

Manual de integração



**OPEN
TEF**

Sumário

RESUMO.....	3
Exemplo de transação pelo Open-Tef.....	5
INTRODUÇÃO.....	6
1 – FRENTE DE CAIXA(PDV).....	6
1.1 – Funções.....	6
1.2 – Bibliotecas.....	7
1.2.1 – Função auxiliar para tratar a mensageiria.....	10
1.2.1.1 – Função “mensagemcreate”.....	10
1.2.1.2 – Função “mensagemcarregatags”.....	10
1.2.1.3 – Função “mensagemcomando”.....	11
1.2.1.4 – Função “mensagemcomandodados”.....	11
1.2.1.5 – Função “mensagemgettag”.....	12
1.2.1.6 – Função “mensagemgettagidx”.....	12
1.2.1.7 – Função “mensagemtagcount”.....	13
1.2.1.8 – Função “mensagemaddtag”.....	13
1.2.1.9 – Função “mensagemaddcomando”.....	14
1.2.1.10 – Função “mensagetagasString”.....	14
1.2.1.11 – Função “mensagemtagtostr”.....	14
1.2.1.12 – Procedimento “mensagemlimpar”.....	15
1.2.1.13 – Procedimento “mensagemfree”.....	15
1.2.1.14 – Procedimento “mensagemdispose”.....	16
1.3 – Configuração.....	16
1.3.1 – IP/porta servidor tef.....	16
1.3.2 – Chave de identificação do Terminal.....	16
1.3.3 – Chave da transação.....	16
1.3.4 – Identificador do terminal.....	17
1.4 – Processo de conexão com o Open-Tef.....	17
1.4.1 – Inicialização – nome da função dentro da biblioteca “inicializar”.....	18
1.4.2 – Login – nome da função dentro da biblioteca “login”.....	20
1.4.3 – Verificar conexão – nome da função dentro da biblioteca “opentefstatus”.....	21
1.4.4 – Solicitação blocante – nome da função dentro da biblioteca “solicitacaoblocante”.....	21
1.5 – Funções do PDV usada pelo Open-Tef.....	22
1.5.1 – Impressão.....	22
1.5.2 – Função mostra menu.....	23
1.5.3 – Função mensagem para o operador.....	24
1.5.4 – Função solicita dados do PDV.....	24
1.5.5 – Função solicita dados da transação.....	26
1.5.6 – Função retorno de comandos diversos.....	27
1.5.7 – Função desaloca ponteiro da memória strdispose.....	29
1.6 – Processo de venda.....	30
1.6.1 – Inicia uma transação – nome da função dentro da biblioteca “transacaocreate”.....	30
1.6.2 – Confirma transação.....	31
1.6.3 – Consultar status da transação – nome da função dentro da biblioteca “transacaostatus”.....	32
1.6.4 – Consultar descrição do erro – nome da função dentro da biblioteca “mensagemerro”.....	33

1.6.5 – Consultar descrição do erro da transação – nome da função dentro da biblioteca “transacaostatusdescricao”.....	33
1.6.6 – Captura dados do cliente.....	34
1.6.7 – Cancelamento de venda.....	34
1.6.8 – Desalocar transação da memória – nome do procedimento dentro da biblioteca “transacaofree”.....	34
1.7 – Consultar saldo.....	35
1.8 – Verificar versão do tef – nome da função dentro da biblioteca “versão”	35
2 – Tabelas auxiliares.....	36
2.1 – Tabela de código de retorno (Tabela erro).....	36
2.2 – TABELA DE TAGS.....	39
2.3 – TABELA DE STATUS DE TRANSAÇÃO.....	49
2.4 – TABELA DE STATUS DA CONEXÃO.....	49
3 – MENSAGEIRIA.....	49
3.1 – ESTRUTURA DO PACOTE.....	49
3.2 – TIPOS DE DADOS.....	50
3.2.1 – TAGS PADRONIZADAS.....	51
3.3 – PACOTES.....	51
3.3.1 – EXEMPLO DE PACOTE.....	51
4 – VERSIONAMENTO.....	52

RESUMO

FUNÇÕES DISPONÍVEIS NO OPEN_TEF ATRAVÉS DA DLL OU LIB

Devem ser consumidas pela frente de caixa

FUNÇÕES DE CONEXÃO

- inicializar (tópico 1.4.1)
- login (tópico 1.4.2)
- opentefstatus (tópico 1.4.3)
- solicitacaoblocante (tópico 1.4.4)

