguir ilustra uma seqüência de dois comandos onde o segundo comando passa a ter sua execução postergada até a conclusão do primeiro.

No exemplo o comando de venda nº 16 é enviado pelo mestre ao ECF antes da conclusão do processamento do comando de venda nº 15.

A numeração de comandos também resolve os problemas associados à dupla execução de um comando causada por problemas na recepção do retorno do comando por parte do mestre. Sem uma identificação inequívoca do comando não é possível ao ECF saber se o novo comando recebido é uma retransmissão do primeiro comando ou se é realmente um novo comando.



A figura anterior ilustra esta situação, onde o equipamento mestre, por um problema que poderia ser por exemplo um mau contato no cabo de comunicação não recebeu a resposta ao comando de venda nº 47.

Como o ECF recebeu duas vezes o comando VendeItem nº 47 de maneira exatamente iguais, seguindo o fluxo de execução dos comandos, o último retorno deste comando é retransmitido.

Observa-se também que, mesmo se apenas o primeiro ou o segundo retorno não fosse recebido pelo mestre, o resultado não acarretaria nunca em uma dupla execução do comando.

1.3.1.2.1 Identificação do Comando Igual a Zero

Quando o número de identificação do comando é zero, o equipamento fiscal assume que o equipamento mestre é incapaz de lidar com execução assíncrona dos comandos. Neste caso a fila interna de execução passa a ser limitada em 1 (uma) posição, impedindo o enfileiramento de comandos.

NOTA

Comandos com identificação igual à zero devem ser evitados sempre que possível.

A identificação do comando com zero torna o ECF incapaz de lidar com o problema de dupla execução do comando. A única proteção implementada pelo ECF neste caso é limpar seu *buffer* de recepção quando do término da execução do comando. No entanto, se o mesmo comando for recebido após o tempo de execução do comando anterior, o comando recebido é assumido como sendo um novo comando e é executado.

1.3.1.3 Comandos e Parâmetros

Todos os comandos são identificados por uma constante ASCII. Cada comando pode ou não possuir parâmetros.

Uma tabela completa de todos os comandos bem como seus parâmetros de entrada obrigatórios e opcionais e seus parâmetros de retorno pode ser encontrada no tópico

[Comandos do Protocolo FiscNET](#) deste documento.

Há uma seção específica para informação dos parâmetros em um comando. Quando um comando exigir um parâmetro, deve-se seguir a seguinte sintaxe:

<parname>=<pardata>

onde <parname> é o nome do parâmetro (identificado sempre por uma constante) e <pardata> é um dos tipos de dados definidos pelo protocolo FiscNET para utilização em parâmetros (ver tópico

[Tipos de Dados](#))

Exemplo de comando com um parâmetro:

```
{202;AvancaPapel;Avanco=100;}
```

Se houver a necessidade de informar mais de um parâmetro para um determinado comando, deverá ser informada uma lista de parâmetros na seção dos parâmetros. A lista de parâmetros exige um espaço que separe um parâmetro do outro.

Exemplo de comando com uma lista de parâmetros:

```
{204;DefineAliquota;CodAliquotaProgramavel=2 PercentualAliquota=10;}
```

Se nenhum parâmetro for retornado ou exigido pelo comando a seção de parâmetros pode ser deixada vazia, como por exemplo um comando solicitando uma Leitura X.

Exemplo de comando sem parâmetros:

```
{172;EmiteLeituraX;;19}
```

Neste caso a primeira seção contém o valor 172 identificando o comando. A segunda seção contém o comando desejado EmiteLeituraX. Como este comando não requer parâmetros a terceira seção está vazia e a quarta está preenchida com o tamanho do comando desde o início da 1ª até o início da 4ª seção.

O tamanho do comando é um valor Inteiro sem sinal que indica o tamanho em bytes do comando, desde primeiro byte da primeira seção (excluído o caracter abre chaves "{" até o caracter ponto e vírgula ";" inclusive, que identifica o início da 4ª seção. O tamanho é fundamental como mecanismo de verificação da integridade do comando, mas pode não ser utilizado se seu valor não for especificado.

Caso o tamanho não seja especificado no comando enviado ao escravo, a resposta enviada pelo escravo também não conterà nada nesta seção.

Os códigos de erro possíveis estão listados no tópico [Retornos do Protocolo FiscNet](#) deste documento.

1.3.1.4 Respostas

Cada comando do protocolo FiscNET recebido pelo escravo tem apenas uma resposta. Esta resposta é obrigatoriamente enviada após o processamento do comando por parte do escravo.

O formato do retorno possui as mesmas 4 (quatro) seções do comando, sendo que:

1	A seção 1 define o identificador do comando ao qual a resposta refere-se;
2	A seção 2 representa o código de erro retornado (conforme tópico Retornos do Protocolo FiscNet);
3	A seção 3 representa os retornos do escravo, quando não ocorrer erro (erro diferente de zero), podendo ser vazio; caso a seção 2 represente um erro, serão retornados obrigatoriamente os seguintes parâmetros: <ul style="list-style-type: none"> · "NomeErro" (string que contém o nome do erro) e · "Circunstância" (string, com o texto explicativo sobre a circunstância do erro).
4	A seção 4 informa o tamanho da resposta, desde que o comando tenha sido enviado também com este campo. Caso o comando não foi enviado com este campo, na resposta este também será vazio.

Em determinadas circunstâncias em que o equipamento escravo possa ser comandado diretamente, sem a necessidade de um equipamento mestre enviar comandos, as informações de retorno podem apresentar-se através da impressão do retorno.

1.3.1.5 Comandos

Os comandos programados no protocolo FiscNET estão listados no [Comandos do Protocolo FiscNet](#), com seu nome de identificação (valor que será enviado na seção 2 do comando), seus parâmetros opcionais e obrigatórios (seção 3 do comando) e um descritivo geral sobre sua funcionalidade. Para saber mais sobre as seções do comando ver tópico [Estrutura dos Comandos](#) neste documento.

1.3.2 Protocolo Logger I

Com o intuito de manter a compatibilidade com a geração de equipamentos Logger anterior, aqui referenciada como Logger I, o *software* básico implementa internamente um **Tradutor de Protocolo**.

1.3.2.1 Tradutor de Protocolo Logger I

O principal objetivo do Tradutor de Protocolos é manter ao máximo a funcionalidade dos aplicativos escritos para *Logger I*, preferencialmente sem alterações, ou caso haja, com a mínima alteração possível.

Neste sentido, suas principais funções são:

- aceitar todos os comandos do protocolo Logger I,
- convertê-lo para um ou mais comandos do protocolo FiscNET que possam ter a mesma funcionalidade ou a mais próxima possível,
- executar os comandos FiscNET,
- sincronizar e obter o(s) retorno(s) do protocolo FiscNET,
- convertê-lo(s) para o erro mais condizente na tabela de erros da Logger I.

O processo de tradução de protocolo funciona com poucas restrições, sendo essas referente a diferenças de protocolo ou até mesmo em função das alterações na legislação fiscal. Ver detalhes sobre as restrições no [Restrições do Tradutor de Protocolo](#).

1.3.2.2 Programação do Protocolo

O ECF possui duas portas seriais, sendo uma de uso exclusivo do fisco e outra para uso do *software* aplicativo.

A porta serial de uso exclusivo do fisco opera somente com o protocolo de comunicação FiscNET. A porta serial do *software* aplicativo pode ser programada com um dos 2 (dois) protocolos: FiscNET (*default*) ou Logger I.

Não é possível usar ambos os protocolos simultaneamente na mesma porta serial, pois suas configurações de porta são diferentes:

FiscNET	Logger I
115200 bps	9600 bps
paridade par	sem paridade
8 data bits	8 data bits
1 stop bit	1 stop bit

A programação do protocolo de comunicação é feita através de um registrador chamado **ProtocoloSerial** (ver tópico [Registradores do Protocolo FiscNet](#)). Ao escrever o valor **Logger I** neste registrador, basta desligar e religar o ECF para que ele aceite apenas comandos referente a Logger I. Para voltar ao protocolo de origem, deve-se escrever o valor **FiscNET** neste mesmo registrador, desligar e religar o ECF. Porém, como não há como fazê-lo através do protocolo Logger I, deve-se usar o canal do fisco para que seja retornado a configuração de protocolo de origem.

1.3.2.3 Uso de Recursos FiscNET através do Protocolo Logger I

Apesar de facilitar substancialmente a adaptação de um *software* aplicativo já implementado para o protocolo Logger I, a limitação dos comandos deste último restringe o acesso aos novos comandos implementados no protocolo FiscNET.

Para solucionar esta limitação, o tradutor implementa um comando extra com a padronização da Logger I com o intuito de permitir o acesso aos comandos nativos FiscNET. Isto permite que o *software* aplicativo não tenha necessidade de ser reescrito totalmente para o protocolo FiscNET para acessar os recursos nativos, tais como acréscimo em item, por exemplo.

O formato do comando *Logger I*, implementado somente pelo tradutor de comandos, segue:

Formato	Comando					
ASCII	ESC	f	6	0	Tamanho 'Comando '	Comando
Hexa	1B	66	36	30	4 bytes numéricos	Comando a ser enviado
Decimal	27	102	54	48	4 bytes numéricos	Comando a ser enviado

Onde:

- **Tamanho 'Comando'** representa o total de bytes do campo seguinte (campo Comando)
- **Comando** deve conter apenas os dados da seção 2 e da seção 3 do protocolo FiscNET.

NOTA

Usuários do *device driver* para DOS e *dll* Windows da Logger I devem solicitar as novas versões que implementam este comando para usá-lo com o tradutor de protocolo.

O parâmetro 'Comando' deve ser montado apenas com as seções 2 e 3 do comando do protocolo FiscNET. Internamente, o *device driver* e a *dll* realizam a montagem do comando da seguinte forma:

Comando FiscNET		
{seção1;	seção 2;seção 3	;seção4}
Montado automaticamente <i>dll</i> e <i>device driver</i>	Montado pelo aplicativo parâmetro <i>Comando</i>	Montado automaticamente <i>dll</i> e <i>device driver</i>

NOTA

Quando um comando FiscNET não exigir parâmetros, a seção 3 será vazia. Porém, o caracter ; , responsável por separar as seções, deve ser montado pelo aplicativo.

Exemplo: Para enviar um comando FiscNET de acréscimo de R\$ 0,55 para o item 1, via tradutor de comandos Logger I – comando 60, seria necessário montar o seguinte comando:

				Tam Cmd	Comando
ESC	f	6	0	0058	AcresceItemFiscal;Cancelar=f NumItem=1 ValorAcrescimo=0,55

Portanto, o parâmetro 'Comando' a ser montado pelo aplicativo deve conter apenas as seções 2 e 3 do protocolo FiscNET. O controle do número e tamanho do pacote do comando – seções 1 e 4 do protocolo FiscNET, são gerenciados internamente pelo tradutor de comandos Logger I. Através deste, comandos FiscNET podem ser executados através do protocolo Logger I. É importante ressaltar que o retorno de um comando Logger I possui apenas 1 (um) byte para a maioria dos comandos (exceto Leitura de Registrador). Como o protocolo FiscNET pode enviar retornos maiores que a Logger I, a recepção do retorno deste comando concluirá sempre por *timeout*. Outro detalhe a ser considerado é o fato de que o retorno FiscNET deverá ser processado de acordo com sua estrutura (ver tópico [Respostas](#)) e não como a estrutura de um retorno da Logger I.

1.3.3 Biblioteca

É disponibilizada uma biblioteca de funções do tipo *DLL* para Windows® 98 ou superior, que implementa funções que auxiliam no uso do protocolo FiscNET. Para maiores informações, consulte o documento [Manual da DLL-G2](#).

1.3.4 Considerações Finais

O novo protocolo de comunicação traz novas funcionalidades que incrementam muito seus recursos quanto ao tratamento de erros e retransmissões de comandos. Estas características, apesar de inexistentes na Logger I, trazem uma maior necessidade de controle e tratamento de situações por parte do *software* aplicativo, porém trazendo-lhe benefícios anteriormente inexistentes.

Para os aplicativos que já comunicam com o ECF Logger I, aconselha-se usar o [tradutor de protocolo](#) da Logger II, porém antes testando o comportamento entre o aplicativo e o tradutor de protocolo para depois liberar a versão final para o cliente. Talvez seja necessário a realização de alguns ajustes no aplicativo (ver tópico [Restrições do Tradutor de Protocolo](#)).

Os aplicativos que não possuem ainda qualquer implementação para equipamentos da linha Logger, aconselha-se iniciar diretamente pelo uso do protocolo FiscNET, pois o aplicativo ganhará em velocidade de comunicação e recursos de protocolo.

É importante salientar que a tendência dos produtos fiscais da linha Logger é o uso do protocolo FiscNET. Neste sentido, a atualização será recompensada.

1.4 O Firmware da Impressora Fiscal

Este tópico visa esclarecer o funcionamento interno da impressora fiscal. O bom entendimento das características internas de funcionamento da impressora fiscal possibilitará um melhor desempenho entre a impressora fiscal e o *software* aplicativo.

O *firmware* da impressora fiscal atende integralmente as exigências da Legislação Fiscal dos Convênios 85/01 e 133/01.

1.5 Tipos de Dados

Os tipos de dados aceitos pelo protocolo de comunicação FiscNET são:

Tipo	Descrição
Valores Inteiros	Números sem ponto decimal, positivos ou negativos, podendo ser classificados da seguinte forma: <byte> Valores entre 0 e 127 para valores positivos e -1 e -128 para valores negativos <ubyte> Valores entre 0 e 255 sem sinal. <int> Inteiro de 16 bits com sinal variando de 0 a 32.767 para valores positivos e de -1 a -32.768 para valores negativos. <uint> Somente valores positivos entre 0 e 65.535. <long> Inteiro de 32 bits variando entre 0 e 2.147.483.647 para valores positivos e de -1 a -2.147.483.648 para valores negativos. <ulong> Inteiro de 32 bits sem sinal variando de 0 a 4.294.967.295.
Valores Moeda	Indicados por <money> São valores positivos ou negativos que utilizam-se do caracter ponto (",") como caracter de separação decimal. A representação máxima deste tipo de dado é de 18 dígitos, incluindo 4 casas decimais.
Valores Indicadores	Indicados por <bool> devem ser informados pelas constantes "Y"/ "N", representando <i>Yes</i> e <i>No</i> respectivamente, além de "t"/"f" representando <i>Verdadeiro (true)</i> ou <i>Falso (false)</i> .
Valores Texto	Indicados por <string> devem ser delimitados por aspas. Veja nota a seguir para constantes tipo texto.
Valores Data	Campos que referenciam datas devem ser delimitados pelo caracter ("#") - exceto quando usar o comando EmitLeituraMF. O formato da data é <i>dd/mm/aaaa</i> .
Valores Hora	Os campos que identificam hora devem ser informados através do formato <i>hh:mm:ss</i> .

NOTA para campos tipo TEXTO

Caracteres acentuados podem ser usados. A tabela de caracteres usada pelo ECF é a PC-850. Se o caracter aspas for necessário, deve ser precedido de um caracter '\' (barra inversa) que será desconsiderado, restando apenas o caracter aspas. Se o caracter barra inversa (\) for necessário devem ser informados dois caracteres barras inversa (\\), mas somente um deles será considerado. Se for necessária a utilização de caracteres abaixo de 32 ou acima de 127 eles devem ser representados pelo seu código em hexadecimal de dois dígitos, precedidos do caracter barra inversa e 'x' (\x). Por exemplo para representar o caracter ESC (escape) da tabela ASCII deve-se usar um texto na forma "\x1B".

TÓPICOS RELACIONADOS:

- [Limites do ECF](#)
- [Situações tributárias](#)
- [Meios de pagamento](#)
- [Totalizadores não fiscais](#)
- [Departamentos](#)
- [Cupom Fiscal](#)
- [Comprovantes não fiscais](#)
- [Relatório Gerencial](#)
- [Relatórios Fiscais](#)
- [Configurações](#)
- [Leitura de registradores](#)
- [Comandos e parâmetros](#)
- [Gerenciais](#)
- [Departamentos](#)
- [Comprovante de Crédito ou Débito](#)

1.6 Tipo de parâmetro

Ao executar o comando DLLG2_AdicionaParam deve ser indicado qual o tipo de parâmetro, de acordo com o tipo de dado do parâmetro:

Tipo de dado	Tipo do parâmetro
byte	
ubyte	4
int	
money	6
bool	0
string	7
data	2
hora	3
long	5
uint	9
ulong	10

1.7 Registradores Protocolo FiscNET

Os registradores dos ECFs Logger II estão relacionados a seguir em ordem alfabética, juntamente com a sua *Descrição*, *Tipo de Acesso* e *Tipo*.

O *Tipo de Acesso* de um registrador pode ser definido como R (Read ou leitura) e/ou W (Write ou escrita).

Os *Tipo* define a faixa de valores mínima e máxima que um registrador pode assumir. Alguns registradores possuem uma restrição além de seu *Tipo*. Os Tipos do ECF estão relacionados no tópico tipos de dados. Alguns tipos requerem o tamanho máximo, tal como o tipo *String*.

Alguns registradores representam uma lista ou vetor de elementos. Nestes casos, são apresentados os seus índices mínimos e máximos.

TÓPICOS RELACIONADOS:

- [Limites do ECF](#)
- [Totalizadores não fiscais](#)
- [Dados Clichê](#)
- [Situações Tributárias](#)
- [Meios de Pagamento](#)
- [Gerenciais](#)
- [Departamentos](#)
- [Cupom Fiscal](#)
- [Comprovantes não fiscais](#)
- [Relatório Gerencial](#)
- [Relatórios Fiscais](#)
- [Configurações](#)
- [Leitura de registradores](#)
- [Dados de Identificação do Proprietário](#)
- [Comprovante de Crédito ou Débito](#)

1.7.1 Aliquota00ICMSReducao

TipoAcesso	R
Descrição	Indica se a alíquota 00 é do tipo ICMS na redução "n".
Tipo	bool

IndiceMinimo 1
 IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota00ICMSReducao[1]', 7);
DLLG2_ExecutaComando(hPorta, 'LeIndicador');
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.2 Aliquota00Reducao

TipoAcesso R
 Descrição Retorna o percentual da alíquota 00 na redução "n".
 Tipo money
 IndiceMinimo 1
 IndiceMaximo NUM_MAX_REDUCOES-1

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota00Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.3 Aliquota01ICMSReducao

TipoAcesso R
 Descrição Indica se a alíquota 01 é do tipo ICMS na redução "n".
 Tipo bool
 IndiceMinimo 1
 IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota01ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.4 Aliquota01Reducao

TipoAcesso R
 Descrição Retorna o percentual da alíquota 01 na redução "n".
 Tipo money
 IndiceMinimo 1
 IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota01Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.5 Aliquota02ICMSReducao

TipoAcesso R
 Descrição Indica se a alíquota 02 é do tipo ICMS na redução "n".
 Tipo bool

IndiceMinimo 1
IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota01ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.6 Aliquota02Reducao

TipoAcesso R
Descrição Retorna o percentual da alíquota 02 na redução "n".
Tipo money
IndiceMinimo 1
IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota01Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.7 Aliquota03ICMSReducao

TipoAcesso R
Descrição Indica se a alíquota 03 é do tipo ICMS na redução "n".
Tipo bool
IndiceMinimo 1
IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota03ICMSReducao[1]', 0);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.8 Aliquota04Reducao

TipoAcesso R
Descrição Retorna o percentual da alíquota 04 na redução "n".
Tipo money
IndiceMinimo 1
IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota04Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.9 Aliquota05ICMSReducao

TipoAcesso R
Descrição Indica se a alíquota 05 é do tipo ICMS na redução "n".

Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota05ICMSReducao[1]', 0);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.10 Aliquota05Reducao

TipoAcesso	R
Descrição	Retorna o percentual da alíquota 05 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota05Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.11 Aliquota06ICMSReducao

TipoAcesso	R
Descrição	Indica se a alíquota 06 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota06ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.12 Aliquota06Reducao

TipoAcesso	R
Descrição	Retorna o percentual da alíquota 06 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota06Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.13 Aliquota07ICMSReducao

TipoAcesso	R
Descrição	Indica se a alíquota 07 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota07ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.14 Aliquota07Reducao

TipoAcesso	R
Descricao	Retorna o percentual da alíquota 07 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota07Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.15 Aliquota08ICMSReducao

TipoAcesso	R
Descricao	Indica se a alíquota 08 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota08ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.16 Aliquota08Reducao

TipoAcesso	R
Descricao	Retorna o percentual da alíquota 08 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota08Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
```

```
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.17 Aliquota09ICMSReducao

TipoAcesso	R
Descricao	Indica se a alíquota 09 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota09ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.18 Aliquota09Reducao

TipoAcesso	R
Descricao	Retorna o percentual da alíquota 09 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota09Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.19 Aliquota10ICMSReducao

TipoAcesso	R
Descricao	Indica se a alíquota 10 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota10ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.20 Aliquota10Reducao

TipoAcesso	R
Descricao	Retorna o percentual da alíquota 10 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota10Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
```

```
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.21 Aliquota11ICMSReducao

TipoAcesso	R
Descricao	Indica se a alíquota 11 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota11ICMSReducao[1]', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.22 Aliquota11Reducao

TipoAcesso	R
Descricao	Retorna o percentual da alíquota 11 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota11Reducao[1]', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.23 Aliquota12ICMSReducao

TipoAcesso	R
Descricao	Indica se a alíquota 12 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota12ICMSReducao[1]', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.24 Aliquota12Reducao

TipoAcesso	R
Descricao	Retorna o percentual da alíquota 12 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota10Reducao[1]', 7);
```

```
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.25 Aliquota13ICMSReducao

TipoAcesso	R
Descricao	Indica se a alíquota 13 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota13ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.26 Aliquota13Reducao

TipoAcesso	R
Descricao	Retorna o percentual da alíquota 13 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota13Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.27 Aliquota14ICMSReducao

TipoAcesso	R
Descricao	Indica se a alíquota 14 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota14ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.28 Aliquota14Reducao

TipoAcesso	R
Descricao	Retorna o percentual da alíquota 14 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota14Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.29 Aliquota15ICMSReducao

TipoAcesso	R
Descricao	Indica se a alíquota 15 é do tipo ICMS na redução "n".
Tipo	bool
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'Aliquota15ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.30 Aliquota15Reducao

TipoAcesso	R
Descricao	Retorna o percentual da alíquota 15 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'Aliquota15ICMSReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.31 AliquotaDisponivel

TipoAcesso	R
Descricao	Retorna o indice da proxima aliquota disponivel para carga
Tipo	int

Este registrador limita-se a leitura. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'AliquotaDisponivel', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.32 Arredondamento

TipoAcesso	RW
Descricao	Lê ou define o método de arredondamento para as operações aritméticas. Os valores válidos são 0 para arredondamento segundo ABNT, 1 para truncamento e 2 para arredondamento para cima. O arredondamento só pode ser modificado em Modo de Intervenção Técnica.
Tipo	byte

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'Arredondamento', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.33 BaudRate

TipoAcesso	RW
Descricao	Configura e indica o baudrate para o protocolo FiscNET no canal do usuário. Os valores válidos são 4800, 9600, 19200, 38400, 57600 e 115200. O valor default é 115200. Após alterado o novo baudrate só tem efeito a partir do próximo reset do ECF.
Tipo	ulong

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'BaudRate', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.34 CancelamentosICMSReducao

TipoAcesso	R
Descricao	Retorna o total de cancelamentos em ICMS para a redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'CancelamentoICMSReducao', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.35 CancelamentosISSQNReducao

TipoAcesso	R
Descricao	Retorna o total de cancelamentos de ISSQN para a redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'CancelamentoISSQNReducao', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.36 CCF

TipoAcesso	R
Descricao	Contador de Cupons Fiscais
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CCF', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.37 CDC

TipoAcesso	R
Descricao	Contador de Cupons Crédito/Débito
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CDC', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.38 CER

TipoAcesso	R
Descricao	Seqüência de 'NUM_MAX_GERENCIAIS' Contadores Específicos de Relatórios Gerenciais diários emitidos
Tipo	uint
IndiceMaximo	NUM_MAX_GERENCIAIS-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CER', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.39 CFC

TipoAcesso	R
Descricao	Contador de Cupons Fiscais Cancelados
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CFC', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.40 CFD

TipoAcesso	R
Descricao	Contador de emissao de fita detalhe
Tipo	uint

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CFD', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.41 Cliche

TipoAcesso	R
Descricao	Retorna o cliche do usuario atual
Tipo	string
TamMaximo	255

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'Cliche', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.42 CMC7Campo1

TipoAcesso	R
Descricao	Retorna a ultima leitura do campo 1 do CMC7.
Tipo	string
TamMaximo	50

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'CMC7Campo1', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.43 CMC7Campo2

TipoAcesso	R
Descricao	Retorna a ultima leitura do campo 2 do CMC7
Tipo	string
TamMaximo	50

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'CMC7Campo2', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.44 CMC7Campo3

TipoAcesso	R
Descricao	Retorna a ultima leitura do campo 3 do CMC7
Tipo	string
TamMaximo	50

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'CMC7Campo3', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.45 CMC7Documento

TipoAcesso	R
Descricao	Lê o texto CMC7 de um documento.
Tipo	string
TamMaximo	50

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'CMC7Documento', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.46 CMC7Memoria

TipoAcesso	R
Descricao	Texto CMC7 lido do último documento do qual foi lido o registrador CMC7Documento
Tipo	string
TamMaximo	50

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'CMC7Memoria', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.47 CNPJ

TipoAcesso	R
Descricao	Inscrição no cadastro nacional de pessoas jurídicas.
Tipo	string
TamMaximo	20

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'CNPJ', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.48 CON

TipoAcesso	R
Descricao	Seqüência de 'NUM_MAX_NAO_FISCAIS' Contador de Cupom Não Fiscal diário
Tipo	uint
IndiceMaximo	NUM_MAX_NAO_FISCAIS-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CON[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.49 ContadorDocUltimoItemVendido

TipoAcesso	R
Descricao	Índice do último item vendido no cupom.
Tipo	int

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'ContadorDocUltimoItemVendido', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.50 ContadorProprietarios

TipoAcesso	R
Descricao	Contador de Proprietários
Tipo	byte

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'ContadorProprietarios', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.51 ContadorReinicioReducao

TipoAcesso	R
Descricao	Retorna o contador de reinico de operacao para a redução "n".
Tipo	byte
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUcoes-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'ContadorReinicioReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.52 ContadorVersoes

TipoAcesso	R
Descricao	Contador de versões de software básico instaladas
Tipo	byte

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'ContadorVersoes', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.53 ContrasteBobina

TipoAcesso	RW
Descricao	Registrador utilizado para regulagem da intensidade do contraste em mecanismos de bobina térmica. O valor padrão é ZERO e pode variar entre -100 e 100.

Tipo int

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'ContrasteBobina', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.54 ContrastePagina

TipoAcesso RW

Descricao Registrador utilizado para regulagem da intensidade do contraste em mecanismos de página térmicos. O valor padrão é ZERO e pode variar entre -100 e 100.

Tipo int

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'ContrastePagina', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.55 COO

TipoAcesso R

Descricao COO atual

Tipo ulong

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'COO', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.56 COOInicioDia

TipoAcesso R

Descricao COO do início do dia

Tipo ulong

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'COOInicioDia', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.57 COOReducao

TipoAcesso R

Descricao Retorna o COO da redução "n".

Tipo ulong

IndiceMinimo 1

IndiceMaximo NUM_MAX_REDUcoes-1

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'COOReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.58 CRO

TipoAcesso	R
Descricao	Contador de Reinício Atual
Tipo	byte

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CRO', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.59 CRORestantes

TipoAcesso	R
Descricao	
Tipo	byte

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CRORestantes', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.60 CRZ

TipoAcesso	R
Descricao	Contador de reduções Z.
Tipo	ulong

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CRZ', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.61 CRZRestantes

TipoAcesso	R
Descricao	
Tipo	int

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'CRZRestantes', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.62 DadosUltimaReducaoZ

TipoAcesso	R
Descricao	Retorna os dados da ultima reducao Z.
Tipo	string
TamMaximo	616

Para conhecer o formato do retorno desse registrador ver o tópico [DadosUltimaReducaoZ](#).
Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'DadosUltimaReducao', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.63 Data

TipoAcesso	R
Descricao	Data atual do relógio do ECF
Tipo	data

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeData.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeData', 'Data', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeData'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.64 DataAbertura

TipoAcesso	R
Descricao	Data de abertura do movimento atual
Tipo	data

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeData.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeData', 'DataAbertura', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeData'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.65 DataReducao

TipoAcesso	R
Descricao	Retorna a data da redução "n".
Tipo	data
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeData.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeData', 'DataReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeData'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.66 DataUltimoDoc

TipoAcesso	R
Descricao	Retorna a data do ultimo documento emitido
Tipo	data

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeData.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeData', 'DataUltimoDoc', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeData'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.67 DescontosReducao

TipoAcesso	R
Descricao	Retorna o total de descontos em ICMS para a redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'DescontosReducao', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.68 DiaAberto

TipoAcesso	R
Descricao	Abertura do dia ja iniciada.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'DiaAberto', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.69 DiaFechado

TipoAcesso	R
Descricao	Redução Z já foi realizada. O ECF encontra-se fechado para operações fiscais.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'DiaFechado', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.70 DocumentoAberto

TipoAcesso	R
Descricao	Documento aberto.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'DocumentoAberto', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.71 ECF

TipoAcesso	R
Descricao	Registrador utilizado para ler o número seqüencial deste ECF informado no parâmetro ECF do comando DefineDadosCabecalho.
Tipo	uint

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'ECF', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.72 EnderecoSoftwareBasico

TipoAcesso	RW
Descricao	Indica o endereço do software básico de onde serão retornados os dados lidos através do registrador SoftwareBasico.
Tipo	ulong

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'EnderecoSoftwareBasico', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.73 EspacamentoDocumentos

TipoAcesso	RW
Descricao	Espaço que deve ser automaticamente avançado ao final da impressão de cada documento em décimos de milímetros. Faixa de valores válidos entre ZERO e 1000.
Tipo	uint

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'EspacamentoDocumentos', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.74 EspacamentoLinhas

TipoAcesso	RW
Descricao	Valor entre 0 e 30 que indica o avanço em décimos de milímetro que deve ser acrescentado a cada linha em mecanismos que permitam esta característica. O valor padrão é ZERO.
Tipo	int

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'EspacamentoLinhas', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.75 EstadoFiscal

TipoAcesso	R
Descricao	Código do estado atual do ECF
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'EstadoFiscal', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.76 EstadoGeralECF

TipoAcesso	R
Descricao	Retorna o estado geral do ECF.
Tipo	uint

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'EstadoGeralECF', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.77 GNF

TipoAcesso	R
Descricao	Contador Geral Não Fiscal
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'GNF', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.78 GRG

TipoAcesso	R
Descricao	Contador Geral de Relatório Gerencial
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'GRG', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.79 GT

TipoAcesso	R
Descricao	Totalizador Geral de venda bruta referente ao proprietário em uso
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'GT', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.80 GTInicial

TipoAcesso	R
Descricao	GT no inicio do dia
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'GTInicial', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.81 HabilitaCupomAdicional

TipoAcesso	RW
Descricao	Habilita a impressão automática de Cupom Adicional após a impressão de cada Cupom Fiscal.
Tipo	bool

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'HabilitaCupomAdicional', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.82 Hora

TipoAcesso	R
Descricao	Hora atual do relógio do ECF
Tipo	hora

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeHora.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeHora', 'Hora', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeHora'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.83 HoraReducao

TipoAcesso	R
Descricao	Retorna a hora da redução "n".
Tipo	hora

IndiceMinimo 1
 IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeHora.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeHora', 'HoraReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeHora'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.84 HorárioVerão

TipoAcesso RW
 Descricao Indica se o ECF encontra-se ou não em horário de verão.
 Tipo bool

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'HoraVerão', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.85 HoraUltimoDoc

TipoAcesso R
 Descricao Retorna a hora do ultimo documento emitido.
 Tipo hora

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeHora.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeHora', 'HoraUltimoDoc', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeHora'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.86 IE

TipoAcesso R
 Descricao Inscrição Estadual.
 Tipo string
 TamMaximo 20

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'IE', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.87 IM

TipoAcesso R
 Descricao Inscrição municipal
 Tipo string
 TamMaximo 20

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'IM', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.88 Indicadores

TipoAcesso	R
Descricao	Retorna as situacoes do ECF.
Tipo	ulong

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'Indicadores', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.89 IntervencaoTecnica

TipoAcesso	R
Descricao	Retorna true caso o ECF esteja em MIT, e false caso contrario.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'IntervencaoTecnica', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.90 Loja

TipoAcesso	R
Descricao	Número da Loja deste ECF informada no parâmetro Loja do comando DefineDadosCabecalho.
Tipo	uint

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'Loja', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.91 Marca

TipoAcesso	R
Descricao	Marca do ECF
Tipo	string
TamMaximo	15

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'Marca', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.92 MecanismoBobina

TipoAcesso	R
Descricao	Registrador utilizado para consultar o nome do mecanismo impressor dos documentos fiscais e não fiscais atualmente selecionado.
Tipo	string
TamMaximo	40

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'MecanismoBobina', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.93 MecanismoBobinaPadrao

TipoAcesso	RW
Descricao	Registrador que serve para ler ou alterar o número do mecanismo de bobina padrão a ser utilizado para a impressão dos documentos do ECF, variando entre ZERO e <MecanismosBobina> - 1. Não pode ser modificado se houver documento em emissão.
Tipo	int

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'MecanismoBobinaPadrao', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.94 MecanismoPagina

TipoAcesso	R
Descricao	Registrador utilizado para consultar o nome do mecanismo impressor de cheques e autenticações.
Tipo	string
TamMaximo	40

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'MecanismoPagina', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.95 MecanismosBobina

TipoAcesso	R
Descricao	Retorna o número de mecanismos de bobina simultâneos existentes neste ECF.
Tipo	int

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'MecanismoBobina', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.96 MFDDesconectada

TipoAcesso	R
Descricao	MFD desconectada ou ausente.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'MFDDesconectada', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.97 MFDesconectada

TipoAcesso	R
Descricao	Memória fiscal desconectada ou ausente.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'MFDesconectada', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.98 MFDEsgotada

TipoAcesso	R
Descricao	O(s) recurso(s) de MFD instalado(s) no ECF se encontra(m) esgotado(s).
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'MFDEsgotada', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.99 MFDLivre

TipoAcesso	R
Descricao	Retorna o numero de bytes livres na MFD
Tipo	long

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'MFDLivre', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.100 Modelo

TipoAcesso	R
Descricao	Modelo do ECF
Tipo	string
TamMaximo	20

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'Modelo', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.101 NCN

TipoAcesso	R
Descricao	Contador de Cupons Crédito/Débito pendentes para emissão
Tipo	uint

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'Modelo', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.102 NFC

TipoAcesso	R
Descricao	Contador de Cupons Não Fiscais Cancelados
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'NFC', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.103 NomeEstadoFiscal

TipoAcesso	R
Descricao	Nome do atual estado do ECF em ASCII
Tipo	string
TamMaximo	50

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'NomeEstadoFiscal', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.104 NomeMoedaPlural

TipoAcesso	RW
Descricao	Registrador utilizado para definir ou ler o nome da moeda no plural para a impressão de valores por extenso nos cheques.
Tipo	string
TamMaximo	22

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'NomeMoedaPlural', 7);
```

```
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.105 NomeMoedaSingular

TipoAcesso	RW
Descricao	Registrador utilizado para definir ou ler o nome da moeda no singular para a impressão de valores por extenso nos cheques.
Tipo	string
TamMaximo	19

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'NomeMoedaSingular', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.106 NumeroMeiosPagamentoCupomAtual

TipoAcesso	R
Descricao	
Tipo	byte

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'NumeroMeiosPagamentoCupomAtual', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.107 NumeroSerieECF

TipoAcesso	R
Descricao	Número de série de fabricação do ECF.
Tipo	string

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'NumeroSerieECF', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.108 NumeroSerieMFD

TipoAcesso	R
Descricao	Retorna o numero de serie do ultimo cartucho de MFD instalado
Tipo	string

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'NumeroSerieMFD', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.109 NumLinhasImpressas

TipoAcesso	R
Descricao	Número total de linhas impressas pelo mecanismo desde o início da vida útil do equipamento.
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'NumLinhasImpressas', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.110 OperacoesNaoFiscaisReducao

TipoAcesso	R
Descricao	Retorna o total de operações não fiscais para a redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'OperacoesNaoFiscaisReducao', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.111 PercentualMFDUtilizada

TipoAcesso	R
Descricao	Retorna o percentual utilizado na MFD
Tipo	uint

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'PercentualMFDUtilizada', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.112 ProtocoloSerial

TipoAcesso	RW
Descricao	Indica qual o protocolo que será atendido pelo canal de comunicação serial do usuário no próximo religamento do ECF. O padrão é "FiscNET", podendo assumir opcionalmente "Logger I", caso em que será emulado o protocolo dos equipamentos da série Logger ZPM.
Tipo	string
TamMaximo	30

Este registrador pode ser lido e escrito. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'ProtocoloSerial', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.113 RAMInvalida

TipoAcesso	R
Descricao	RAM com dados invalidos
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'RAMInvalida', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.114 RelogioInconsistente

TipoAcesso	R
Descricao	Relogio com valores inválidos.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'RelogioInconsistente', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.115 SemPapel

TipoAcesso	R
Descricao	ECF sem bobina de papel na estação de cupom fiscal.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'SemPapel', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.116 SensorAutenticacao

TipoAcesso	R
Descricao	Retorna o estado do sensor de autenticacao
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'SensorAutenticacao', 7);  
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));  
strBuffer := StringOfChar(' ',100);  
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.117 SensorCabecote

TipoAcesso	R
Descricao	Retorna o estado do sensor de cabecote levantado do mecanismo termico.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'SensorCabecote', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.118 SensorCheque

TipoAcesso	R
Descricao	Retorna o estado do sensor de cheque.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'SensorCheque', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.119 SensorGaveta

TipoAcesso	R
Descricao	Retorna o estado do sensor da gaveta de valores.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'SensorGaveta', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.120 SensorPoucoPapel

TipoAcesso	R
Descricao	Retorna o estado do sensor de pouco papel.
Tipo	bool

Este registrador pode ser apenas lido. Para ler este registrador deve ser utilizado o comando LeIndicador.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeIndicador', 'SensorPoucoPapel', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeIndicador'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.121 SimboloMoeda

TipoAcesso	RW
Descricao	Registrador que define ou retorna o símbolo a ser utilizado como valor monetário. Em caso de ser escrito, seu valor só será aceito imediatamente após uma Redução Z.
Tipo	string
TamMaximo	4

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'SimboloMoeda', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.122 SoftwareBasico

TipoAcesso	R
Descricao	Retorna os 1024 bytes em formato hexadecimal do software básico a partir do endereço indicado pelo registrador EnderecoSoftwareBasico.
Tipo	string
TamMaximo	2048

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'SoftwareBasico', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.123 TamanhoSoftwareBasico

TipoAcesso	R
Descricao	Retorna o tamanho em bytes do software básico.
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'TamanhoSoftwareBasico', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.124 TempoAcionamentoPadrao