FUNÇÕES USADAS PARA TRATAR A MENSAGEIRIA

- mensagemcreate (tópico 1.2.1.1)
- mensagemcarregatags (tópico 1.2.1.2)
- mensagemcomando (tópico 1.2.1.3)
- mensagemcomandodados (tópico 1.2.1.4)
- mensagemgettag (tópico 1.2.1.5)
- mensagemgettagidx (tópico 1.2.1.6)
- mensagemtagcount (tópico 1.2.1.7)
- mensagemaddtag (tópico 1.2.1.8)
- mensagemaddcomando (tópico 1.2.1.9)
- mensagemtagasstring (tópico 1.2.1.10)

mensagemtagtostr (tópico 1.2.1.11)
mensagemlimpar (tópico 1.2.1.12)
mensagemfree (tópico 1.2.1.13)

FUNÇÕES USADAS PARA TRANSAÇÃO

transacaocreate (tópico 1.6.1)
transacaostatus (tópico 1.6.3)
transacaostatusdescricao (tópico 1.6.5)
transacaofree (tópico 1.6.8)

FUNÇÕES DIVERSAS

mensagemerro (tópico 1.6.4)
versão (tópico 1.8)
StrDispose

FUNÇÕES DO PDV USADA PELO OPEN-TEF

Devem ser criadas pela frente de caixa para a utilização do Open-Tef

IMPRESSÃO (tópico 1.5.1)
MOSTRA MENU (tópico 1.5.2)
MENSAGEM PARA O OPERADOR (tópico 1.5.3)
SOLICITA DADOS DO PDV (tópico 1.5.4)
SOLICITA DADOS DA TRANSAÇÃO (tópico 1.5.5)
RETORNO COMANDOS DIVERSOS (tópico 1.5.6)

Exemplo de transação pelo Open-Tef

Frente de caixa	OpenTef
	
Função “inicializar” (tópico 1.4.1) ➔	
Função “login” (tópico 1.4.2) ➔	Retorno 0 ou código de erro
Função “transacaocreate” (tópico 1.6.1) ➔	Devolve id da transação
Função “transacaostatus (loop)” (tópico 1.6.3) ➔	Devolve status da transação (aprovada, negada, processando) e a chave da transação, inicia as etapas automatizadas
Etapas automatizadas pelo Open Tef, as seguintes funções podem ser utilizadas	
Mostra um menu com as opções de venda do Open-Tef	⬅ Função “mostramenu” (tópico 1.5.2)
Devolve os dados solicitados com interação do operador de caixa	⬅ Função “solicitadadospdv” (tópico 1.5.4)
Devolve tag solicitadas referente a transação como valor, parcela..	⬅ Função “solicitadadostransacao” (tópico 1.5.5)
Mostra uma mensagem para o operador de caixa	⬅ Função “mensagemoperador” (tópico 1.5.3)
Imprimir o comprovante da venda	⬅ Função “impressao” (tópico 1.5.1)
Sair do loop	⬅ Altera o status da transação para aprovado ou negado.

INTRODUÇÃO

O Open-Tef é um sistema que realiza a comunicação entre a frente de caixa e a operadora/bandeira/adquirência

Sem custo

Por ser código aberto não possui nenhum tipo de custo, além disso qualquer um consegue analisar o código ou enviar sugestões

Segurança

A biblioteca utiliza a criptografia RSA para a troca de mensagens, assim nem o Open-Tef tem acesso a ela, além das transações não serem armazenadas

Comunicação

A comunicação ocorre através de mensagens no formato de mensageira(conforme tópico 3.1) contendo tags de forma que cada uma possua um significado

Liberdade

O Open-Tef permite a personalização das vendas como menus, bins, captura de dados como mensagens, imagens, qrcode. Também pode-se utilizar o configurador para configurar pinpad, restringir menus para o PDV

Funções automatizadas

Funções como criptografia de mensagens, comunicação com o pinpad são automatizadas

1 – FRENTE DE CAIXA(PDV)

1.1 – Funções

A integração de seu programa frente de caixa (PDV) com o Open-Tef deve seguir as etapas seguintes:

Você deve criar 6 funções que o Open-Tef chamará quando necessário (verifique o tópico 1.5)