TipoAcesso	RW
Descricao	Tempo padrao de acionamento da gaveta em milissegundos. Valor padrão é 100 ms.
Tipo	uint

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'TempoAcionamentoPadrao', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.125 TempoEmitindoDocFiscal

TipoAcesso	R
Descricao	Tempo contabilizado pelo equipamento em que ficou emitindo documentos fiscais no dia.
Tipo	ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'TempoEmitindoDocFiscal', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.126 TempoOperacional

TipoAcesso	R
Descricao	Tempo que o equipamento ficou ligado durante o dia

Tipo ulong

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'TempoOperacional', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.127 TextoLivre

TipoAcesso RW
 Descricao Registrador de livre uso pelo aplicativo. Pode ser livremente lido e escrito pelo aplicativo e não é interpretado pelo ECF.
 Tipo string
 TamMaximo 250

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'TextoLivre', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.128 TipoDocEmEmissao

TipoAcesso R
 Descricao Tipo do documento em emissão
 Tipo int

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'TipoDocEmEmissao', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.129 TipoECF

TipoAcesso R
 Descricao Tipo do ECF (ECF-IF, ECF-PDV, ECF-MR)
 Tipo string
 TamMaximo 7

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'TipoECF', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.130 TipoUltimoDocEmitido

TipoAcesso R
 Descricao Tipo do último documento emitido
 Tipo int

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'TipoUltimoDocEmitido', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.131 TotalAliquota00Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 00 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota00Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.132 TotalAliquota01Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 01 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota01Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.133 TotalAliquota02Reducao

TipoAcesso	R
Descricao	Retorna o total da aliquota 02 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota02Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.134 TotalAliquota03Reducao

TipoAcesso	R
Descricao	Retorna o total da aliquota 03 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota03Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.135 TotalAliquota04Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 04 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota04Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.136 TotalAliquota05Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 05 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota05Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.137 TotalAliquota06Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 06 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota06Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.138 TotalAliquota07Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 07 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota07Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.139 TotalAliquota08Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 08 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota08Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.140 TotalAliquota09Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 09 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota09Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.141 TotalAliquota10Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 10 na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota10Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.142 TotalAliquota11Reducao

TipoAcesso	R
Descricao	Retorna o total da alíquota 11 na redução "n".
Tipo	money
IndiceMinimo	1

IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota11Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.143 TotalAliquota12Reducao

TipoAcesso R
 Descricao Retorna o total da alíquota 12 na redução "n".
 Tipo money
 IndiceMinimo 1
 IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota12Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.144 TotalAliquota13Reducao

TipoAcesso R
 Descricao Retorna o total da alíquota 13 na redução "n".
 Tipo money
 IndiceMinimo 1
 IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota13Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.145 TotalAliquota14Reducao

TipoAcesso R
 Descricao Retorna o total da alíquota 14 na redução "n".
 Tipo money
 IndiceMinimo 1
 IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota14Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.146 TotalAliquota15Reducao

TipoAcesso R
 Descricao Retorna o total da alíquota 15 na redução "n".
 Tipo money

IndiceMinimo 1
IndiceMaximo NUM_MAX_REDUCOES-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquota15Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.147 TotalAliquotaF1Reducao

TipoAcesso R
Descricao Retorna o total da Substituição Tributária na redução "n".
Tipo money
IndiceMinimo 1
IndiceMaximo NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquotaF1Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.148 TotalAliquotaFS1Reducao

TipoAcesso R
Descricao Retorna o total da Substituição Tributária de ISSQN na redução "n".
Tipo money
IndiceMinimo 1
IndiceMaximo NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquotaFS1Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.149 TotalAliquotaI1Reducao

TipoAcesso R
Descricao Retorna o total da alíquota de Isenção de ICMS na redução "n".
Tipo money
IndiceMinimo 1
IndiceMaximo NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquotaI1Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.150 TotalAliquotaIS1Reducao

TipoAcesso R
Descricao Retorna o total da alíquota de Isenção de ISSQN na redução "n".

Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquotaS1Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.151 TotalAliquotaN1Reducao

TipoAcesso	R
Descricao	Retorna o total de Não Incidência de ICMS na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquotaN1Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.152 TotalAliquotaNS1Reducao

TipoAcesso	R
Descricao	Retorna o total de Não Incidência de ISSQN na redução "n".
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalAliquotaNS1Reducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.153 TotalDiaAcrescimos

TipoAcesso	R
Descricao	Totalizador de Acréscimos diário realizados sobre operações de venda em ICMS
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaAcrescimos', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.154 TotalDiaAcrescimosItens

TipoAcesso	R
Descricao	Total de acrescimos em itens
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaAcrescimosItens', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.155 TotalDiaAcrescimosNaoFiscais

TipoAcesso	R
Descricao	Totalizador de Acréscimos Não Fiscais diário
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaAcrescimosNaoFiscais', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.156 TotalDiaCancelamentoItens

TipoAcesso	R
Descricao	Total de cancelamentos de itens no dia
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaCancelamentoItens', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.157 TotalDiaCancelamentosICMS

TipoAcesso	R
Descricao	Totalizador de Cancelamentos de ICMS diário
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaCancelamentosICMS', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.158 TotalDiaCancelamentosISSQN

TipoAcesso	R
Descricao	Totalizador de Cancelamentos de ISSQN diário
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaCancelamentosISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.159 TotalDiaCancelamentosNaoFiscais

TipoAcesso	R
Descricao	Totalizador de Cancelamentos Não Fiscais diário
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaCancelamentosNaoFiscais', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.160 TotalDiaDescontos

TipoAcesso	R
Descricao	Totalizador de Descontos diário realizados sobre operações de venda em ICMS
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaDescontos', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.161 TotalDiaDescontosItens

TipoAcesso	R
Descricao	Total de descontos em itens
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaDescontosItens', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.162 TotalDiaDescontosNaoFiscais

TipoAcesso	R
Descricao	Totalizador de Descontos Não Fiscais diário
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaDescontosNaoFiscais', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.163 TotalDiaDinheiro

TipoAcesso	R
Descricao	Totalizador do meio de pagamento DINHEIRO pré-cadastrado
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaDinheiro', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.164 TotalDiaImpostoAliquota

TipoAcesso	R
Descricao	Metodo que calcula o imposto devido por situacao tributaria
Tipo	money
IndiceMaximo	15

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaImpostoAliquota', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.165 TotalDiaImpostoICMS

TipoAcesso	R
Descricao	Soma dos impostos tributados pelo ICMS
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaImpostoICMS', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.166 TotalDiaImpostoISSQN

TipoAcesso	R
Descricao	Soma dos impostos tributados pelo ISSQN
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaImpostoISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.167 TotalDiaIsencaoICMS

TipoAcesso	R
Descricao	Totalizador diário da alíquota I (Isencao) de ICMS pré-cadastrado
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```

DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaIsencaoICMS', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);

```

1.7.168 TotalDialsencaoISSQN

TipoAcesso	R
Descricao	Totalizador diário da alíquota I (Isencao) de ISSQN pré-cadastrado
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaIsencaoISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.169 TotalDiaISSQN

TipoAcesso	R
Descricao	Totalizador de Venda diária acumulado nas alíquotas de ISSQN
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.170 TotalDiaMeioPagamento

TipoAcesso	R
Descricao	Seqüência de 'NUM_MEIOS_PAGAMENTO' Totalizadores de Meios de Pagamento diários
Tipo	money
IndiceMaximo	NUM_MEIOS_PAGAMENTO-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaMeioPagamento[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.171 TotalDiaNaoFiscal

TipoAcesso	R
Descricao	Seqüência de 'NUM_MAX_NAO_FISCAIS' Totalizadores Não Fiscais diários
Tipo	money
IndiceMaximo	NUM_MAX_NAO_FISCAIS-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaNaoFiscal[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.172 TotalDiaNaoTributadoICMS

TipoAcesso	R
Descricao	Totalizador diário da alíquota N (Nao Tributado) de ICMS pré-cadastrado

Tipo money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaNaoTributadoICMS', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.173 TotalDiaNaoTributadoISSQN

TipoAcesso R
 Descricao Totalizador diário da alíquota N (Nao Tributado) de ISSQN pré-cadastrado
 Tipo money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaNaoTributadoISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.174 TotalDiaOperacoesNaoFiscais

TipoAcesso R
 Descricao Retorna a soma de operacoes nao fiscais no dia
 Tipo money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaOperacoesNaoFiscais', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.175 TotalDiaQuantidadeVendasDepartamento

TipoAcesso R
 Descricao Sequência de 'NUM_MEIOS_PAGAMENTO' Quantidades Vendidas por Meio de Pagamento no dia
 Tipo money
 IndiceMaximo NUM_DEPARTAMENTOS-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaQuantidadeVendasDepartamento[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.176 TotalDiaSubstituicaoTributariaICMS

TipoAcesso R
 Descricao Totalizador diário da alíquota F (Substituição Tributária) de ICMS pré-cadastrado
 Tipo money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaSubstituicaoTributariaICMS', 7);
```

```
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.177 TotalDiaSubstituicaoTributariaISSQN

TipoAcesso	R
Descricao	Totalizador diário da alíquota F (Substituição Tributária) de ISSQN pré-cadastrado
Tipo	money
IndiceMinimo	
IndiceMaximo	

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaSubstituicaoTributariaISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.178 TotalDiaTrocoTipoAcessorR

Descricao	Totalizador diário de Troco
Tipo	money
IndiceMinimo	
IndiceMaximo	

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaTrocoTipoAcessorR', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.179 TotalDiaValorAliquota

TipoAcesso	R
Descricao	Seqüência de 'NUM_ALIQUOTAS_PROGRAMAVEIS' Totalizadores de Alíquotas diários
Tipo	money
IndiceMaximo	NUM_ALIQUOTAS_PROGRAMAVEIS-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaValorAliquota[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.180 TotalDiaValorEmCaixa

TipoAcesso	R
Descricao	Valor em caixa no dia
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaValorEmCaixa', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
```

```
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.181 TotalDiaValorRecebido

TipoAcesso	R
Descricao	Total recebido nos meios de pagamento
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaValorRecebido', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.182 TotalDiaValorTributadoICMS

TipoAcesso	R
Descricao	Soma dos valores tributados pelo ICMS
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaValorTributadoICMS', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.183 TotalDiaValorTributadoISSQN

TipoAcesso	R
Descricao	Soma dos valores tributados pelo ISSQN
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaValorTributadoISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.184 TotalDiaValorVendasDepartamento

TipoAcesso	R
Descricao	Seqüência de 'NUM_MEIOS_PAGAMENTO' Totalizadores de Venda por Departamento no dia
Tipo	money
IndiceMaximo	NUM_DEPARTAMENTOS-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaValorVendasDepartamento[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.185 TotalDiaVendaBruta

TipoAcesso	R
Descricao	Totalizador de Venda Bruta diária
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaVendaBruta', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.186 TotalDiaVendaLiquida

TipoAcesso	R
Descricao	Venda líquida no dia.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDiaVendaLiquida', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.187 TotalDocAcrecimos

TipoAcesso	R
Descricao	Total de Acréscimos realizados no cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocAcrecimos', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.188 TotalDocAliquota

TipoAcesso	R
Descricao	Totalizador de cada alíquota referente a venda no cupom atual.
Tipo	money
IndiceMaximo	NUM_ALIQUOTAS_PROGRAMAVEIS-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocAliquota[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.189 TotalDocBruto

TipoAcesso	R
Descricao	Total Bruto do cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocBruto', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.190 TotalDocCancelamentoICMS

TipoAcesso	R
Descricao	Refere-se ao total cancelado em ICMS do cupom fiscal corrente. Este totalizador é zerado ao iniciar cada cupom fiscal.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocCancelamentoICMS', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.191 TotalDocCancelamentoISSQN

TipoAcesso	R
Descricao	Refere-se ao total cancelado em ISSQN do cupom fiscal corrente. Este totalizador é zerado ao iniciar cada cupom fiscal.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocCancelamentoISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.192 TotalDocCancelamentosNaoFiscais

TipoAcesso	R
Descricao	Retorna o total cancelado em cupom nao fiscal
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocCancelamentosNaoFiscais', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.193 TotalDocDescontos

TipoAcesso	R
Descricao	Total de Descontos realizados no cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocDescontos', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
```

```
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.194 TotalDocDinheiro

TipoAcesso	R
Descricao	Total pago em dinheiro no cupom atual
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocDinheiro', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.195 TotalDocIsencaolCMS

TipoAcesso	R
Descricao	Totalizador da alíquota I (Isenca) ICMS referente a venda no cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocIsencaolCMS', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.196 TotalDocIsencaolISSQN

TipoAcesso	R
Descricao	Totalizador da alíquota I (Isenca) ISSQN referente a venda no cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocIsencaolISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.197 TotalDocISSQN

TipoAcesso	R
Descricao	Total vendido em ISSQN no cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.198 TotalDocLiquido

TipoAcesso	R
Descricao	Retorna o total liquido do cupom

Tipo money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocLiquido', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.199 TotalDocMeioPagamento

TipoAcesso R
 Descricao Total pago em cada meio de pagamento no cupom atual.
 Tipo money
 IndiceMaximo NUM_MEIOS_PAGAMENTO_PROGRAMAVEIS-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocMeioPagamento[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.200 TotalDocNaoFiscal

TipoAcesso R
 Descricao Total acumulado por operacao nao fiscal
 Tipo money
 IndiceMaximo NUM_MAX_NAO_FISCAIS-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocNaoFiscal[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.201 TotalDocNaoTributadoICMS

TipoAcesso R
 Descricao Totalizador da alíquota N (Não Tributado) ICMS referente a venda no cupom atual.
 Tipo money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocNaoTributadoICMS', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.202 TotalDocNaoTributadoISSQN

TipoAcesso R
 Descricao Totalizador da alíquota N (Não Tributado) ISSQN referente a venda no cupom atual.
 Tipo money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocNaoTributadoISSQN', 7);
```

```
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.203 TotalDocSubstituicaoTributariaICMS

TipoAcesso	R
Descricao	Totalizador da alíquota F (Substituicao Tributaria) ICMS referente a venda no cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocSubstituicaoTributariaICMS', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.204 TotalDocSubstituicaoTributariaISSQN

TipoAcesso	R
Descricao	Totalizador da alíquota F (Substituicao Tributaria) ISSQN referente a venda no cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocSubstituicaoTributariaISSQN', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.205 TotalDocTransacoesVenda

TipoAcesso	R
Descricao	Informa o total de transações de venda efetuados no cupom atual.
Tipo	uint

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'TotalDocTransacoesVenda', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.206 TotalDocTroco

TipoAcesso	R
Descricao	Total de Troco do cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocTroco', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.207 TotalDocValorPago

TipoAcesso	R
Descricao	Total dos meios de pagamento do cupom atual.
Tipo	money

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'TotalDocValorPago', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.208 TradutorIndiceMeioPagamentoDinheiro

TipoAcesso	RW
Descricao	Informa qual o índice de Meio de Pagamento da Logger II o tradutor de protocolos da Logger I deve usar quando receber o Meio de pagamento dinheiro.
Tipo	int

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'TradutorIndiceMeioPagamentoDinheiro', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.209 UltimoCmdNumero

TipoAcesso	R
Descricao	Retorna o Número de Identificação do último comando executado com sucesso. Somente comandos que desencadeiem alguma ação no ECF como impressão ou gravação de dados na MFD afetam esta variável. Comandos de leitura de dados não alteram este registrador.
Tipo	uint

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'UltimoCmdNumero', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.210 UltimoErroCircunstancia

TipoAcesso	R
Descricao	Circunstância do último erro (comando executado sem sucesso) retornado pelo ECF. Após o ECF ser ligado e antes de ocorrer algum comando com erro o retorno será um texto vazio.
Tipo	string
TamMaximo	250

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'UltimoErroCircunstancia', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.211 UltimoErroCodigo

TipoAcesso	R
Descricao	Código do último erro (comando executado sem sucesso) retornado pelo ECF. Após o ECF ser ligado e antes de ocorrer algum comando com erro o retorno será zero.
Tipo	uint

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeInteiro.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeInteiro', 'UltimoErroCodigo', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeInteiro'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.212 UltimoErroNome

TipoAcesso	R
Descricao	Nome do último erro (comando executado sem sucesso) retornado pelo ECF. Após o ECF ser ligado e antes de ocorrer algum comando com erro o retorno será um texto vazio.
Tipo	string
TamMaximo	50

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'UltimoErroNome', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.213 VendaBrutaReducao

TipoAcesso	R
Descricao	Retorna a venda bruta da redução "n"
Tipo	money
IndiceMinimo	1
IndiceMaximo	NUM_MAX_REDUCOES-1

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeMoeda.

Exemplo em Delphi:

```
DLLG2_AdicionaParam(hPorta, 'NomeDadoMonetario', 'VendaBrutaReducao[1]', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeMoeda'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.7.214 VersaoSW

TipoAcesso	R
Descricao	Versão do software básico
Tipo	string
TamMaximo	8

Este registrador pode apenas ser lido. Para ler este registrador deve ser utilizado o comando LeTexto.

Exemplo:

```
DLLG2_AdicionaParam(hPorta, 'NomeTexto', 'VersaoSW', 7);
TrataErro(DLLG2_ExecutaComando(hPorta, 'LeTexto'));
strBuffer := StringOfChar(' ',100);
strBuffer := DLLG2_ObtemRetornos(hPorta, strBuffer, 100);
```

1.8 Retornos do Protocolo FiscNET

0	NoError	Não houve erro algum.
1	ErroGeralFaltaRAM	Não foi possível alocar mais memória.
2	ErroGeralPerdaRAM	Memória RAM foi corrompida.
1000	ErroMFDesconectada	Memória Fiscal foi desconectada.
1001	ErroMFLeitura	Erro de leitura na Memória Fiscal.
1002	ErroMFApenasLeitura	Memória está setada apenas para leitura.
1003	ErroMFTamRegistro	Registro fora dos padrões (erro interno).
1004	ErroMFCheia	Memória Fiscal está lotada.
1005	ErroMFCartuchosExcedidos	Número máximo de cartuchos excedidos.
1006	ErroMFJaInicializada	Tentativa de gravar novo modelo de ECF.
1007	ErroMFNaoInicializada	Tentativa de gravação de qualquer dado antes da inicialização da Memória Fiscal.
1008	ErroMFUsuariosExcedidos	Número máximo de usuários foi atingido.
1009	ErroMFIntervencoesExcedidas	Número máximo de intervenções foi atingido.
1010	ErroMFVersoesExcedidas	Número máximo de versões foi atingido.
1011	ErroMFReducoesExcedidas	Número máximo de reduções foi atingido.
1012	ErroMFGravacao	Erro na gravação de registro na memória fiscal
2000	ErroTransactDrvrLeitura	Erro de leitura no dispositivo físico.
2001	ErroTransactDrvrEscrita	Erro de leitura no dispositivo.
2002	ErroTransactDrvrDesconectado	Dispositivo de transações foi desconectado.
3000	ErroTransactRegInvalido	Tipo de registro a ser gravado inválido.
3001	ErroTransactCheio	Registro de transações está esgotado.
3002	ErroTransactTransAberta	Tentativa de abrir nova transação com transação já aberta.
3003	ErroTransactTransNaoAberta	Tentativa de fechar uma transação que não se encontrava
4000	ErroContextDrvrLeitura	Erro de leitura no dispositivo físico.
4001	ErroContextDrvrEscrita	Erro de escrita no dispositivo.
4002	ErroContextDrvrDesconectado	Dispositivo de contexto foi desconectado.
4003	ErroContextDrvrLeituraAposFim	Leitura após final do arquivo.
4004	ErroContextDrvrEscritaAposFim	Escrita após final do arquivo.
5000	ErroContextVersaoInvalida	Versão de contexto fiscal no dispositivo não foi reconhecida.
5001	ErroContextCRC	CRC do dispositivo está incorreto.
5002	ErroContextLimitesExcedidos	Tentativa de escrita fora da área de contexto.
6000	ErroRelogioInconsistente	Relógio do ECF inconsistente.
6001	ErroRelogioDataHoraInvalida	Data/hora informadas não estão consistentes.
7000	ErroPrintSemMecanismo	Nenhum mecanismo de impressão presente.
7001	ErroPrintDesconectado	Atual mecanismo de impressão está desconectado.
7002	ErroPrintCapacidadeInexistente	Mecanismo não possui capacidade suficiente para realizar esta operação.

7003	ErroPrintSemPapel	Impressora está sem papel para imprimir.
7004	ErroPrintFaltouPapel	Faltou papel durante a impressão do comando.
8000	ErroCMDForaDeSequencia	Comando fora de seqüência.
8001	ErroCMDCodigoInvalido	Código mercadoria não válido.
8002	ErroCMDDescricaoInvalida	Descrição inválida.
8003	ErroCMDQuantidadeInvalida	Quantidade não inválida.
8004	ErroCMDAliquotaInvalida	Índice da alíquota não válido.
8005	ErroCMDAliquotaNaoCarregada	Alíquota não carregada.
8006	ErroCMDValorInvalido	Valor contém caracter inválido.
8007	ErroCMDMontanteOperacao	Total da operação igual a 0 (zero).
8008	ErroCMDAliquotaIndisponivel	Alíquota não disponível para carga.
8009	ErroCMDValorAliquotaInvalido	Valor da alíquota não válido.
8010	ErroCMDTrocaSTAposFechamento	Troca de situação tributária somente após Redução Z.
8011	ErroCMDFormaPagamentoInvalida	Índice do Meio de Pagamento não válido.
8012	ErroCMDPayIndisponivel	Meio de Pagamento indisponível para carga.
8013	ErroCMDCupomTotalizadoEmZero	Cupom totalizado em 0 (zero).
8014	ErroCMDFormaPagamentoIndefinida	Meio de Pagamento não definido.
8015	ErroCMDTrocaUsuarioAposFechamento	Carga de usuário permitido somente após Redução Z.
8016	ErroCMDSemMovimento	Dia sem movimento.
8017	ErroCMDPagamentoIncompleto	Total pago inferior ao total do cupom.
8018	ErroCMDGerencialNaoDefinido	Gerencial não definido.
8019	ErroCMDGerencialInvalido	Índice do Gerencial fora da faixa.
8020	ErroCMDGerencialIndisponivel	Gerencial não disponível para carga.
8021	ErroCMDNomeGerencialInvalido	Nome do Gerencial inválido.
8022	ErroCMDNaoHaMaisRelatoriosLivres	Esgotado número de Gerenciais.
8023	ErroCMDAcertoHVPermitidoAposZ	Acerto do horário de verão somente após a Redução Z.
8024	ErroCMDHorarioVeraoJaRealizado	Já acertou horário de verão.
8025	ErroCMDAliquotasIndisponiveis	Sem Alíquotas disponíveis para carga.
8026	ErroCMDItemInexistente	Item não vendido no cupom.
8027	ErroCMDQtdCancInvalida	Quantidade a ser cancelada maior do que a quantidade vendida.
8028	ErroCMDCampoCabecalhoInvalido	Cabeçalho possui campo(s) inválido(s).
8029	ErroCMDNomeDepartamentoInvalido	Nome do Departamento não válido.
8030	ErroCMDDepartamentoNaoEncontrado	Departamento não encontrado.
8031	ErroCMDDepartamentoIndefinido	Departamento não definido.
8032	ErroCMDFormasPagamentosIndisponiveis	Não há Meio de Pagamento disponível.
8033	ErroCMDAltPagamentoSoAposZ	Alteração de Meio de Pagamento somente após a Redução Z.
8034	ErroCMDNomeNaoFiscalInvalido	Nome do Documento Não Fiscal não pode ser vazio.
8035	ErroCMDDocsNaoFiscaisIndisponiveis	Não há mais Documentos Não Fiscais disponíveis.

8036	ErroCMDNaoFiscalIndisponivel	Documento Não Fiscal indisponível para carga.
8037	ErroCMDReducaoInvalida	Número da redução inicial inválida.
8038	ErroCMDCabecalhoJaImpresso	Cabeçalho do documento já foi impresso.
8039	ErroCMDLinhasSuplementaresExcedidas	Número máximo de linhas de propaganda excedidas.
8040	ErroCMDHorarioVeraoJaAtualizado	Relógio já está no estado desejado.
8041	ErroCMDValorA acrescimoInvalido	Valor do acréscimo inconsistente.
8042	ErroCMDNaoHaMeiodePagamento	Não há meio de pagamento definido.
8043	ErroCMDCOOVinculadoInvalido	COO do documento vinculado inválido.
8044	ErroCMDIndiceItemInvalido	Índice do item inexistente no contexto.
8045	ErroCMDCodigoNaoEncontrado	Código de item não encontrado no cupom atual.
8046	ErroCMDPercentualDescontoInvalido	Percentual do desconto ultrapassou 100%.
8047	ErroCMDDescontoItemInvalido	Desconto do item inválido.
8048	ErroCMDFaltaDefinirValor	Falta definir valor percentual ou absoluto em operação de desconto/acréscimo.
8049	ErroCMDItemCancelado	Tentativa de operação sobre item cancelado.
8050	ErroCMDCancelaAcrDescInvalido	Cancelamento de acréscimo/desconto inválidos.
8051	ErroCMDAcrDescInvalido	Operação de acréscimo/desconto inválida.
8052	ErroCMDNaoHaMaisDepartamentosLivres	Número de Departamentos esgotados.
8053	ErroCMDIndiceNaoFiscalInvalido	Índice de Documento Não Fiscal fora da faixa.
8054	ErroCMDTrocaNaoFiscalAposZ	Troca de Documento Não Fiscal somente após a Redução Z.
8055	ErroCMDInscricaoInvalida	CNPJ e/ou Inscrição Estadual inválida(s).
8056	ErroCMDVinculadoParametrosInsuficientes	Falta(m) parâmetro(s) no comando de abertura de Comprovante Crédito ou Débito.
8057	ErroCMDNaoFiscalIndefinido	Código e Nome do Documento Não Fiscal indefinidos.
8058	ErroCMDFaltaAliquotaVenda	Alíquota não definida no comando de venda.
8059	ErroCMDFaltaMeioPagamento	Código e Nome do Meio de Pagamento não definidos.
8060	ErroCMDFaltaParametro	Parâmetro de comando não informado.
8061	ErroCMDNaoHaDocNaoFiscaisDefinidos	Não há Documentos Não Fiscais definidos.
8062	ErroCMDOperacaoJaCancelada	Acréscimo/Desconto de item já cancelado.
8063	ErroCMDNaoHaAcrescDescItem	Não há acréscimo/desconto em item.
8064	ErroCMDItemAcrescido	Item já possui acréscimo.
8065	ErroCMDOperSoEmICMS	Operação de acréscimo em item ou subtotal só é válido para ICMS
8066	ErroCMDFaltaInformarValor	Valor do Comprovante Crédito ou Débito não informado.
8067	ErroCMDCOOInvalido	COO inválido.
8068	ErroCMDIndiceInvalido	Índice do Meio de Pagamento no cupom inválido.
8069	ErroCMDCupomNaoEncontrado	Documento Não Fiscal não encontrado.
8070	ErroCMDSequenciaPagamentoNaoEncontrada	Seqüência de pagamento não encontrada no cupom.
8071	ErroCMDPagamentoNaoPermiteCDC	Meio de pagamento não permite CDC.
8072	ErroCMDUltimaFormaPagamentoInv	Valor insuficiente para pagar o cupom.
8073	ErroCMDMeioPagamentoNEncontrado	Meio de pagamento origem ou destino não encontrado no último cupom emitido
8074	ErroCMDValorEstornoInvalido	Valor do estorno não pode exceder o valor do pagamento no meio origem.

8075	ErroCMDMeiosPagamentoOrigemDestinoIguais	Meios de pagamento origem e destino devem ser diferentes no estorno.
8076	ErroCMDPercentualInvalido	Percentual da alíquota inválido.
8077	ErroCMDNaoHouveOpSubtotal	Não houve operação em subtotal para ser cancelada.
8078	ErroCMDOpSubtotalInvalida	Só é permitida uma operação de acréscimo em subtotal por cupom.
8079	ErroCMDTextoAdicional	Texto adicional do meio de pagamento deve ter no máximo 2 linhas
8080	ErroCMDPrecoUnitarioInvalido	Preço unitário ultrapassou o número máximo de dígitos permitido
8081	ErroCMDDepartamentoInvalido	Código do departamento fora da faixa.
8082	ErroCMDDescontoInvalido	O valor do desconto não pode zerar o valor do cupom ou ser maior que o item.
8083	ErroCMDPercentualAcrescimoInvalido	Percentual de acréscimo não pode ser superior a 999,99%
8084	ErroCMDAcrescimoInvalido	Valor do acréscimo ultrapassa o número máximo de dígitos permitido (13 dígitos)
8085	ErroCMDNaoHouveVendaEmICMS	Cupom sem venda em alíquota de ICMS.
8086	ErroCMDCancelamentoInvalido	Cancelamento inválido.
8087	ErroCMDCliche	Texto de clique do usuário deve ter no máximo três linhas.
8088	ErroCMDNaoHouveVendaNaoFiscal	Não houve venda de item não fiscal
8089	ErroCMDDataInvalida	A data não pode ser inferior a data do último documento emitido.
8090	ErroCMDHoraInvalida	A hora informada no comando não pode ser inferior ao horário do último documento
8091	ErroCMDEstorno	Sem função
8092	ErroCMDAcertoRelogio	Estado inválido para ajuste de relógio ou horário de verão
8093	ErroCMDCDCInvalido	A operação de CDC deve preceder as operações de estorno de meio de pagamento.
8094	ErroCMDSenhaInvalida	Senha inválida para inicialização do proprietário.
8095	ErroCMDMecanismoCheque	Erro gerado pelo mecanismo de cheques
8096	ErroFaltaIniciarDia	Comando válido somente após a abertura do dia
9000	ErroMFDNenhumCartuchoVazio	Não foi encontrado nenhum cartucho de dados vazio para ser inicializado.
9001	ErroMFDCartuchoInexistente	Cartucho com o número de série informado não foi encontrado.
9002	ErroMFDNumSerie	Número de série do ECF é inválido na inicialização.
9003	ErroMFDCartuchoDesconectado	Cartucho de MFD desconectado ou com problemas.
9004	ErroMFDEscrita	Erro de escrita no dispositivo de MFD.
9005	ErroMFDSseek	Erro na tentativa de posicionar ponteiro de leitura.
9006	ErroMFDBadBadSector	Endereço do Bad Sector informado é inválido.
9007	ErroMFDLeitura	Erro de leitura na MFD.
9008	ErroMFDLeituraAlemEOF	Tentativa de leitura além dos limites da MFD.
9009	ErroMFDEsgotada	MFD não possui mais espaço para escrita.
9010	ErroMFDLeituraInterrompida	Leitura da MFD serial é interrompida por comando diferente de LeImpressao
10000	ErroBNFEstadoInvalido	Estado inválido para registro sendo codificado.
10001	ErroBNFParametroInvalido	Inconsistência nos parâmetros lidos no Logger.
10002	ErroBNFRegistroInvalido	Registro inválido detectado no Logger.
10003	ErroBNFErroMFD	Erro interno.

11000	ErroProtParamInvalido	Parâmetro repassado ao comando é inválido.
11001	ErroProtParamSintaxe	Erro de sintaxe na lista de parâmetros.
11002	ErroProtParamValorInvalido	Valor inválido para parâmetro do comando.
11003	ErroProtParamStringInvalido	String contém seqüência de caracteres inválidos.
11004	ErroProtParamRedefinido	Parâmetro foi declarado 2 ou mais vezes na lista.
11005	ErroProtParamIndefinido	Parâmetro obrigatório ausente na lista.
11006	ErroProtComandoInexistente	Não existe o comando no protocolo.
11007	ErroProtSequenciaComando	Estado atual não permite a execução deste comando.
11008	ErroProtAborta2aVia	Sinalização indicando que comando aborta a impressão da segunda via.
11009	ErroProtSemRetorno	Sinalização indicando que comando não possui retorno.
11010	ErroProtTimeout	Tempo de execução esgotado.
11011	ErroProtNomeRegistrador	Nome de registrador inválido.
11012	ErroProtTipoRegistrador	Tipo de registrador inválido.
11013	ErroProtSomenteLeitura	Tentativa de escrita em registrador de apenas leitura.
11014	ErroProtSomenteEscrita	Tentativa de leitura em registrador de apenas escrita.
11015	ErroProtComandoDiferenteAnterior	Comando recebido diferente do anterior no buffer de recepção.
11016	ErroProtFilaCheia	Fila de comandos cheia.
11017	ErroProtIndiceRegistrador	Índice de registrador indexado fora dos limites.
11018	ErroProtNumEmissoesExcedido	Número de emissões do Logger foi excedido na Intervenção Técnica.
11019	ErroMathDivisaoPorZero	Divisão por 0 (zero) nas rotinas de BDC.
15001	ErroApenasIntTecnica	Comando aceito apenas em modo de Intervencao Técnica.
15002	ErroECFIntTecnica	Comando não pode ser executado em modo de Intervenção Técnica.
15003	ErroMFDPresente	Já existe MFD presente neste ECF.
15004	ErroSemMFD	Não existe MFD neste ECF.
15005	ErroRAMInconsistente	Memória RAM do ECF não está consistente.
15006	ErroMemoriaFiscalDesconectada	Memória fiscal não encontrada.
15007	ErroDiaFechado	Dia já fechado.
15008	ErroDiaAberto	Dia aberto.
15009	ErroZPendente	Falta reducao Z.
15010	ErroMecanismoNaoConfigurado	Mecanismo impressor não selecionado.
15011	ErroSemPapel	Sem bobina de papel na estação de documento fiscal.
15012	ErroDocumentoEncerrado	Tentativa de finalizar documento já encerrado.
15013	ErroSemSinalDTR	Não há sinal de DTR.
15014	ErroSemInscricoes	Sem inscrições do usuário no ECF.
15015	ErroSemCliche	Sem dados do proprietário no ECF.
15016	ErroEmLinha	ECF encontra-se indevidamente em linha.
15017	ErroForaDeLinha	ECF não encontra-se em linha para executar o comando.
15018	ErroMecanismoBloqueado	Mecanismo está indisponível para impressão.

1.9 Comandos do Protocolo FiscNET

Para executar qualquer comando usando a dllg2 deve ser seguida a seguinte seqüência:

- 1- Limpar porta serial;
- 2- Adicionar os parâmetros;
- 3- Executar o comando;

Este tópico relaciona a lista de todos os comandos do protocolo FiscNet.

Para maiores informações, sobre a utilização da DLL consulte o documento [Manual da DLL-G2](#).

1.9.1 AbreCreditoDebito

Descrição: Inicia a emissão de um Cupom de Crédito ou Débito (CCD). Imprime o cabeçalho e dos dados referente ao Meio de Pagamento que habilitou sua impressão.

Observações: Pode-se gerar um CCD relativo a Meios de Pagamento emitidos em Cupons Fiscais (CF) ou Não-Fiscais (CNF), desde que o Meio de Pagamento tenha sido definido para habilitar CCD (ou

Parâmetros

Variável: CodMeioPagamento

TipoDado: [byte](#)

Tamanho Máximo:

Opcional

Descrição: Índice do Meio de Pagamento, sendo -2 ou um valor no intervalo entre 0 (zero) e NUM_MEIOS_PAGAMENTO, onde: -2 representa o Meio de Pagamento pré-definido como "Dinheiro"; qualquer valor do intervalo representa o índice do meio de pagamento programável. Identifica o Meio de Pagamento que habilitou a emissão do CCD a ser impresso. Quando definido, este parâmetro deve ser usado em conjunto com o parâmetro <Valor>.

Variável: COO

TipoDado: [ulong](#)

Tamanho Máximo:

Opcional

Descrição: Contador de Ordem de Operação. COO do cupom que houve o Meio de Pagamento relacionado ao CCD. Quando não definido, assume o COO do último documento emitido.

Variável: NomeMeioPagamento

TipoDado: [string](#)

Tamanho Máximo: 16

Opcional

Descrição: Nome do Meio de Pagamento. Identifica o Meio de Pagamento que habilitou a emissão do CCD a ser impresso. Quando definido, este parâmetro deve ser usado em conjunto com o parâmetro <Valor>. Parâmetro ignorado quando usado com <CodMeioPagamento>.

Variável: NumItem

TipoDado: [int](#)

Tamanho Máximo:

Opcional

Descrição: Número seqüencial de lançamento do item a que se refere esta operação no cupom em emissão. Seqüência de emissão do Meio de Pagamento dentro de um cupom. O primeiro Meio de Pagamento lançado em qualquer cupom possui o valor <NumItem> = 0 (zero), o próximo como 1, e assim sucessivamente.

Variável: NumParcelas

TipoDado: [byte](#)

Tamanho Máximo: 0

Opcional

Descrição: Número de parcelas do Comprovante de Crédito ou Débito.

Variável: Valor

TipoDado: [money](#)

Tamanho Máximo:

Opcional

Descrição: Valor da operação. Juntamente c/ <CodMeioPagamento> ou <NomeMeioPagamento> e <COO>, identifica o pagamento que habilitou a emissão do CCD. Havendo duplicidade de valores, considera o 1º ainda não impresso. Ignorado qdo usado com <NumItem>

TÓPICOS RELACIONADOS:

- [Comprovante de Crédito ou Débito](#)
- [Tipos de Parâmetros](#)

1.9.2 AbreCupomFiscal

Descrição: Abre um Cupom Fiscal (CF), imprimindo o cabeçalho do cupom, o cabeçalho de venda e opcionalmente os dados do consumidor.

Parâmetros

Variável: EnderecoConsumidor

TipoDado: [string](#)

Tamanho Máximo: 80

Opcional

Descrição: Endereço do consumidor.

Variável: IdConsumidor

TipoDado: [string](#)

Tamanho Máximo: 29

Opcional

Descricao: Identificação do consumidor.

Variavel: NomeConsumidor

TipoDado: [string](#) **Tamanho Máximo:** 30 **Opcional**

Descricao: Nome do consumidor.

Exemplo:

```
DLLG2_LimpaParams(Port);
DLLG2_AdicionaParam(Port, 'EnderecoConsumidor', EndConsumidor.text, 7);
DLLG2_AdicionaParam(Port, 'IdConsumidor', IdConsumidor.text, 7);
DLLG2_AdicionaParam(Port, 'NomeConsumidor', NomeConsumidor.text, 7);
retorno := DLLG2_ExecutaComando(Port, 'AbreCupomFiscal');
```

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)
- [Tipos de Parâmetros](#)

1.9.3 AbreCupomNaoFiscal

Descrição: Abre um Cupom Não-Fiscal (CNF), imprimindo o cabeçalho do cupom e opcionalmente a identificação do consumidor.

Parâmetros

Variavel: EnderecoConsumidor

TipoDado: [string](#) **Tamanho Máximo:** 80 **Opcional**

Descricao: Endereço do consumidor.

Variavel: IdConsumidor

TipoDado: [string](#) **Tamanho Máximo:** 29 **Opcional**

Descricao: Identificação do consumidor.

Variavel: NomeConsumidor

TipoDado: [string](#) **Tamanho Máximo:** 30 **Opcional**

Descricao: Nome do consumidor.