- A)Função de Impressão(tópico 1.5.1)
- B)Função de Mostrar Menu(tópico 1.5.2)
- C)Função de Mensagem para o Operador (tópico 1.5.3)
- D)Função de Solicitação de Dados com interação com o operador do PDV (tópico 1.5.4)
- E)Função de Solicitação de Dados da Transação sem interação(tópico 1.5.5)
- F)Função de Retorno de comandos diversos(tópico 1.5.6)

1.2 – Bibliotecas

As bibliotecas a serem usadas no Frente de Caixa (PDV) são:

tef_lib (biblioteca do Open-Tef)
pinpad_lib (biblioteca do pinpad no Open-Tef)
dll proprietária (biblioteca do fabricante do pinpad)

Devem ser referenciadas com a definição “cdecl” compatível com Linux e Windows
Usa somente 3 tipos de parâmetros para simplificação das trocas de mensagens:

Ponteiro = **ponteiro** para algum objeto manipulado pela biblioteca.

PChar = **ponteiro** para uma string / **ponteiro** para um char em “c++”

Integer = valor **inteiro** de 32 bits

Ponteiro de Função = **ponteiro** para Funções ou Procedimentos que a biblioteca usa no programa que a chama, os tipos definidos são: Retorno, Impressão, MostraMenu, MensagemOperador, SolicitaDadosTransacao, SolicitaDadosPDV

Observação: Para facilitar a manipulação da messageiria use as funções do tópico 1.2.1

“Retorno”

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Dados solicitados a serem executados no formato de mensageira	PChar
Entrada/Saída	Retorno dos dados solicitados no formato de mensageira	PChar

function(VP_DadosEntrada: PChar; var VO_DadosSaida: PChar): integer; cdecl;

“Impressão”

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Texto a ser impresso (Seguir estrutura de montagem conforme o tópico 3.2 (impressão))	PChar

function(VP_Dados: PChar): integer; cdecl;

“MostraMenu”

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Menu estruturado no formato de mensageira	PChar
Entrada/Saída	Tag do botão selecionado	PChar

function(VP_Menu: PChar; var VO_Botao: PChar): integer; cdecl;

“MensagemOperador”

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Mensagem a ser exibida (Seguir estrutura de montagem conforme o tópico 3.2 (mensagem))	PChar

function(VP_Dados: PChar): integer; cdecl;

“SolicitaDadosTransacao”

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Tags solicitadas no formato de mensageira (Tags conforme tópico 1.6, além da chave da transação tag 00F1)	PChar
Entrada/Saída	Dados de retorno das tags solicitadas no formato de mensageira	PChar

function(VP_Mensagem: PChar; var VO_Dados: PChar): integer; cdecl;

“SolicitaDadosPDV”

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Dados solicitados no formato de mensageira	PChar
Entrada/Saída	Retorno da tag do botão selecionado	PChar
Entrada/Saída	Retorno dos dados solicitados	PChar

function(VP_Menu: PChar; var VO_Botao, VO_Dados: PChar): integer; cdecl;

1.2.1 – Função auxiliar para tratar a mensageiria

Funções auxiliares para gerar e ler a Mensageiria(Tópico 3) dos exemplos citados:

1.2.1.1 – Função “mensagemcreate”

Essa função cria um ponteiro para mensageiria

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada/Saída	Ponteiro para a mensageiria	Ponteiro

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_MensagemCreate = function(var VO_Mensagem: Pointer): integer; cdecl;
```

1.2.1.2 – Função “mensagemcarregatags”

Essa função carrega texto estruturado no formato de mensageira em ponteiro para mensageiria

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro
Entrada	Texto que deseja carregar	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
F_MensagemCarregaTags = function(VP_Mensagem: Pointer; VP_Dados: PChar): integer; cdecl;
```

1.2.1.3 – Função “mensagemcomando”

Essa função retorna a tag de comando de uma messageira

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro
Entrada/Saída	Retorno da tag de comando	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
F_MensagemComando = function (VP_Mensagem: Pointer; var VO_Dados: PChar): integer; cdecl;
```

Obs: Após usar os dados de retorno(VO_Dados), deve chamar a função

F_MensagemDispose(VO_Dados) para ser liberado da memória.

1.2.1.4 – Função “mensagemcomandodados”

A função “mensagemcomandodados” retorna o dado da tag de comando de uma messageira

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro
Entrada/Saída	Retorno do dado da tag de comando	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
F_MensagemComandoDados = function mensagemcomandodados(VP_Mensagem: Pointer; var VO_Dados: PChar): integer; cdecl;
```

Obs: Após usar os dados de retorno(VO_Dados), deve chamar a função

F_MensagemDispose(VO_Dados) para ser liberado da memória.