TÓPICOS RELACIONADOS:

- [Comprovante Não Fiscal](#)
- [Tipos de Parâmetros](#)

1.9.4 AbreGaveta

Descrição: Abre a gaveta de valores conectada ao equipamento.

Parâmetros

Variavel: TempoAcionamentoGaveta

TipoDado: [int](#) **Tamanho Máximo:** **Opcional**

Descricao: Tempo, em milissegundos, de acionamento do solenóide.

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)
- [Comprovante Não Fiscal](#)
- [Tipos de Parâmetros](#)

1.9.5 AbreGerencial

Descrição: Abre um Relatório Gerencial, imprimindo seu cabeçalho e nome.

Observações: É possível informar o tipo de Relatório Gerencial através dos parâmetros. Quando não informado, será usado o último tipo de Relatório Gerencial impresso ou 0 (zero) quando tratar-se do primeiro emitido após uma Redução Z.

Parâmetros

Variavel: CodGerencial

TipoDado: [ubyte](#) **Tamanho Máximo:** **Opcional**

Descrição: Índice do relatório gerencial entre 0 e NUM_GERENCIAIS. Identifica o Relatório Gerencial. Este parâmetro tem precedência a <NomeGerencial>.

Variável: NomeGerencial

TipoDado: [string](#)

Tamanho Máximo:

30 Opcional

Descrição: Nome do Relatório Gerencial. Identifica o Relatório Gerencial.

Exemplo:

```
dLLG2_LimpaParams(hPorta);
dLLG2_AdicionaParam(hPorta,'CodGerencial' , '01',4);
TrataErro(DLLG2_ExecutaComando(Porta,'AbreGerencial'));
dLLG2_AdicionaParam(Porta,'TextoLivre' , 'Testando relatório Gerencial',7);
TrataErro(DLLG2_ExecutaComando(Porta,'ImprimeTexto'));
TrataErro(DLLG2_ExecutaComando(Porta,'EncerraDocumento'));
```

TÓPICOS RELACIONADOS:

- [Relatório Gerencial](#)
- [Tipos de Parâmetros](#)

1.9.6 AcertaHorarioVerao

Descrição: Acerta o relógio para entrada ou saída do horário de verão, aumentando ou diminuindo 1 hora.

Observações: Se o parâmetro não for informado, inverte a situação do Horário de Verão (entrada/saída).

Parâmetros

Variável: EntradaHV

TipoDado: [byte](#)

Tamanho Máximo:

Opcional

Descrição: 0 (zero) = saída de HV; 1 = entrada de HV.

TÓPICOS RELACIONADOS:

- [Horário de Verão](#)
- [MIT](#)
- [Tipos de Parâmetros](#)

1.9.7 AcresceltemFiscal

Descrição: Executa uma operação de acréscimo/desconto em item lançado no Cupom Fiscal aberto. Também cancela uma operação de desconto/acrécimo previamente emitida.

Observações: Permite executar a operação sobre qualquer item registrado. Se <NumItem> não for informado, a operação será realizada sobre o último item emitido. Pelo menos um dos valores deve ser informado, se a operação não for cancelada.

Parâmetros

Variável: Cancelar

TipoDado: [bool](#)

Tamanho Máximo:

Obrigatório

Descrição: Indicador de cancelamento da operação. Quando 'true' (cancelar acréscimo/desconto de item), tem precedência sobre os demais parâmetros deste comando.

Variável: NumItem

TipoDado: [int](#)

Tamanho Máximo:

Opcional

Descrição: Número sequencial de lançamento do item a que se refere esta operação no cupom em emissão.

Variável: ValorAcrescimo

TipoDado: [money](#)

Tamanho Máximo:

Opcional

Descrição: Valor do desconto (quando negativo) ou acréscimo (quando positivo). Este parâmetro tem precedência sobre o parâmetro <ValorPercentual>.

Variável: ValorPercentual

TipoDado: [money](#)

Tamanho Máximo:

Opcional

Descrição: Percentual de desconto (quando negativo) ou acréscimo (quando positivo), com precisão máxima de 2 casas decimais. Este parâmetro será desconsiderado quando usado em conjunto com o parâmetro <ValorAcrescimo>.

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)
- [Tipos de Parâmetros](#)

1.9.8 AcresceltemNaoFiscal

Descrição: Executa uma operação de acréscimo/desconto em item não fiscal lançado em Cupom Não-Fiscal (CNF) aberto. Também cancela uma operação de desconto/acrécimo previamente emitida.

Observações: Permite executar a operação sobre qualquer item registrado. Se <NumItem> não for informado, a operação será realizada sobre o último item emitido. Pelo menos um dos valores deve ser informado, se a operação não for cancelada.

Parâmetros

Variável: Cancelar

TipoDado: [bool](#)

Tamanho Máximo:

Obrigatório

Descrição: Indicador de cancelamento da operação. Quando 'true' (cancelar acréscimo/desconto de item), tem precedência sobre os demais parâmetros deste comando.

Variável: NumItem

TipoDado: [int](#)

Tamanho Máximo:

Opcional

Descrição: Número seqüencial de lançamento do item a que se refere esta operação no cupom em emissão.

Variável: ValorAcrescimo

TipoDado: [money](#)

Tamanho Máximo:

Opcional

Descrição: Valor do desconto (quando negativo) ou acréscimo (quando positivo). Este parâmetro tem precedência sobre o parâmetro <ValorPercentual>.

Variável: ValorPercentual

TipoDado: [money](#)

Tamanho Máximo:

Opcional

Descrição: Percentual de desconto (quando negativo) ou acréscimo (quando positivo), com precisão máxima de 2 casas decimais. Este parâmetro será desconsiderado quando usado em conjunto com o parâmetro <ValorAcrescimo>.

TÓPICOS RELACIONADOS:

- [Comprovante Não Fiscal](#)
- [Tipos de Parâmetros](#)

1.9.9 AcresceSubtotal

Descrição: Realiza operações de acréscimo/desconto no subtotal no cupom aberto (Fiscal ou Não-Fiscal). Este comando também permite cancelar a última operação de subtotal realizada no cupom.

Parâmetros

Variável: Cancelar

TipoDado: [bool](#)

Tamanho Máximo:

Obrigatório

Descrição: Indicador de cancelamento da operação. Se este parâmetro for informado (= 'true'), cancela o último desconto/acrécimo de subtotal informado. Este parâmetro tem precedência sobre os demais neste comando.

Variável: ValorAcrescimo

TipoDado: [money](#)

Tamanho Máximo:

Opcional

Descrição: Valor do desconto (quando negativo) ou acréscimo (quando positivo). Este parâmetro tem precedência sobre o <ValorPercentual> se informado.

Variável: ValorPercentual

TipoDado: [money](#)

Tamanho Máximo:

Opcional

Descrição: Percentual de desconto (quando negativo) ou acréscimo (quando positivo), com precisão máxima de 2 casas decimais. Esta operação é realizada sobre o subtotal (Líquido) do cupom. Este parâmetro será desconsiderado quando usado em conjunto com o parâmetro <ValorAcrescimo>.

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)
- [Comprovante Não Fiscal](#)

- [Tipos de Parâmetros](#)

1.9.10 AvancaPapel

Descrição: Avança o papel por um certo número de dots para espaçar impressões.

Parâmetros

Variável: Avanco

TipoDado: [uint](#)

Tamanho Máximo:

Obrigatório

Descricao: Número de dots a serem avançados.

TÓPICOS RELACIONADOS:

- [Mecanismos](#)

- [Tipos de Parâmetros](#)

1.9.11 CancelaCupom

Descrição: Cancela o cupom em andamento ou o último cupom encerrado.

Observações: Aplica-se a Cupom Fiscal (CF) e Cupom Não-Fiscal (CNF).

Parâmetros

Variável: Operador

TipoDado: [string](#)

Tamanho Máximo:

8 Opcional

Descricao: Identificação do operador.

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)

- [Comprovante Não Fiscal](#)

- [Tipos de Parâmetros](#)

1.9.12 CancelaItemFiscal

Descrição: Cancela total ou parcialmente item emitido em um Cupom Fiscal (CF).

Observações: Se nenhum parâmetro for informado, cancela o último item registrado. Não permite o cancelamento parcial de item com quantidade que possui mais de 2 (duas) casas decimais.

Parâmetros

Variável: NumItem

TipoDado: [int](#)

Tamanho Máximo:

Opcional

Descricao: Número seqüencial de lançamento do item a que se refere esta operação no cupom em emissão.

Variável: Quantidade

TipoDado: [money](#)

Tamanho Máximo:

Opcional

Descricao: Quantidade envolvida na transação. Quando este parâmetro não for informado, cancela quantidade total referente ao item ou ao código.

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)

- [Tipos de Parâmetros](#)

1.9.13 CancelaItemNaoFiscal

Descrição: Cancela item emitido em um Cupom Não-Fiscal.

Observações: Se nenhum parâmetro for informado, cancela o último item registrado.

Parâmetros

Variável: NumItem

TipoDado: [int](#)

Tamanho Máximo:

Opcional

Descricao: Número seqüencial de lançamento do item a que se refere esta operação no cupom em emissão.

TÓPICOS RELACIONADOS:

- [Comprovante Não Fiscal](#)

- [Tipos de Parâmetros](#)

1.9.14 ChancelaCheque

Descrição: Cancela cheque.

Parâmetros

Variavel: HPosChancelamento

TipoDado: [uint](#)

Tamanho Máximo:

Obrigatório

Descricao: Posição horizontal do início de impressão da chancela do cheque em 1/10 mm.

Variavel: TextoChancela

TipoDado: [string](#)

Tamanho Máximo: 18

Obrigatório

Descricao: Texto da chancela.

Variavel: VPosChancelamento

TipoDado: [uint](#)

Tamanho Máximo:

Obrigatório

Descricao: A partir do topo do cheque, posição vertical da base do caracter referente ao início de impressão da chancela do cheque em 1/10 mm.

TÓPICOS RELACIONADOS:

- [Cheque e Autenticação](#)

- [Tipos de Parâmetros](#)

1.9.15 CortaPapel

Descrição: Emite o comando para o mecanismo cortar o papel.

Observações: Somente alguns mecanismos possuem este recurso.

Parâmetros

Variavel: TipoCorte

TipoDado: [ubyte](#)

Tamanho Máximo:

Obrigatório

Descricao: Tipo do corte de papel: 0 - corte total; 1 - corte incompleto.

TÓPICOS RELACIONADOS:

- [Mecanismos](#)

- [Tipos de Parâmetros](#)

1.9.16 CruzaCheque

Descrição: Cruza o cheque.

Observações: Permite a impressão de até 3 (três) linhas adicionais paralelas ao cruzamento.

Parâmetros

Variavel: LinhaCruzamento1

TipoDado: [string](#)

Tamanho Máximo: 15

Opcional

Descricao: Texto da primeira linha a ser impressa no cruzamento do cheque.

Variavel: LinhaCruzamento2

TipoDado: [string](#)

Tamanho Máximo: 15

Opcional

Descricao: Texto da segunda linha a ser impressa no cruzamento do cheque.

Variavel: LinhaCruzamento3

TipoDado: [string](#)

Tamanho Máximo: 15

Opcional

Descricao: Texto da terceira linha a ser impressa no cruzamento do cheque.

Variavel: TipoCruzamento

TipoDado: [ubyte](#)

Tamanho Máximo:

Opcional

Descricao: Tipo de cruzamento. Define a relação de impressão entre as linhas e os traços do cruzamento: 0=dois traços, ignorando as linhas; 1=apenas um traço, ignorando as linhas; 2=Linha1, traço, Linha2, traço, Linha3; 3=traço, Linha1, Linha2, Linha3, traço.

TÓPICOS RELACIONADOS:

- [Cheque e Autenticação](#)
- [Tipos de Parâmetros](#)

1.9.17 DefineAliquota

Descrição: Define ou redefine os atributos de uma alíquota programável no ECF. Uma alíquota é definida pelo seu código ou pelo seus atributos percentual e tipo (AliquotaICMS), conjuntamente.

Observações: Possibilita a programação de alíquotas com os mesmos atributos apenas quando definidas pelo código. Sempre que o parâmetro <CodAliquotaProgramavel> for informado no comando, o retorno necessariamente terá o mesmo valor indicado neste parâmetro.

Retornos

Variável: CodAliquotaProgramavel

TipoDado: [byte](#) **Tamanho Máximo:** **Obrigatório**

Descrição: Índice da alíquota entre 0 e NUM_ALIQUOTAS_PROGRAMAVEIS. Informa o índice definido pela impressora para a alíquota programada.

Parâmetros

Variável: AliquotaICMS

TipoDado: [bool](#) **Tamanho Máximo:** **Opcional**

Descrição: Identifica a alíquota como ICMS ('true') ou ISSQN ('false'). Quando não informada, assume ICMS.

Variável: CodAliquotaProgramavel

TipoDado: [byte](#) **Tamanho Máximo:** **Opcional**

Descrição: Índice da alíquota entre 0 e NUM_ALIQUOTAS_PROGRAMAVEIS. Quando não definido, procura pelo próximo índice de alíquota programável disponível.

Variável: DescricaoAliquota

TipoDado: [string](#) **Tamanho Máximo:** 80 **Opcional**

Descrição: Texto associado a esta alíquota em particular. De livre uso do programa aplicativo.

Variável: PercentualAliquota

TipoDado: [money](#) **Tamanho Máximo:** **Obrigatório**

Descrição: Valor percentual com precisão de 2 casas decimais. Define o percentual da alíquota.

TÓPICOS RELACIONADOS:

- [Situações Tributárias](#)
- [Tipos de Parâmetros](#)

1.9.18 DefineDadosCabecalho

Descrição: Define os dados que são impressos no cabeçalho dos documentos emitidos pelo ECF.

Parâmetros

Variável: AlturaLogotipo

TipoDado: [uint](#) **Tamanho Máximo:** **Opcional**

Descrição: Define a altura do logotipo a ser impresso. A unidade de medida para este campo é de 90 pontos por polegada (90 dpi).

Variável: ECF

TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**

Descrição: Identificação do número seqüencial do ECF.

Variável: LarguraLogotipo

TipoDado: [uint](#) **Tamanho Máximo:** **Opcional**

Descrição: Define a largura do logotipo a ser impresso.

Variável: Logotipo

TipoDado: [string](#) **Tamanho Máximo:** 2025 **Opcional**

Descrição: Definição do logotipo segundo o padrão bitmap (.bmp), transmitido no formato ASCII-HEXA.

Variável: Loja

TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**

Descrição: Identificação da loja.

Variável: TextoCabecalho

TipoDado: [string](#)

Tamanho Máximo: 255

Obrigatório

Descrição: Texto do cabeçalho ou clichê. O caracter ASCII 10 ('\n' ou line feed) define uma quebra de linha. Máximo 3 (três) linhas. Se não houver o indicador de quebra de linha, o conteúdo deste parâmetro será truncado e impresso em apenas 1 (uma) linha.

TÓPICOS RELACIONADOS:

- [Dados do Clichê](#)

- [Tipos de Parâmetros](#)

1.9.19 DefineDepartamento

Descrição: Associa um nome de Departamento a um totalizador específico gerenciado pelo ECF. O totalizador de departamento é acrescido no comando de Venda de Item. Um Departamento pode ser definido pelo seu código ou pelo seu nome.

Observações: Os valores totalizados podem ser lidos do ECF p/ uso do aplicativo além de poderem ser usados na impressão de Relatórios Gerenciais. Sempre que o parâmetro <CodDepartamento> for informado, o retorno necessariamente terá o mesmo valor indicado no parâmetro

Retornos

Variável: CodDepartamento

TipoDado: [byte](#)

Tamanho Máximo:

Obrigatório

Descrição: Índice do departamento entre 1 e NUM_DEPARTAMENTOS. O Departamento 0 (zero) é pré-programado como "Combustível". Retorna o índice do totalizador de Departamento programado. O valor retornado neste campo deve necessariamente coincidir com o valor passado no parâmetro de mesmo nome, quando informado na chamada do comando.

Parâmetros

Variável: CodDepartamento

TipoDado: [byte](#)

Tamanho Máximo:

Opcional

Descrição: Índice do departamento entre 1 e NUM_DEPARTAMENTOS. O Departamento 0 (zero) é pré-programado como "Combustível". Quando não definido, procura pelo próximo código de departamento disponível (não programado).

Variável: DescricaoDepartamento

TipoDado: [string](#)

Tamanho Máximo: 80

Opcional

Descrição: Texto associado a este departamento em particular. De livre uso do programa aplicativo.

Variável: NomeDepartamento

TipoDado: [string](#)

Tamanho Máximo: 15

Opcional

Descrição: Nome do departamento. Exemplo: Padaria, Açougue, Têxtil, etc.

TÓPICOS RELACIONADOS:

- [Departamentos](#)

- [Tipos de Parâmetros](#)

1.9.20 DefineGerencial

Descrição: Define ou redefine um Relatório Gerencial. Um Relatório Gerencial pode ser definido pelo código ou pelo nome.

Observações: Possibilita a definição de mais de um Relatório Gerencial com o mesmo nome apenas quando definido pelo código. Sempre que o parâmetro <CodGerencial> for informado, o retorno necessariamente terá o mesmo valor indicado neste parâmetro.

Retornos

Variável: CodGerencial

TipoDado: [ubyte](#)

Tamanho Máximo:

Obrigatório

Descrição: Índice do relatório gerencial entre 0 e NUM_GERENCIAIS. Retorna o índice do Relatório Gerencial programado. O valor retornado neste campo deve necessariamente coincidir com o valor passado no parâmetro de mesmo nome, quando informado na chamada do comando.

Parâmetros**Variável:** CodGerencial**TipoDado:** [ubyte](#)**Tamanho Máximo:****Opcional****Descrição:** Índice do relatório gerencial entre 0 e NUM_GERENCIAIS. Identifica o Relatório Gerencial a ser definido ou redefinido. Quando não informado, procura pelo próximo índice disponível.**Variável:** DescricaoGerencial**TipoDado:** [string](#)**Tamanho Máximo:** 80**Opcional****Descrição:** Texto associado a este relatório gerencial em particular. De livre uso do programa aplicativo.**Variável:** NomeGerencial**TipoDado:** [string](#)**Tamanho Máximo:** 30**Opcional****Descrição:** Nome do Relatório Gerencial. Identifica o nome do Relatório Gerencial sempre que o parâmetro <CodGerencial> não for informado.**TÓPICOS RELACIONADOS:**

- [Gerenciais](#)
- [Tipos de Parâmetros](#)

1.9.21 DefineMeioPagamento**Descrição:** Define ou redefine os atributos de um Meio de Pagamento. Um Meio de Pagamento pode ser definido pelo código ou pelo nome.**Observações:** Possibilita a definição de mais de um Meio de Pagamento com os mesmos atributos apenas quando definidos pelo código. Sempre que o parâmetro <CodMeioPagamentoProgram> for informado, o retorno necessariamente terá o mesmo valor indicado neste parâmetro.**Retornos****Variável:** CodMeioPagamentoProgram**TipoDado:** [byte](#)**Tamanho Máximo:****Obrigatório****Descrição:** Índice do Meio de Pagamento programável entre 0 (zero) e NUM_MEIOS_PAGAMENTO. Retorna o índice do meio de pagamento programado.**Parâmetros****Variável:** CodMeioPagamentoProgram**TipoDado:** [byte](#)**Tamanho Máximo:****Opcional****Descrição:** Índice do Meio de Pagamento programável entre 0 (zero) e NUM_MEIOS_PAGAMENTO. Identifica o Meio de Pagamento a ser definido ou redefinido. Quando não informado, procura pelo próximo índice disponível.**Variável:** DescricaoMeioPagamento**TipoDado:** [string](#)**Tamanho Máximo:****80 Opcional****Descrição:** Texto associado a este Meio de Pagamento em particular. De livre uso do programa aplicativo.**Variável:** NomeMeioPagamento**TipoDado:** [string](#)**Tamanho Máximo:****16 Opcional****Descrição:** Nome do Meio de Pagamento. Identifica o nome do Meio de Pagamento sempre que o parâmetro <CodMeioPagamentoProgram> não for informado.**Variável:** PermiteVinculado**TipoDado:** [bool](#)**Tamanho Máximo:****Opcional****Descrição:** Informa se o Meio de Pagamento habilita ou não a emissão de CCD (Comprovante de Crédito ou Débito). Se este campo não for informado neste comando, seu valor será definido após a primeira utilização do respectivo Meio de Pagamento.**TÓPICOS RELACIONADOS:**

- [Meios de pagamentos](#)
- [Tipos de Parâmetros](#)

1.9.22 DefineNaoFiscal

Descrição: Define ou redefine um totalizador não-fiscal. Um totalizador não fiscal pode ser definido pelo código ou pelo nome.

Observações: Possibilita a definição de mais de um totalizador não fiscal com os mesmos atributos apenas quando definidos pelo código. Sempre que o parâmetro <CodNaoFiscalProgramavel> for informado, o retorno necessariamente terá o mesmo valor indicado neste parâmetro

Retornos

Variável: CodNaoFiscal

TipoDado: [byte](#)

Tamanho Máximo:

Obrigatório

Descrição: Índice do totalizador não fiscal, entre 0 e NUM_NAO_FISCAIS. Indica o código (índice) do totalizador não fiscal.

Parâmetros

Variável: CodNaoFiscal

TipoDado: [byte](#)

Tamanho Máximo:

Opcional

Descrição: Índice do totalizador não fiscal, entre 0 e NUM_NAO_FISCAIS. Identifica o não vinculado a ser definido ou redefinido. Quando este parâmetro não for informado, procura pelo próximo índice disponível.

Variável: DescricaoNaoFiscal

TipoDado: [string](#)

Tamanho Máximo: 80

Opcional

Descrição: Texto associado a este não fiscal em particular. De livre uso do programa aplicativo.

Variável: NomeNaoFiscal

TipoDado: [string](#)

Tamanho Máximo: 19

Opcional

Descrição: Nome do totalizador não fiscal. Identifica o nome do não fiscal sempre que o parâmetro <CodNaoFiscalProgramavel> não for informado.

Variável: TipoNaoFiscal

TipoDado: [bool](#)

Tamanho Máximo:

Opcional

Descrição: O tipo do totalizador não fiscal pode ser de entrada (= 'true' - totalizador de recebimentos ou entrada no caixa) ou de saída (= 'false' - totalizador de sangrias ou saída de caixa). Quando não definido, assume 'true' como valor padrão.

TÓPICOS RELACIONADOS:

- [Totalizadores Não Fiscais](#)
- [Tipos de Parâmetros](#)

1.9.23 DefineProprietario

Descrição: Define os dados de identificação do proprietário.

Parâmetros

Variável: CNPJ

TipoDado: [string](#)

Tamanho Máximo: 20

Obrigatório

Descrição: Identificação do CNPJ.

Variável: InscricaoEstadual

TipoDado: [string](#)

Tamanho Máximo: 20

Obrigatório

Descrição: Identificação na Inscrição Estadual.

Variável: InscricaoMunicipal

TipoDado: [string](#)

Tamanho Máximo: 20

Opcional

Descrição: Identificação na Inscrição Municipal.

Variável: Senha

TipoDado: [ulong](#)

Tamanho Máximo: 0

Obrigatório

Descrição: Senha de habilitação do proprietário.

TÓPICOS RELACIONADOS:

- [Dados de Identificação do Proprietário](#)

- [Tipos de Parâmetros](#)

1.9.24 EmiteltemNaoFiscal

Descrição: Emite valor em totalizador não fiscal. É necessária a identificação do totalizador não fiscal através do código ou do nome.

Observações: Se não houver cupom aberto, inicia automaticamente o Cupom Não-Fiscal. Cupons Não-Fiscais de saída e de entrada não podem ser aceitos no mesmo cupom não fiscal. Não fiscais de saída não devem aceitar pagamento.

Parâmetros

Variável: CodNaoFiscal

TipoDado: [byte](#) **Tamanho Máximo:** **Opcional**

Descrição: Índice do totalizador não fiscal, entre 0 e NUM_NAO_FISCAIS. Este parâmetro tem precedência sobre o parâmetro <NomeNaoFiscal>.

Variável: NomeNaoFiscal

TipoDado: [string](#) **Tamanho Máximo:** 19 **Opcional**

Descrição: Nome do totalizador não fiscal.

Variável: Valor

TipoDado: [money](#) **Tamanho Máximo:** **Obrigatório**

Descrição: Valor da operação.

TÓPICOS RELACIONADOS:

- [Comprovante Não Fiscal](#)
- [Totalizadores Não Fiscais](#)
- [Tipos de Parâmetros](#)

1.9.25 EmiteLeituraFitaDetalhe

Descrição: Emite uma leitura da Memória de Fita Detalhe Eletrônica impressa ou via serial.

Parâmetros

Variável: COOFinal

TipoDado: [ulong](#) **Tamanho Máximo:** **Opcional**

Descrição: Contador de Ordem de Operação final. COO final a ser considerado na emissão da Memória de Fita Detalhe Eletrônica.

Variável: COOInicial

TipoDado: [ulong](#) **Tamanho Máximo:** **Opcional**

Descrição: Contador de Ordem de Operação inicial. COO inicial a ser considerado na emissão da Memória de Fita Detalhe Eletrônica.

Variável: DataFinal

TipoDado: [data](#) **Tamanho Máximo:** **Opcional**

Descrição: Data final a ser considerada na emissão da Fita Detalhe eletrônica.

Variável: DataInicial

TipoDado: [data](#) **Tamanho Máximo:** **Opcional**

Descrição: Data inicial a ser considerada na emissão da Fita Detalhe eletrônica.

Variável: Destino

TipoDado: [string](#) **Tamanho Máximo:** 1 **Opcional**

Descrição: Destino do retorno do comando: 'I' - Impressora; 'S' - Recepção Serial. O valor padrão é 'I'.

Variável: Proprietario

TipoDado: [ubyte](#) **Tamanho Máximo:** **Opcional**

Descrição: Número do proprietário. Quando não informado, emite a leitura de fita detalhe do proprietário atual.

Variável: TipoDocumento

TipoDado: [string](#) **Tamanho Máximo:** 1 **Opcional**

Descrição: Identificação do tipo de documento de acordo com tabela anexa. É válido apenas quando o parâmetro <Destino> = 'S'. Emite a leitura da fita detalhe apenas dos documentos que satisfazem ao parâmetro

<TipoDocumento>.

TÓPICOS RELACIONADOS:

- [Memória de fita detalhe](#)
- [Modo Intervenção Técnica](#)
- [Outros Documentos](#)
- [Tipos de Parâmetros](#)

1.9.26 EmiteLeituraMF

Descrição: Emite uma leitura da Memória Fiscal impressa ou via serial.

Observações: O uso de um dos parâmetros <ReducaoInicial> ou <ReducaoFinal> excluem os parâmetros <DataInicial> e <DataFinal>. Não informando os parâmetros obrigatórios, emite uma leitura completa.

Parâmetros

Variável: DataFinal

TipoDado: [data](#) **Tamanho Máximo:** **Opcional**

Descricao: Data desejada para encerrar a emissão da leitura da Memória Fiscal.

Variável: DataInicial

TipoDado: [data](#) **Tamanho Máximo:** **Opcional**

Descricao: Data desejada para iniciar a emissão da leitura da Memória Fiscal.

Variável: Destino

TipoDado: [string](#) **Tamanho Máximo:** 1 **Opcional**

Descricao: Destino do retorno do comando: 'I' - Impressora; 'S' - Recepção Serial.

Variável: LeituraSimplificada

TipoDado: [bool](#) **Tamanho Máximo:** 0 **Obrigatório**

Descricao: Indicador de leitura simplificado, quando setado.

Variável: Operador

TipoDado: [string](#) **Tamanho Máximo:** 8 **Opcional**

Descricao: Identificação do operador.

Variável: ReducaoFinal

TipoDado: [uint](#) **Tamanho Máximo:** **Opcional**

Descricao: Redução desejada para encerrar a emissão da leitura da Memória Fiscal.

Variável: ReducaoInicial

TipoDado: [uint](#) **Tamanho Máximo:** **Opcional**

Descricao: Redução desejada para iniciar a emissão da leitura da Memória Fiscal.

Exemplo:

```

DLLG2_LimpaParams(hPorta);
DLLG2_AdicionaParam (Porta, 'DataFinal', '01/11/2006', 2);
DLLG2_AdicionaParam (Porta, 'DataInicial', '03/11/2006', 2);
DLLG2_AdicionaParam (hPorta, 'Destino', 'I', 7);
DLLG2_AdicionaParam (hPorta, 'LeituraSimplificada', 'F', 0);
DLLG2_ExecutaComando(hPorta, 'EmiteLeituraMF');

```

TÓPICOS RELACIONADOS:

- [Modo Intervenção Técnica](#)
- [Relatórios Fiscais](#)
- [Tipos de Parâmetros](#)

1.9.27 EmiteLeituraX

Descrição: Emite uma leitura X impressa ou via serial.

Parâmetros

Variável: Destino

TipoDado: [string](#) **Tamanho Máximo:** 1 **Opcional**

Descricao: Destino do retorno do comando: 'I' - Impressora; 'S' - Recepção Serial.

Variavel: ImprimeBitmap

TipoDado: [bool](#)

Tamanho Máximo:

Opcional

Descricao: Quando setada (= 'true'), imprime o 'bitmap' neste documento. Se não informado, não será impresso.

Variavel: Operador

TipoDado: [string](#)

Tamanho Máximo: 8

Opcional

Descricao: Identificação do operador.

Exemplo de utilização do comando:

```
//Adiciona parâmetros
DLLG2_AdicionaParam(Porta, 'Operador', Elgin, 7);

//Executa o comando
If DLLG2_ExecutaComando(Porta, 'EmiteLeituraX') > 0 Then
  NomeErro := DLLG2_ObtemNomeErro(pchar(Porta), NomeErro, 0);
  CodErro := DLLG2_ObtemCodErro(Porta);
  Circunstancia := DLLG2_ObtemCircunstancia(Porta, Circunstancia, 200);
  Txt_Erro.Text := NomeErro;
  Txt_Circunstancia.Text := Circunstancia;
end;
```

TÓPICOS RELACIONADOS:

- [Relatórios Fiscais](#)
- [Modo Intervenção Técnica](#)
- [Tipos de Parâmetros](#)

1.9.28 EmiteReducaoZ

Descrição: Emite a Redução Z - relatório de fechamento fiscal diário, permitindo opcionalmente o ajuste do relógio com tolerância de 5 minutos a maior ou a menor com relação ao horário atual do relógio do ECF.

Parâmetros

Variavel: Hora

TipoDado: [hora](#)

Tamanho Máximo:

Opcional

Descricao: Nova hora a ser ajustada dentro do intervalo permitido.

Variavel: Operador

TipoDado: [string](#)

Tamanho Máximo: 8

Opcional

Descricao: Identificação do operador.

TÓPICOS RELACIONADOS:

- [Relatórios Fiscais](#)
- [Modo Intervenção Técnica](#)
- [Tipos de Parâmetros](#)

1.9.29 EmiteViaCreditoDebito

Descrição: Emite uma nova via do CCD previamente emitido.

Observações: Este comando deve suceder seqüencialmente a emissão do CCD desejado para replicação de via.

TÓPICOS RELACIONADOS:

- [Comprovante de Crédito ou Débito](#)
- [Relatórios Fiscais](#)
- [Tipos de Parâmetros](#)

1.9.30 EncerraDocumento

Descrição: Encerra o documento aberto com opção de imprimir a mensagem promocional, quando possível.

Observações: A mensagem promocional será desconsiderada quando não aplicada ao documento aberto.

Parâmetros**Variavel:** Operador**TipoDado:** [string](#)**Tamanho Máximo:** 8**Opcional****Descricao:** Identificação do operador.**Variavel:** TextoPromocional**TipoDado:** [string](#)**Tamanho Máximo:** 492**Opcional****Descricao:** Texto da mensagem promocional a ser impressa. O caracter ASCII 10 ('\n' ou line feed) é interpretado como separador de linhas do texto promocional.**TÓPICOS RELACIONADOS:**

- [Cupom Fiscal](#)
- [Comprovante Não Fiscal](#)
- [Comprovante de Crédito ou Débito](#)
- [Relatório Gerencial](#)
- [Tipos de Parâmetros](#)

1.9.31 EscreveData**Descrição:** Escreve um valor do tipo 'data' em um registrador.**Parâmetros****Variavel:** NomeData**TipoDado:** [string](#)**Tamanho Máximo:****50 Obrigatório****Descricao:** Nome da variável (registrador) do tipo 'data' desejada. Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '[']' após o nome.**Variavel:** ValorData**TipoDado:** [data](#)**Tamanho Máximo:****Obrigatório****Descricao:** Valor da data referenciada.**TÓPICOS RELACIONADOS:**

- [Tipos de Parâmetros](#)

1.9.32 EscreveHora**Descrição:** Escreve um valor do tipo 'hora' em um registrador.**Parâmetros****Variavel:** NomeHora**TipoDado:** [string](#)**Tamanho Máximo:** 50**Obrigatório****Descricao:** Nome da variável (registrador) do tipo 'hora' desejada. Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '[']' após o nome.**Variavel:** ValorHora**TipoDado:** [hora](#)**Tamanho Máximo:****Obrigatório****Descricao:** Valor da hora solicitada.**TÓPICOS RELACIONADOS:**

- [Relógio](#)
- [Tipos de Parâmetros](#)

1.9.33 EscreveIndicador**Descrição:** Este comando permite escrever até 16 indicadores ('flags').**Observações:** Para cada nome de indicador pode ser associado um valor. Os valores podem ser indicados via string de caracteres (seqüência de "0"s e "1"s) ou via um valor inteiro (forma binária). Pelo menos um campo de valor deve ser informado no comando.**Parâmetros**

Variavel: NomeIndicador

TipoDado: [string](#)

Tamanho Máximo: 2000

Obrigatório

Descricao: Nome da variável (registrador) tipo indicador desejada. Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '['' após o nome.

Variavel: ValorNumericoIndicador

TipoDado: [long](#)

Tamanho Máximo:

Opcional

Descricao: Representação binária do valor referente a um ou mais indicadores (máximo 16), sendo 1 para ligado e 0 para desligado.

Variavel: ValorTextoIndicador

TipoDado: [string](#)

Tamanho Máximo: 16

Opcional

Descricao: Representação em um string de caracteres do valor referente a um ou mais indicadores (máximo 16), sendo "1" para ligado e "0" para desligado. Tem precedência sobre

TÓPICOS RELACIONADOS:

- [Indicadores](#)
- [Tipos de Parâmetros](#)

1.9.34 EscreveInteiro

Descrição: Escreve um valor do tipo 'inteiro' em um registrador.

Parâmetros

Variavel: NomeInteiro

TipoDado: [string](#)

Tamanho Máximo: 50

Obrigatório

Descricao: Nome da variável (registrador) inteira solicitada. Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '['' após o nome.

Variavel: ValorInteiro

TipoDado: [long](#)

Tamanho Máximo:

Obrigatório

Descricao: Valor inteiro.

TÓPICOS RELACIONADOS:

- [Tipos de Parâmetros](#)

1.9.35 EscreveMoeda

Descrição: Escreve um valor do tipo 'moeda' em um registrador.

Parâmetros

Variavel: NomeDadoMonetario

TipoDado: [string](#)

Tamanho Máximo: 50

Obrigatório

Descricao: Nome do valor monetário (registrador). Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '['' após o nome.

Variavel: ValorMonetario

TipoDado: [money](#)

Tamanho Máximo:

Obrigatório

Descricao: Valor monetário.

TÓPICOS RELACIONADOS:

- [Tipos de Parâmetros](#)

1.9.36 EscreveTexto

Descrição: Escreve um valor do tipo 'texto' em um registrador.

Parâmetros

Variavel: NomeTexto

TipoDado: [string](#)

Tamanho Máximo: 50

Obrigatório

Descricao: Nome da variável de texto (registrador) desejada. Lista de registradores presente em um dos

anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '[']' após o nome.

Variável: ValorTexto
TipoDado: string **Tamanho Máximo:** 4000 **Obrigatório**
Descricao: Valor da variável de texto solicitada.

1.9.37 EstornaCreditoDebito

Descrição: Estorna um Comprovante de Crédito ou Débito (CCD) informando o COO do CCD.

Parâmetros

Variável: COO
TipoDado: ulong **Tamanho Máximo:** **Obrigatório**
Descricao: Contador de Ordem de Operação.

TÓPICOS RELACIONADOS:

- [Comprovante de Crédito ou Débito](#)
- [Tipos de Parâmetros](#)

1.9.38 EstornaMeioPagamento

Descrição: Transfere valores entre 2 (dois) Meios de Pagamento, sendo que o Meio de Pagamento origem deve ter sido usado no último cupom emitido.

Observações: Ambos os Meios de Pagamento devem estar definidos.

Parâmetros

Variável: CodMeioPagamentoDestino
TipoDado: byte **Tamanho Máximo:** **Opcional**
Descricao: Índice do Meio de Pagamento destino da operação (-2 ou um valor entre zero e NUM_MEIOS_PAGAMENTO) na qual o valor será creditado. Este parâmetro tem precedência sobre o parâmetro <NomeMeioPagamentoDest>.

Variável: CodMeioPagamentoOrigem
TipoDado: byte **Tamanho Máximo:** **Opcional**
Descricao: Índice do Meio de Pagamento, sendo -2 ou um valor entre zero e NUM_MEIOS_PAGAMENTO onde o valor será estornado (debitado). Este parâmetro tem precedência sobre o parâmetro <NomeMeioPagamentoOrig>.

Variável: NomeMeioPagamentoDest
TipoDado: string **Tamanho Máximo:** 16 **Opcional**
Descricao: Nome do Meio de Pagamento no qual o valor será creditado.

Variável: NomeMeioPagamentoOrig
TipoDado: string **Tamanho Máximo:** 16 **Opcional**
Descricao: Nome do Meio de Pagamento que deve ter seu valor estornado (debitado).

Variável: Operador
TipoDado: string **Tamanho Máximo:** 8 **Opcional**
Descricao: Identificação do operador.

Variável: Valor
TipoDado: money **Tamanho Máximo:** **Obrigatório**
Descricao: Valor da operação.

TÓPICOS RELACIONADOS:

- [Outros Documentos](#)
- [Tipos de Parâmetros](#)

1.9.39 ExcluiAliquota

Descrição: Exclui uma alíquota (Situação Tributária) previamente programada.

Observações: A alíquota a ser excluída deve ser necessariamente identificada ou pelo seu código ou por seu percentual e tipo.

Parâmetros

Variável: AliquotaICMS

TipoDado: [bool](#)

Tamanho Máximo:

Opcional

Descrição: Identifica a alíquota como ICMS ('true') ou ISSQN ('false').

Variável: CodAliquotaProgramavel

TipoDado: [byte](#)

Tamanho Máximo:

Opcional

Descrição: Índice da alíquota entre 0 e NUM_ALIQUOTAS_PROGRAMAVEIS. Identifica a alíquota a ser excluída. Este parâmetro tem precedência sobre os parâmetros <PercentualAliquota> e <AliquotaICMS>.