1.2.1.5 – Função “mensagemgettag”

Essa função retorna a mensageiria em formato texto.

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro
Entrada	Tag da mensageiria que deseja recuperar(tabela de Tag do tópico 2.2 deste manual)	PChar
Entrada/Saída	Dados de retorno da tag	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_MensagemGetTag = function(VO_Mensagem: Pointer; VP_Tag: PChar; var VO_Dados: PChar): integer;
cdecl;
```

Obs: Após usar os dados de retorno(VO_Dados), deve chamar a função

F_MensagemDispose(VO_Dados) para ser liberado da memória.

1.2.1.6 – Função “mensagemgettagidx”

Essa função retorna a tag e os dados contido nela no índice de uma mensageira.

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro
Entrada	Índice da tag na mensageiria	Inteiro
Entrada/Saída	Retorno da tag da mensageiria	PChar
Entrada/Saída	Retorno dos dados da tag	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_MensagemGetTagIdx = function(VO_Mensagem: Pointer; VL_Indice: integer; var VO_Tag: PChar; var
VO_Dados: PChar): integer; cdecl;
```

Obs: Após usar os dados de retorno deve chamar a função

F_MensagemDispose(VO_Tag) para ser liberado da memória.
F_MensagemDispose(VO_Dados) para ser liberado da memória.

1.2.1.7 – Função “mensagemtagcount”

A função “mensagemtagcount” conta a quantidade de tags que uma mensageiria possui.

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro

O retorno da função devolve um inteiro correspondente a quantidade de tags na mensageiria.

Exemplo em pascal:

```
//estrutura
F_MensagemTagCount = function(VO_Mensagem: Pointer): integer; cdecl;
```

1.2.1.8 – Função “mensagemaddtag”

Essa função adiciona uma tag na mensageira.

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro
Entrada	Tag	PChar
Entrada	Dados da tag	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_MensagemAddTag = function(VP_Mensagem: Pointer; VP_Tag, VP_Dados: PChar): integer; cdecl;
```

1.2.1.9 – Função “mensagemaddcomando”

Essa função adiciona uma tag de comando na messageira.

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a messageiria	Ponteiro
Entrada	Tag de comando	PChar
Entrada	Dados da tag	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_MensagemAddComando = function(VP_Mensagem: Pointer; VP_Tag, VP_Dados: PChar): integer; cdecl;
```

1.2.1.10 – Função “mensagetagasString”

Essa função carrega uma messageiria em texto.

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a messageiria	Ponteiro
Entrada/Saída	Retorno da messageira em texto	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_MensagemTagsAsString = function(VP_Mensagem: Pointer; var VO_PChar: PChar): integer; cdecl;
```

Obs: Após usar os dados de retorno(VO_PChar), deve chamar a função F_MensagemDispose(VO_PChar) para ser liberado da memória.

1.2.1.11 – Função “mensagetagtostr”

Essa função carrega uma mensageiria em texto.

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro
Entrada/Saída	Retorno da mensageira em texto	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_MensagemTagToStr = function(VP_Mensagem: Pointer; var VO_PChar: PChar): integer; cdecl;
```

Obs: Após usar os dados de retorno(VO_PChar), deve chamar a função

F_MensagemDispose(VO_PChar) para ser liberado da memória.

1.2.1.12 – Procedimento “mensagemlimpar”

Esse procedimento limpa todas as tags de um ponteiro para a mensageira

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro

Exemplo em pascal:

```
F_MensagemLimpar = procedure mensagemlimpar(VP_Mensagem: Pointer); cdecl;
```

1.2.1.13 – Procedimento “mensagemfree”

Esse procedimento desloca da memória esse ponteiro para mensageira

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a mensageiria	Ponteiro

Exemplo em pascal:

```
F_MensagemLimpar = procedure(VP_Mensagem: Pointer); cdecl;
```

1.2.1.14 – Procedimento “mensagemdispose”

Esse procedimento desloca da memória uma variável alocada para o retorno de alguma função.