Variável: PercentualAliquota

TipoDado: [money](#)

Tamanho Máximo:

Opcional

Descrição: Valor percentual com precisão de 2 casas decimais. Define o percentual da alíquota a ser excluída, juntamente com o parâmetro <AliquotaICMS>.

TÓPICOS RELACIONADOS:

- [Situações Tributárias](#)
- [Tipos de Parâmetros](#)

1.9.40 ExcluiDepartamento

Descrição: Exclui um Departamento previamente cadastrado.

Observações: O Departamento deve ser necessariamente identificado pelo seu código ou pelo seu nome para a exclusão ser executada.

Parâmetros

Variável: CodDepartamento

TipoDado: [byte](#)

Tamanho Máximo:

Opcional

Descrição: Índice do departamento entre 1 e NUM_DEPARTAMENTOS. O Departamento 0 (zero) é pré-programado como "Combustível". Este parâmetro tem precedência sobre

Variável: NomeDepartamento

TipoDado: [string](#)

Tamanho Máximo: 15

Opcional

Descrição: Nome do departamento. Exemplo: Padaria, Açougue, Têxtil, etc.

TÓPICOS RELACIONADOS:

- [Departamentos](#)
- [Tipos de Parâmetros](#)

1.9.41 ExcluiGerencial

Descrição: Exclui um Relatório Gerencial previamente cadastrado.

Observações: O Relatório Gerencial deve ser necessariamente identificado pelo seu código ou pelo seu nome para

exclusão ser executada.

Parâmetros

Variável: CodGerencial

TipoDado: [ubyte](#)

Tamanho Máximo:

Opcional

Descrição: Índice do relatório gerencial entre 0 e NUM_GERENCIAIS. Este parâmetro tem precedência com relação ao parâmetro <NomeGerencial>.

Variável: NomeGerencial

TipoDado: [string](#)

Tamanho Máximo: 30

Opcional

Descrição: Nome do Relatório Gerencial.

TÓPICOS RELACIONADOS:

- [Gerenciais](#)
- [Tipos de Parâmetros](#)

1.9.42 ExcluiMeioPagamento

Descrição: Exclui um Meio de Pagamento previamente cadastrado.

Observações: O Meio de Pagamento deve ser necessariamente identificado pelo seu código ou pelo seu nome para exclusão ser executada.

Parâmetros

Variavel: CodMeioPagamentoProgram

TipoDado: [byte](#)

Tamanho Máximo:

Opcional

Descricao: Índice do Meio de Pagamento programável entre 0 (zero) e NUM_MEIOS_PAGAMENTO. Este parâmetro tem precedência sobre <NomeMeioPagamento>.

Variavel: NomeMeioPagamento

TipoDado: [string](#)

Tamanho Máximo: 16

Opcional

Descricao: Nome do Meio de Pagamento.

TÓPICOS RELACIONADOS:

- [Meios de pagamentos](#)
- [Tipos de Parâmetros](#)

1.9.43 ExcluiNaoFiscal

Descrição: Exclui um totalizador não fiscal previamente definido.

Observações: O totalizador não fiscal deve ser necessariamente identificado pelo seu código ou pelo seu nome para exclusão ser executada.

Parâmetros

Variavel: CodNaoFiscal

TipoDado: [byte](#)

Tamanho Máximo:

Opcional

Descricao: Índice do totalizador não fiscal, entre 0 e NUM_NAO_FISCAIS. Este parâmetro tem precedência sobre <NomeNaoFiscal>.

Variavel: NomeNaoFiscal

TipoDado: [string](#)

Tamanho Máximo: 19

Opcional

Descricao: Nome do totalizador não fiscal.

TÓPICOS RELACIONADOS:

- [Totalizadores Não Fiscais](#)
- [Tipos de Parâmetros](#)

1.9.44 ExpulsaDocumento

Descrição: Expulsa o documento do mecanismo impressor secundário (mecanismo de impressão de cheques, autenticações, etc).

TÓPICOS RELACIONADOS:

- [Mecanismos](#)
- [Cheque e Autenticação](#)
- [Tipos de Parâmetros](#)

1.9.45 IdentificaConsumidor

Descrição: Imprime os dados de identificação do consumidor em um cupom.

Parâmetros

Variavel: EnderecoConsumidor

TipoDado: [string](#)

Tamanho Máximo: 80

Opcional

Descricao: Endereço do consumidor.

Variavel: IdConsumidor

TipoDado: [string](#) **Tamanho Máximo:** 29 **Opcional**
Descricao: Identificação do consumidor.

Variavel: NomeConsumidor
TipoDado: [string](#) **Tamanho Máximo:** 30 **Opcional**
Descricao: Nome do consumidor.

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)
- [Tipos de Parâmetros](#)

1.9.46 ImprimeAutenticacao

Descrição: Autentica o documento.

Observações: Possibilita a impressão de um texto adicional. Se o documento não for posicionado para autenticação após tempo definido pelo parâmetro <TempoEspera>, cancela ação do comando.

Parâmetros

Variavel: CaracterAutenticacao
TipoDado: [string](#) **Tamanho Máximo:** 36 **Opcional**
Descricao: Define o caracter de autenticação considerando uma matriz de 8 linhas por 18 colunas. O formato de transmissão deve ser ASCII HEXA.

Variavel: SentidoInsercao
TipoDado: [uint](#) **Tamanho Máximo:** 0 **Opcional**
Descricao: Sentido de inserção do documento para autenticação: 0 - inferior, 1 - lateral, 2 - superior.

Variavel: TempoEspera
TipoDado: [ubyte](#) **Tamanho Máximo:** **Opcional**
Descricao: Tempo de espera, em segundos, pela ação do comando. Caso não informado, assume como padrão o te.po de 60 segundos.

Variavel: TextoAutenticacao
TipoDado: [string](#) **Tamanho Máximo:** 50 **Opcional**
Descricao: Texto complementar usado para autenticação de documentos.

TÓPICOS RELACIONADOS:

- [Cheque e Autenticação](#)
- [Tipos de Parâmetros](#)

1.9.47 ImprimeCheque

Descrição: Envia o comando de impressão de cheque para a impressora.

Observações: Se o cheque não for posicionado para impressão após tempo definido pelo parâmetro <TempoEspera>, cancela ação do comando.

Parâmetros

Variavel: Cidade
TipoDado: [string](#) **Tamanho Máximo:** 27 **Obrigatório**
Descricao: Cidade a ser impressa no documento.

Variavel: Data
TipoDado: [data](#) **Tamanho Máximo:** **Obrigatório**
Descricao: Data para preenchimento do cheque.

Variavel: Favorecido
TipoDado: [string](#) **Tamanho Máximo:** 45 **Obrigatório**
Descricao: Identificação do favorecido ou beneficiário do cheque.

Variavel: HPosAno
TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**
Descricao: Distância horizontal entre a margem esquerda do documento e o início do caracter, em 1/10 mm, para impressão do campo Ano no documento.A posição vertical usada para este campo é a VPosCidade.

- Variável:** HPosCidade
TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão do campo Cidade no documento.
- Variável:** HPosDia
TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão do campo Dia no documento. A posição vertical usada para este campo é VPosCidade.
- Variável:** HPosExtensoLinha1
TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão da primeira linha do Extenso, gerado pelo ECF, referente ao valor do cheque.
- Variável:** HPosExtensoLinha2
TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão da segunda linha do Extenso, gerado pelo ECF, referente ao valor do cheque.
- Variável:** HPosFavorecido
TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão do campo Favorecido no documento.
- Variável:** HPosMes
TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão do campo Mes no documento. A posição vertical usada para este campo é VPosCidade.
- Variável:** HPosMsgLinha1
TipoDado: [uint](#) **Tamanho Máximo:** **Opcional**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão do campo MensagemDocLinha1 no documento.
- Variável:** HPosMsgLinha2
TipoDado: [uint](#) **Tamanho Máximo:** **Opcional**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão do campo MensagemDocLinha2 no documento.
- Variável:** HPosMsgLinha3
TipoDado: [uint](#) **Tamanho Máximo:** **Opcional**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão do campo MensagemDocLinha3 no documento.
- Variável:** HPosValor
TipoDado: [uint](#) **Tamanho Máximo:** **Obrigatório**
Descrição: Distância horizontal entre a margem esquerda do documento e o início do carácter, em 1/10 mm, para impressão do campo Valor no documento.
- Variável:** MensagemDocLinha1
TipoDado: [string](#) **Tamanho Máximo:** 80 **Opcional**
Descrição: Linha 1 do texto livre adicional a ser impresso no documento.
- Variável:** MensagemDocLinha2
TipoDado: [string](#) **Tamanho Máximo:** 80 **Opcional**
Descrição: Linha 2 do texto livre adicional a ser impresso no documento.
- Variável:** MensagemDocLinha3
TipoDado: [string](#) **Tamanho Máximo:** 80 **Opcional**
Descrição: Linha 3 do texto livre adicional a ser impresso no documento.
- Variável:** TempoEspera
TipoDado: [ubyte](#) **Tamanho Máximo:** **Opcional**

Descricao: Tempo de espera, em segundos, pela ação do comando. Caso não informado, assume como padrão o tempo de 60 segundos.

Variavel: Valor

TipoDado: [money](#)

Tamanho Máximo:

Obrigatório

Descricao: Valor da operação.

Variavel: VPosCidade

TipoDado: [uint](#)

Tamanho Máximo:

Obrigatório

Descricao: Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo Cidade no documento.

Variavel: VPosExtensoLinha1

TipoDado: [uint](#)

Tamanho Máximo:

Obrigatório

Descricao: Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão da primeira linha do Extenso no documento.

Variavel: VPosExtensoLinha2

TipoDado: [uint](#)

Tamanho Máximo:

Obrigatório

Descricao: Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão da segunda linha do Extenso no documento.

Variavel: VPosFavorecido

TipoDado: [uint](#)

Tamanho Máximo:

Obrigatório

Descricao: Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo Favorecido no documento.

Variavel: VPosMsgLinha1

TipoDado: [uint](#)

Tamanho Máximo:

Opcional

Descricao: Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo MensagemDocLinha1 no documento.

Variavel: VPosMsgLinha2

TipoDado: [uint](#)

Tamanho Máximo:

Opcional

Descricao: Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo MensagemDocLinha2 no documento.

Variavel: VPosMsgLinha3

TipoDado: [uint](#)

Tamanho Máximo:

Opcional

Descricao: Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo MensagemDocLinha3 no documento.

Variavel: VPosValor

TipoDado: [uint](#)

Tamanho Máximo:

Obrigatório

Descricao: Distância vertical entre o topo do documento e a base do caracter, em 1/10 mm, para impressão do campo Valor no documento.

TÓPICOS RELACIONADOS:

- [Cheque e Autenticação](#)
- [Tipos de Parâmetros](#)

1.9.48 ImprimeCupomAdicional

Descrição: Imprime o cupom adicional.

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)
- [Tipos de Parâmetros](#)

1.9.49 ImprimeDadosDepartamento

Descrição: Imprime os dados referente aos Departamentos definidos em um Relatório Gerencial.

Observações: O Relatório Gerencial deve estar aberto.

TÓPICOS RELACIONADOS:

- [Departamentos](#)
- [Tipos de Parâmetros](#)

1.9.50 ImprimeTexto

Descrição: Imprime texto em um documento.

Observações: Este comando é usado para impressão de texto livre em relatórios gerenciais, comprovantes de crédito ou débito, Cupons Fiscais, etc.

Parâmetros

Variável: TextoLivre

TipoDado: [string](#)

Tamanho Máximo: 492

Obrigatório

Descrição: Conteúdo do texto livre. Podem ser usados caracteres modificadores de impressão para gerar impressão negrito, expandido, código de barras, etc.

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)
- [Comprovante de Crédito ou Débito](#)
- [Relatório Gerencial](#)
- [Tipos de Parâmetros](#)

1.9.51 IniciaFitaDetalhe

Descrição: Inicia uma nova Memória de Fita Detalhe (MFD) instalada no equipamento, desde que não seja a primeira.

Parâmetros

Variável: NumSerie

TipoDado: [string](#)

Tamanho Máximo: 10

Obrigatório

Descrição: Número de série.

TÓPICOS RELACIONADOS:

- [Modo Intervenção Técnica](#)
- [Tipos de Parâmetros](#)

1.9.52 LeAliquota

Descrição: Lê os atributos de uma alíquota programável já definida.

Observações: A alíquota a ser lida deve ser informada nos parâmetros pelo seu código ou ainda pelo seu percentual em conjunto com o tipo (ICMS ou ISS). Caso não seja passado nenhum parâmetro, este comando não será executado.

Retornos

Variável: AliquotaICMS

TipoDado: [bool](#)

Tamanho Máximo:

Obrigatório

Descrição: Identifica a alíquota como ICMS ('true') ou ISSQN ('false').

Variável: CodAliquotaProgramavel

TipoDado: [byte](#)

Tamanho Máximo:

Obrigatório

Descrição: Índice da alíquota entre 0 e NUM_ALIQUOTAS_PROGRAMAVEIS.

Variável: DescricaoAliquota

TipoDado: [string](#)

Tamanho Máximo: 80

Obrigatório

Descrição: Texto associado a esta alíquota em particular. De livre uso do programa aplicativo.

Variável: PercentualAliquota

TipoDado: [money](#)

Tamanho Máximo:

Obrigatório

Descrição: Valor percentual com precisão de 2 casas decimais.

Parâmetros

Variável: AliquotaICMS

TipoDado: [bool](#) **Tamanho Máximo:** **Opcional**
Descricao: Identifica a aliquota como ICMS ('true') ou ISSQN ('false'). Quando não informada, assume ICMS.

Variavel: CodAliquotaProgramavel

TipoDado: [byte](#) **Tamanho Máximo:** **Opcional**

Descricao: Índice da alíquota entre 0 e NUM_ALIQUOTAS_PROGRAMAVEIS. Este parâmetro tem precedência sobre <PercentualAliquota> e <AliquotaICMS>.

Variavel: PercentualAliquota

TipoDado: [money](#) **Tamanho Máximo:** **Opcional**

Descricao: Valor percentual com precisão de 2 casas decimais.

TÓPICOS RELACIONADOS:

- [Situções Tributárias](#)
- [Tipos de Parâmetros](#)

1.9.53 LeData

Descricao: Le um registrador do tipo 'data'.

Retornos

Variavel: ValorData

TipoDado: [data](#) **Tamanho Máximo:** **Obrigatório**

Descricao: Valor da data referenciada.

Parâmetros

Variavel: NomeData

TipoDado: [string](#) **Tamanho Máximo:** 50 **Obrigatório**

Descricao: Nome da variável (registrador) do tipo 'data' desejada. Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '[']' após o nome.

TÓPICOS RELACIONADOS:

- [Tipos de Parâmetros](#)

1.9.54 LeDepartamento

Descricao: Le os atributos de um Departamento definido.

Observações: O Departamento a ser lido deve ser identificado pelo seu código ou pelo seu nome.

Retornos

Variavel: CodDepartamento

TipoDado: [byte](#) **Tamanho Máximo:** **Obrigatório**

Descricao: Índice do departamento entre 1 e NUM_DEPARTAMENTOS. O Departamento 0 (zero) é pré-programado como "Combustível".

Variavel: DescricaoDepartamento

TipoDado: [string](#) **Tamanho Máximo:** 80 **Obrigatório**

Descricao: Texto associado a este departamento em particular. De livre uso do programa aplicativo.

Variavel: NomeDepartamento

TipoDado: [string](#) **Tamanho Máximo:** 15 **Obrigatório**

Descricao: Nome do departamento. Exemplo: Padaria, Açougue, Têxtil, etc.

Parâmetros

Variavel: CodDepartamento

TipoDado: [byte](#) **Tamanho Máximo:** **Opcional**

Descricao: Índice do departamento entre 1 e NUM_DEPARTAMENTOS. O Departamento 0 (zero) é pré-programado como "Combustível". Este parâmetro tem precedência sobre

Variavel: NomeDepartamento

TipoDado: [string](#) **Tamanho Máximo:** 15 **Opcional**

Descricao: Nome do departamento. Exemplo: Padaria, Açougue, Têxtil, etc.

TÓPICOS RELACIONADOS:

- [Departamentos](#)
- [Tipos de Parâmetros](#)

1.9.55 LeGerencial

Descrição: Le os atributos de um Relatório Gerencial definido.

Observações: O Relatório Gerencial a ser lido deve ser identificado pelo seu código ou pelo seu nome.

Retornos

Variável: CodGerencial

TipoDado: [ubyte](#)

Tamanho Máximo:

Obrigatório

Descrição: Índice do relatório gerencial entre 0 e NUM_GERENCIAIS.

Variável: DescricaoGerencial

TipoDado: [string](#)

Tamanho Máximo: 80

Obrigatório

Descrição: Texto associado a este relatório gerencial em particular. De livre uso do programa aplicativo.

Variável: NomeGerencial

TipoDado: [string](#)

Tamanho Máximo: 30

Obrigatório

Descrição: Nome do Relatório Gerencial.

Parâmetros

Variável: CodGerencial

TipoDado: [ubyte](#)

Tamanho Máximo:

Opcional

Descrição: Índice do relatório gerencial entre 0 e NUM_GERENCIAIS. Este parâmetro tem precedência sobre <NomeGerencial>.

Variável: NomeGerencial

TipoDado: [string](#)

Tamanho Máximo: 30

Opcional

Descrição: Nome do Relatório Gerencial.

TÓPICOS RELACIONADOS:

- [Gerenciais](#)
- [Tipos de Parâmetros](#)

1.9.56 LeHora

Descrição: Obtém um registrador do tipo 'hora'.

Retornos

Variável: ValorHora

TipoDado: [hora](#)

Tamanho Máximo:

Obrigatório

Descrição: Valor da hora solicitada.

Parâmetros

Variável: NomeHora

TipoDado: [string](#)

Tamanho Máximo: 50

Obrigatório

Descrição: Nome da variável (registrador) do tipo 'hora' desejada. Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '[]' após o nome.

TÓPICOS RELACIONADOS:

- [Tipos de Parâmetros](#)

1.9.57 LeImpressao

Descrição: Obtém o texto referente a última impressão realizada pelo equipamento.

Observações: Refere-se ao último buffer de impressão enviado para o mecanismo impressor.

Retornos

Variável: TextoImpressao

TipoDado: [string](#)

Tamanho Máximo: 4000

Obrigatório

Descrição: Conteúdo do texto de impressão solicitado.

TÓPICOS RELACIONADOS:

- [Leitura do texto impresso](#)
- [Tipos de Parâmetros](#)

1.9.58 LeIndicador

Descrição: Este comando lê até 16 indicadores ('flags'). Os nomes dos indicadores devem estar separados por espaços.

Observações: Um indicador tem somente dois valores possíveis: 0 (desligado) ou 1 (ligado). Exemplo de indicadores: intervenção técnica, modo treinamento, horário de verão entre os demais definidos na tabela de indicadores anexa.

Retornos

Variável: ValorNumericoIndicador

TipoDado: [long](#)

Tamanho Máximo:

Obrigatório

Descrição: Representação binária do valor referente a um ou mais indicadores (máximo 16), sendo 1 para ligado e 0 para desligado. No caso de leitura simultânea de até 16 indicadores, o valor retornado será a composição binária dos resultados organizados por ordem de solicitação, sendo o primeiro indicador solicitado o bit menos significativo do resultado.

Variável: ValorTextoIndicador

TipoDado: [string](#)

Tamanho Máximo: 16

Obrigatório

Descrição: Representação em um string de caracteres do valor referente a um ou mais indicadores (máximo 16), sendo "1" para ligado e "0" para desligado. No caso de leitura simultânea de até 16 indicadores, o valor retornado será um string de zeros e uns dos resultados organizados por ordem de solicitação, sendo o primeiro indicador solicitado o último dígito do resultado.

Parâmetros

Variável: NomeIndicador

TipoDado: [string](#)

Tamanho Máximo: 2000

Obrigatório

Descrição: Nome da variável (registrador) tipo indicador desejada. Lista de registradores presente em um dos anexos. A leitura simultânea de vários indicadores é possível separando-os por espaços. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '['' após o nome.

TÓPICOS RELACIONADOS:

- [Indicadores do Software Básico](#)
- [Indicadores](#)
- [Tipos de Parâmetros](#)

1.9.59 LeInteiro

Descrição: Obtém um registrador do tipo 'inteiro'.

Retornos

Variável: ValorInteiro

TipoDado: [long](#)

Tamanho Máximo:

Obrigatório

Descrição: Valor inteiro.

Parâmetros

Variável: NomeInteiro

TipoDado: [string](#)

Tamanho Máximo: 50

Obrigatório

Descrição: Nome da variável (registrador) inteira solicitada. Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '['' após o nome.

TÓPICOS RELACIONADOS:

- [Tipos de Parâmetros](#)

1.9.60 LeMeioPagamento

Descrição: Le os atributos de um Meio de Pagamento definido.

Observações: O Meio de Pagamento a ser lido deve ser identificado pelo seu código ou pelo seu nome.

Retornos

Variavel: CodMeioPagamentoProgram

TipoDado: [byte](#) **Tamanho Máximo:** **Obrigatório**

Descricao: Índice do Meio de Pagamento programável entre 0 (zero) e NUM_MEIOS_PAGAMENTO.

Variavel: DescricaoMeioPagamento

TipoDado: [string](#) **Tamanho Máximo:** 80 **Obrigatório**

Descricao: Texto associado a este Meio de Pagamento em particular. De livre uso do programa aplicativo.

Variavel: NomeMeioPagamento

TipoDado: [string](#) **Tamanho Máximo:** 16 **Obrigatório**

Descricao: Nome do Meio de Pagamento.

Variavel: PermiteVinculado

TipoDado: [bool](#) **Tamanho Máximo:** **Obrigatório**

Descricao: Informa se o Meio de Pagamento habilita ou não a emissão de CCD (Comprovante de Crédito ou Débito).

Parâmetros

Variavel: CodMeioPagamentoProgram

TipoDado: [byte](#) **Tamanho Máximo:** **Opcional**

Descricao: Índice do Meio de Pagamento programável entre 0 (zero) e NUM_MEIOS_PAGAMENTO. Este parâmetro tem precedência sobre <NomeMeioPagamento>.

Variavel: NomeMeioPagamento

TipoDado: [string](#) **Tamanho Máximo:** 16 **Opcional**

Descricao: Nome do Meio de Pagamento.

TÓPICOS RELACIONADOS:

- [Meios de pagamentos](#)
- [Tipos de Parâmetros](#)

1.9.61 LeMoeda

Descrição: Obtém um registrador do tipo 'moeda'.

Retornos

Variavel: ValorMoeda

TipoDado: [money](#) **Tamanho Máximo:** **Obrigatório**

Descricao: Valor monetario representado em 4 casas decimas

Parâmetros

Variavel: NomeDadoMonetario

TipoDado: [string](#) **Tamanho Máximo:** 50 **Obrigatório**

Descricao: Nome do valor monetário (registrador). Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '[]' após o nome.

TÓPICOS RELACIONADOS:

- [Tipos de Parâmetros](#)

1.9.62 LeNaoFiscal

Descrição: Le os atributos de um totalizador nao fiscal definido.

Observações: O totalizador nao fiscal a ser lido deve ser identificado pelo seu código ou pelo seu nome.

Retornos

Variavel: CodNaoFiscal

TipoDado: [byte](#) **Tamanho Máximo:** **Obrigatório**

Descrição: Índice do totalizador não fiscal, entre 0 e NUM_NAO_FISCAIS.

Variável: DescricaoNaoFiscal

TipoDado: [string](#) **Tamanho Máximo:** 80 **Obrigatório**

Descrição: Texto associado a este não fiscal em particular. De livre uso do programa aplicativo.

Variável: NomeNaoFiscal

TipoDado: [string](#) **Tamanho Máximo:** 19 **Obrigatório**

Descrição: Nome do totalizador não fiscal.

Variável: TipoNaoFiscal

TipoDado: [bool](#) **Tamanho Máximo:** **Obrigatório**

Descrição: O tipo do totalizador não fiscal pode ser de entrada (= 'true' - totalizador de recebimentos ou entrada no caixa) ou de saída (= 'false' - totalizador de sangrias ou saída de caixa).

Parâmetros

Variável: CodNaoFiscal

TipoDado: [byte](#) **Tamanho Máximo:** **Opcional**

Descrição: Índice do totalizador não fiscal, entre 0 e NUM_NAO_FISCAIS. Este parâmetro tem precedência sobre <NomeNaoFiscal>.

Variável: NomeNaoFiscal

TipoDado: [string](#) **Tamanho Máximo:** 19 **Opcional**

Descrição: Nome do totalizador não fiscal.

TÓPICOS RELACIONADOS:

- [Totalizadores Não Fiscais](#)
- [Tipos de Parâmetros](#)

1.9.63 LeTexto

Descrição: Obtém um registrador do tipo 'texto'.

Retornos

Variável: ValorTexto

TipoDado: [string](#) **Tamanho Máximo:** 4000 **Obrigatório**

Descrição: Valor da variável de texto solicitada.

Parâmetros

Variável: NomeTexto

TipoDado: [string](#) **Tamanho Máximo:** 50 **Obrigatório**

Descrição: Nome da variável de texto (registrador) desejada. Lista de registradores presente em um dos anexos. Quando este parâmetro referenciar um vetor, o índice do vetor deve ser identificado entre colchetes '[']' após o nome.

TÓPICOS RELACIONADOS:

- [Cheque e Autenticação](#)
- [Tipos de Parâmetros](#)

1.9.64 PagaCupom

Descrição: Registra um Meio de Pagamento e o valor pago neste Meio de Pagamento em um cupom (transação de pagamento).

Observações: É necessária a identificação do Meio de Pagamento através do código ou do nome.

Parâmetros

Variável: CodMeioPagamento

TipoDado: [byte](#) **Tamanho Máximo:** **Opcional**

Descrição: Índice do Meio de Pagamento, sendo -2 ou um valor no intervalo entre 0 (zero) e NUM_MEIOS_PAGAMENTO, onde: -2 representa o Meio de Pagamento pré-definido como "Dinheiro"; qualquer valor do intervalo representa o índice do meio de pagamento programável. Este parâmetro tem precedência sobre o parâmetro <NomeFormaPagamento>.

Variavel: NomeMeioPagamento

TipoDado: [string](#)

Tamanho Máximo: 16

Opcional

Descricao: Nome do Meio de Pagamento.

Variavel: TextoAdicional

TipoDado: [string](#)

Tamanho Máximo: 80

Opcional

Descricao: Texto adicional explicativo referente a operação.

Variavel: Valor

TipoDado: [money](#)

Tamanho Máximo:

Obrigatório

Descricao: Valor da operação. Indica o montante pago com o Meio de Pagamento informado.

Exemplo:

```
DLLG2_LimpaParams(Port);
DLLG2_AdicionaParam(Port, 'CodMeioPagamento', edit1.text, 4);
DLLG2_AdicionaParam(Port, 'NomeMeioPagamento', edit2.text, 7);
DLLG2_AdicionaParam(Port, 'Valor', edit3.text, 6);
DLLG2_ExecutaComando(Port, 'PagaCupom');
```

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)
- [Comprovante Não Fiscal](#)
- [Tipos de Parâmetros](#)

1.9.65 ProgramaRelogio

Descrição: Programa o relógio: data e/ou hora.

Observações: Um dos parâmetros deve ser obrigatoriamente enviado.

Parâmetros

Variavel: Data

TipoDado: [data](#)

Tamanho Máximo:

Opcional

Descricao: Nova data a ser programada.

Variavel: Hora

TipoDado: [hora](#)

Tamanho Máximo:

Opcional

Descricao: Nova hora a ser programada.

TÓPICOS RELACIONADOS:

- [Relógio](#)
- [MIT](#)
- [Tipos de Parâmetros](#)

1.9.66 ReimprimeViaCreditoDebito

Descrição: Reimprime via do Comprovante de Crédito ou Débito (CCD) previamente impressa.

TÓPICOS RELACIONADOS:

- [Comprovante de Crédito ou Débito](#)
- [Tipos de Parâmetros](#)

1.9.67 Vendeltem

Descrição: Emite a venda de um item em cupom fiscal.

Observações: É possível indicar que o valor da venda seja totalizado em um departamento previamente definido.

Parâmetros

Variavel: AliquotaICMS

TipoDado: [bool](#)

Tamanho Máximo:

Opcional

Descricao: Identifica a aliquota como ICMS ('true') ou ISSQN ('false'). Deve ser utilizado em conjunto com o parâmetro <PercentualAliquota> para identificar a alíquota deste produto.

Variavel: CodAliquota

TipoDado: [byte](#) **Tamanho Máximo:** **Opcional / Obrigatório**
Descricao: Índice da alíquota, sendo válidos os valores: intervalo entre 0 e NUM_ALIQUOTAS_PROGRAMAVEIS; -4 = N.Trib. ICMS ou 'N'; -3 = Isento ICMS ou 'I'; -2 = Subst.Trib. ICMS ou 'F'; -11 = 'F' ISSQN, -12 = 'T' ISSQN e -13 = 'N' ISSQN. Se informado tem precedência sobre os parâmetros <AliquotaICMS> e <PercentualAliquota>.

Variavel: CodDepartamento

TipoDado: [byte](#) **Tamanho Máximo:** **Opcional**
Descricao: Índice do departamento entre 1 e NUM_DEPARTAMENTOS. O Departamento 0 (zero) é pré-programado como "Combustível". Quando não informado, o comando de venda de item não acresce o valor vendido a totalizadores de Departamento. Este parâmetro tem precedência sobre o parâmetro <NomeDepartamento>.

Variavel: CodProduto

TipoDado: [string](#) **Tamanho Máximo:** 48 **Obrigatório**
Descricao: Código do produto.

Variavel: NomeDepartamento

TipoDado: [string](#) **Tamanho Máximo:** 15 **Opcional**
Descricao: Nome do departamento. Exemplo: Padaria, Açougue, Têxtil, etc. O Departamento de venda deste produto pode ser informado pelo seu nome opcionalmente ao seu código no parâmetro <CodDepartamento>.

Variavel: NomeProduto

TipoDado: [string](#) **Tamanho Máximo:** 200 **Obrigatório**
Descricao: Nome descritivo do produto.

Variavel: PercentualAliquota

TipoDado: [money](#) **Tamanho Máximo:** **Opcional / Obrigatório**
Descricao: Valor percentual com precisão de 2 casas decimais. Utilizado em conjunto com o parâmetro <AliquotaICMS> é um modo alternativo à indicação da alíquota quando o parâmetro <CodAliquota> não for informado. A alíquota deve estar necessariamente definida.

Variavel: PrecoUnitario

TipoDado: [money](#) **Tamanho Máximo:** **Obrigatório**
Descricao: Preço Unitário. O comando de venda de item trata preços unitários que possuam até 3 casas decimais.

Variavel: Quantidade

TipoDado: [money](#) **Tamanho Máximo:** **Obrigatório**
Descricao: Quantidade envolvida na transação. O comando de venda de item trata quantidades com até 3 casas decimais.

Variavel: Unidade

TipoDado: [string](#) **Tamanho Máximo:** 2 **Opcional**
Descricao: Unidade do produto. Se não informado será assumido o texto "un" (sem aspas).

Exemplo:

```
DLLG2_AdicionaParam (Porta, 'CodAliquota', Produto.Aliq, 4);
DLLG2_AdicionaParam (Porta, 'CodProduto', Produto.Codigo, 7);
DLLG2_AdicionaParam (Porta, 'NomeProduto', Produto.Descricao, 7);
DLLG2_AdicionaParam (Porta, 'PrecoUnitario', Produto.Valor, 6);
DLLG2_AdicionaParam (Porta, 'Quantidade', Produto.Quant, 6);
DLLG2_ExecutaComando(Porta, 'VendeItem');
```

TÓPICOS RELACIONADOS:

- [Cupom Fiscal](#)
- [Tipos de Parâmetros](#)

1.10 Retorno dos Registradores

Este Tópico apresenta o formato de retorno de alguns registradores.

TÓPICOS RELACIONADOS:

- [Mecanismos](#)

1.10.1 Tipos de Documentos

Documento	Tipo
0	Nenhum
1	Relatório X
2	Redução Z
3	Cupom Fiscal
4	Cupom Não Fiscal
5	Cupom Não Fiscal Entrada
6	Cupom Não Fiscal Saída
7	Cupom Fiscal Cancelado
8	Cupom Fiscal Cancelamento
9	Cupom Não Fiscal Cancelamento
10	Cupom Adicional
11	Leitura da Memória Fiscal
12	Comprovante de Crédito ou Débito
13	Comprovante Vinculado
14	Comprovante Não Vinculado
15	Relatório Gerencial
16	Estorno de Meio de Pagamento
17	Estorno de Comprovante de Crédito ou Débito

1.10.2 Estado Geral ECF

O Estado Geral do ECF é a composição de indicadores informando as seguintes situações, a saber:

Bit	Indicação
Bit 1	Inscrições NÃO Ok
Bit 5	RAM NÃO Ok
Bit 9	Documento Aberto
Bit 13	Relógio NÃO Ok
Bit 15	Sem Papel

1.10.3 Dados Última Redução Z

Retorna os seguintes dados seqüencialmente:

Informação	Nº de bytes
Constante "00"	2
GT da última Redução	18
Cancelamentos	14
Descontos	14
Tributos	64
Totalizadores Parciais Tributados	266
Sangria	14
Suprimentos	14
Totalizadores não sujeitos ao ICMS	126
Contadores dos Totalizadores Parciais Tributados não sujeitos ao ICMS	36
Contador de Ordem de Operação (COO)	6
Contador de Operações não sujeitas ao ICMS	6
Número de Alíquotas Cadastradas	2
Data de Movimento	6
Acréscimo	14
Acréscimo Financeiro	14

1.11 Tabela de estados do Software Básico

O equipamento pode estar em apenas um estado em um determinado momento.

Valor	Nome Estado	Descrição
1	ESTADO_ECF_PRONTO	Equipamento está disponível para operação.
2	ESTADO_VENDA_DE_ITEM	Está em venda de cupom fiscal.
4	ESTADO_TOTALIZANDO	Estado do equipamento após operação em subtotal (acréscimo/desconto).
8	ESTADO_PAGAMENTO	Aguarda pagamento após totalização do documento.
16	ESTADO_COMERCIAL	Após o pagamento completo do cupom, recebeu um comando de Mensagem promocional ou identificação do Comprador.
32	ESTADO_GERENCIAL	Relatório Gerencial está aberto.
64	ESTADO_VINCULADO	Comprovante de Crédito/Débito está aberto.
128	ESTADO_VENDA_NFISCAL	Está em venda de itens não fiscais.

TÓPICOS RELACIONADOS:

- [Estados do software básico](#)

1.12 Indicadores do Software Básico

Codificação dos Indicadores (*Flags*), retornado a partir do registrador de Indicadores. Vários indicadores podem estar ativos ou não simultaneamente.

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

É comum que várias situações informadas pelos indicadores ocorram simultaneamente, como por exemplo, Inscrições Carregadas (2048) e Cabeçalho Carregado (4096).

O valor retornado pela leitura de indicadores através do comando 'LeInteiro' representa a soma dos valores de todos os indicadores ativos naquele momento. No exemplo citado anteriormente, se apenas aqueles 2 (dois) indicadores estivessem ativos, o retorno seria 6144 (soma de 2048 + 4096).

Propositalmente, em nenhuma hipótese é possível que a soma dos indicadores ativos representem outro indicador. A rotina necessária para saber quais indicadores estão ativos a partir de um valor lido do ECF é a seguinte:

Passo 1 – Lê o valor do registrador 'Indicadores'

Passo 2 – A partir do valor obtido como retorno, faz:

Se Valor \geq 16384 (Indicador Valor MFD Esgotada)

Então

Indicador Valor MFD Esgotada está ativo

Valor = Valor – 16384

Senão

Indicador Valor MFD Esgotada está inativo

Se Valor \geq 8192 (Indicador Em Linha)

Então

Indicador Em Linha está ativo

Valor = Valor – 8192

Senão

Indicador Em Linha está inativo ...(e assim sucessivamente na ordem decrescente dos valores dos indicadores até chegar ao último)

1.13 Restrições do Tradutor de Protocolo

As seguintes restrições são aplicadas ao Tradutor de Protocolo *Logger I - FiscNET*:

Restrição tradução Logger I	Motivo
· Não é possível realizar a carga de proprietário	A legislação obriga o uso de uma senha para iniciar novo proprietário, o que torna os comandos inconsistentes.
· Somente podem ser definidos (programados) Meios de Pagamento que permitem a emissão de documentos de Crédito ou Débito (CCD).	É obrigatória a definição do atributo PermiteVinculado no momento da carga do Meio de Pagamento. Por default, no protocolo FiscNET, este parâmetro é True (verdadeiro). Portanto, todo meio de pagamento programado através do tradutor de protocolo Logger I habilitará um CCD.
· Código de Barras impresso apenas em Relatórios Gerenciais	O Convênio não prevê a impressão de informações fora de cupons.
· Interpretação do Meio de Pagamento 0 (zero) para o tradutor de protocolos	Os Meios de Pagamento pré-configurados como DINHEIRO possuem códigos diferentes. Na Logger I seu valor era 0 (zero). Na Logger II seu valor é -2. Portanto, é necessário informar ao tradutor de protocolo o significado para o Meio de Pagamento 0 (zero). O comportamento padrão é traduzir o valor 0 (zero) para -2. Caso seja necessário, a tradução deste índice pode ser configurada através do registrador 'TradutorÍndiceMeioPagamentoDinheiro'. O valor deste registrador representará a tradução do Meio de Pagamento 0 (zero).
· Leitura de CMC7 – Comando 53	Não foi implementado o comando de leitura de CMC7 via serial por ser um comando obsoleto e exigir muita memória. Para leitura do código CMC7, aconselha-se usar os comandos 47 – Leitura de CMC7 do cheque, seguido por 3 (três) leituras do comando 04 – Leitura de Registradores: Registrador 159 – campo1 CMC7 Registrador 160 – campo2 CMC7 Registrador 161 – campo3 CMC7
· Códigos de Erro	Nem todos os códigos de retorno da Logger I são usados na tradução do retorno. A tradução é realizada o mais próximo possível para o código de retorno da Logger I. Alguns códigos Logger II não possuem tradução correspondente para a Logger I. Neste caso, o tradutor utiliza o código 42 (decimal) como retorno padrão para aqueles que não possuem tradução correspondente.
· Emissão de nova via de um Comprovante Vinculado	O processo de emissão de nova via do comprovante vinculado não está implementado no tradutor de protocolo. De forma mais simples e fácil, a emissão de nova via de Comprovante Vinculado deve ser realizada através do comando 60 do tradutor, enviando o comando FiscNET "EmiteViaCreditoDebito".

1.14 Recomendações de Uso da MFD

- O comando `ImprimeTexto`, quando utilizado dentro de um Relatório Gerencial ou de um Cupom de Crédito/Débito, deve conter o máximo de conteúdo a ser impresso em cada comando, pois a cada comando é registrado um certo número de informações adicionais ao texto na MFD. Não recomenda-se a utilização do comando `ImprimeTexto` para cada linha impressa.
- Evitar o uso de dados variáveis nos campos de descrição e/ou código dos produtos na venda de item. Esta prática reduz a eficácia do compactador de dados do equipamento.