Deve conter os seguintes parâmetros:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	PChar de retorno de alguma função	PChar

Exemplo em pascal:

```
F_MensagemDispose = procedure(VP_Retorno: PChar); cdecl;
```

1.3 – Configuração

Para a configuração da frente de caixa para o uso do Open-Tef, o mesmo deve conter:

Descrição	Tipo de Dados
Código de identificação	Inteiro
IP do PDV	PChar
Chave de identificação do Terminal(tópico 1.3.2)	PChar
Identificador do terminal(tópico 1.3.4)	PChar

1.3.1 – IP/porta servidor tef

Deve se configurar o host e a porta do local onde o Open-Tef rodará.

1.3.2 – Chave de identificação do Terminal

A chave de identificação do Terminal junto ao Open-Tef, essa chave deve ser única e preferencialmente gerada no Open-Tef.

1.3.3 – Chave da transação

A chave da transação(tag 00F1) é gerada automaticamente pelo Open-Tef quando a solicitação for iniciada. Essa chave é comum para toda referência a essa transação, pois contém todas as informações necessárias para o PDV, Open-Tef, e bandeira ou adquirência.

Ela está no formato de mensageira e contém:

(tag 00F9) código da loja na adquirência
(tag 0107) código do PDV na adquirência
(tag 0091) documento do estabelecimento
(tag 0034) id da transação
(tag 0036) bin da bandeira
(tag 00F2) tag do modulo
(tag 00A2) tag do tipo de terminal
(tag 0108) tag de identificação do Terminal (Comum em várias instalações do OpenTef)
(tag 0109) identificação do adquirente
(tag 00F7) autorização da venda(gerada pela operadora/bandeira/adquirência quando a venda for concluída).

1.3.4 – Identificador do terminal

Esse identificador é a referência desse terminal para o Open-Tef, portanto ele é único e não será aceito caso já esteja sendo utilizado.

Por isso recomendamos que a criação do identificador siga a seguinte sugestão:

*“Sugestão: cnpj + id do terminal + descrição”
“Exemplo: 11.111.111/0001-1123001/lojacentro”*

1.4 – Processo de conexão com o Open-Tef

A conexão com o Open-Tef ocorre através da biblioteca que cria uma conexão criptografada e dessa forma não é necessário a criação de uma vpn.

No processo de transação para aprovar uma venda a comunicação ocorre criando um túnel dentro da conexão do Open-Tef direto com a operadora, dessa forma nem o Open-Tef tem acesso as mensagens trocadas como número do cartão, valor, nome do cliente...

O processo de conexão com o Open-Tef segue as seguintes etapas:

- 1 – Inicializar o processo(tópico 1.4.1)
- 2 – Logar no Open-Tef(tópico 1.4.2)

Abaixo o diagrama do fluxo de dados no processo de transação



“Atenção: os nomes das funções devem ser declaradas exatamente como consta nas explicações abaixo, pois são referenciadas na biblioteca(dll ou so)”

1.4.1 – Inicialização – nome da função dentro da biblioteca “inicializar”

Esta função tem a seguinte estrutura: e tem retorno do tipo Inteiro

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada/Saída	Ponteiro para a instância do OpenTef (ponteiro alocado pela biblioteca que será solicitado pelas demais funções)	Ponteiro
Entrada	Modelo do PinPad (0 = nenhum 1 = Gertec PPC 930)	Inteiro
Entrada	Modelo da Lib do PinPad (Caminho da biblioteca LIB do PinPad conforme fabricante)	PChar
Entrada	Porta do PinPad (Porta de comunicação que o PinPad usará)	PChar
Entrada	PinPad Lib (Caminho da Lib (pinpad_lib) do PinPad para uso do Open-Tef)	PChar
Entrada	Arquivo de Log (Caminho onde será gravado o arquivo de Log do sistema)	PChar
Entrada	Procedimento (Função que desaloca da memória um ponteiro PChar)	Ponteiro para a Função (StrDispose)
Entrada	Procedimento (Função de retorno criada para uso do Open-Tef)	Ponteiro para a Função (Retorno)
Entrada	Solicita dados transação (Função de solicitação de dados da transação sem interação criada para uso do Open-Tef tópico 1.5.5)	Ponteiro para a Função (SolicitaDadosTransacao)
Entrada	Solicita dados do PDV (Função de solicitação de dados com interação com o operador do PDV, função criada para uso do Open-Tef tópico 1.5.4)	Ponteiro para a Função (SolicitaDadosPDV)
Entrada	Imprimir (Função de impressão de dados, função criada para uso do Open-Tef tópico 1.5.1)	Ponteiro para a Função (Impressão)
Entrada	Mostra menu	Ponteiro para a Função

	(Função de visualização do menu pelo operador do PDV, função criada para uso do Open-Tef tópico 1.5.2)	(MostraMenu)
Entrada	Mensagem para o operador (Função de mensagem a ser exibida no PDV para o operador, função criada para uso do Open-Tef tópico 1.5.3)	Ponteiro para a Função (MensagemOperador)
Entrada	Ambiente de teste (Quando ativado o PDV realizará somente transações de testes e não conectará externamente a lib simulará o Open-Tef 1:ativado 0:desativado)	Inteiro

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
function inicializar(var VO_Tef: Pointer; VP_PinPadModelo: integer; VP_PinPadModeloLib,
VP_PinPadModeloPorta, VP_PinPadLib, VP_ArquivoLog: PChar; VP_StrDispose: TStrDispose;
VP_Procedimento: TRetorno; VP_SolicitaDadosTransacao: TSolicitaDadosTransacao;
VP_SolicitaDadosPDV: TSolicitaDadosPDV;
VP_Imprime: TImprime; VP_MostraMenu: TMostraMenu; VP_MensagemOperador:
TMensagemOperador; VP_AmbienteTeste: integer): integer; cdecl;

//aplicação
1. VL_Erro := F_TefIniciar(F_Tef, VL_PinPadModelo, VL_PinPadModeloLib,
VL_PinPadModeloPorta, VL_PinPadLib, VL_ArquivoLog, VL_StrDispose, VL_Procedimento,
SolicitaDadosTransacao, SolicitaDadosPDVTSolicitaDadosPDV, Imprime, MostraMenu,
MensagemOperador,0);
```

1.4.2 – Login – nome da função dentro da biblioteca “login”

Esta função tem a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a instância do OpenTef (ponteiro retornado pela biblioteca no tópico 1.4.1)	Ponteiro
Entrada	Host (Host de acesso)	PChar
Entrada	Porta (Porta de acesso)	Inteiro

Entrada	ID do terminal	Inteiro
Entrada	Chave de acesso (Chave de identificação do terminal descrita no tópico 1.3.2 deste manual)	PChar
Entrada	Versão da mensagem (Número da versão da messageira)	Inteiro
Entrada	Identificação do terminal (tópico 1.3.4)	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
function(VP_Tef: Pointer; VP_Host: PChar; VP_Porta,VP_ID: integer; VP_Chave: PChar;
VP_Versao_Comunicacao:    integer;VP_Identificador: PChar): integer; cdecl;

//aplicação
VL_Erro := F_Login(F_Tef, VL_Host, VL_Porta, VL_ID, VL_Chave, VL_Mensagem, VL_Identificacao);
```

1.4.3 – Verificar conexão – nome da função dentro da biblioteca “opentefstatus”

Esta função verifica o status da conexão com o Open-Tef e tem a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a instância do OpenTef (ponteiro retornado pela biblioteca no tópico 1.4.1)	Ponteiro
Entrada/Saída	Retorno do status da conexão (Retornado pelo Open-Tef, esse código representa o status da conexão conforme tópico 2.4)	Inteiro

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
function opentefstatus(VP_Tef: pointer; var VO_StatusRetorno: integer): integer; cdecl

//aplicação
VL_Erro := F_StatusOpenTef(F_Tef, VL_Status);
```

1.4.4 – Solicitação blocante – nome da função dentro da biblioteca “solicitacaoblocante”

Esta função envia uma solicitação e tem a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a instância do OpenTef (ponteiro retornado pela biblioteca no tópico 1.4.1)	Ponteiro
Entrada/Saída	ID da transmissão	PChar
Entrada	Dados a serem envidados no formato de mensageira	PChar
Entrada/Saída	Dados de retorno das tags solicitadas no formato de mensageira	PChar
Entrada	Tempo de espera (Tempo em que o sistema esperará o retorno do Open-Tef em milissegundos)	Inteiro

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
function solicitacaoblocante(VP_Tef: pointer; var VO_Transmissao_ID, VP_Dados: PChar; var
VO_Retorno: PChar; VP_TempoAguarda: integer): integer; cdecl;

//aplicação
VL_Erro := F_SolicitacaoBlocante(F_Tef, VO_Transmissao_ID, VL_Dados_Entrada, VL_Dados_Saida,
60000);
```

1.5 – Funções do PDV usada pelo Open-Tef

Funções necessárias para a integração com o Open-Tef.

1.5.1 – Impressão

Esta função imprime o texto solicitado de estrutura conforme o tópico 3.2 (impressão) com base em sua impressora.

A função de impressão deve ter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Texto a ser impresso (Seguir estrutura de montagem conforme o tópico 3.2 (impressão))	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
function imprime(VP_Dados: PChar): integer; cdecl;
var
  VL_Texto: string;
begin
  Result := 0;
  VL_Texto := VP_Dados;
  // programar impressão do texto de acordo com sua impressora
end;
```

1.5.2 – Função mostra menu

Esta função cria o menu dinamicamente para o operador do PDV e retorna ao Open-Tef a tag do botão selecionado.

A função deve ter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Menu estruturado no formato de mensageira	PChar
Entrada/Saída	Retorno da tag do botão selecionado	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Necessário usar as funções auxiliares(tópico 1.2.1) para tratar o menu ou desenvolver esse tratamento segundo a estrutura apontada no tópico 3.1 desse manual.

Exemplo em pascal

```
function mostramenu(VP_Menu: PChar; var VO_Botao: PChar): integer; cdecl;
var
```

```

VL_btn: TMButton;
VL_I: integer;
VL_Tag: PChar;
VL_Dados: PChar;
VL_MenuVenda: TF_MenuVenda;
begin
  Result := 0;
  VL_Tag := "";
  VL_Dados := "";
  F_MensagemCarregaTags(F_Mensagem, VP_Menu);
  for VL_I := 1 to F_MensagemTagCount(F_Mensagem) do // Loop para montar os botões
    begin
      // adiciona os botões com suas tags correspondentes conforme retorno do Open-Tef
      VL_btn := TMButton.Create(VL_MenuVenda.PBotao); // cria o botão
      F_MensagemGetTagIdx(F_Mensagem, VL_I, VL_Tag, VL_Dados); // pega as informações do botão
      VL_btn.V_tag := VL_Tag; // adiciona a tag do botão
      VL_btn.Caption := VL_Dados; // adiciona o texto do botão
      VL_btn.Parent := VL_MenuVenda.PBotao; // adiciona botão no modal
      VL_btn.OnClick := @F_Principal.CliqueDoBotao; // adiciona evento de clique
    end;
  VL_MenuVenda.ShowModal; // mostra o menu
  VO_Botao := StrAlloc(Length(VL_MenuVenda.V_Botao) + 1);
  StrPCopy(VO_Botao, VL_MenuVenda.V_Botao); // botão selecionado
  VL_MenuVenda.Free;
end;

```

1.5.3 – Função mensagem para o operador

Exibe uma mensagem de estrutura conforme o tópico 3.2 (mensagem) para o operador do PDV.

Esta função deve ter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Mensagem a ser exibida ao operador de caixa (Seguir estrutura de montagem conforme o tópico 5.2 (mensagem))	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal

```

// mostra o texto VP_Dados para o operador, no exemplo como caixa de mensagem
function mensagemoperador(VP_Dados: PChar): integer; cdecl;
var
  VL_String: string;
begin
  Result := 0;

```

```

VL_String := VP_Dados;
ShowMessage(VL_String); // programar exibição de mensagem de acordo com seu sistema
end;

```

1.5.4 – Função solicita dados do PDV

Esta função carrega dinamicamente as solicitações de dados enviadas para o PDV e trata o seu retorno com a interação do operador.

Pode conter as seguintes tags:

- 1º Mostrar uma mensagem ao operador (tag 00DA)
- 2º Mostrar uma imagem ao operador do caixa (Monitor) (tag 002E)

A imagem é passado como texto, então será preciso converter de texto para imagem
- 3º Listar uma opção de botões (tag 00DD)

Está no formato de mensageira e contém a lista de botões a serem carregados
- 4º Mostrar um campo para responder uma pergunta sendo mascarado ou não(tag 0033)

'M' para campo mascarado
'A' para campo não mascarado

A função solicita dados do PDV deve ter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Dados solicitados no formato de mensageira	PChar
Entrada/Saída	Retorno da tag do botão selecionado	PChar
Entrada/Saída	Retorno dos dados solicitados	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Necessário usar as funções auxiliares(tópico 1.2.1) para tratar as solicitações ou desenvolver esse tratamento segundo a estrutura apontada no tópico 5 desse manual.

Exemplo em pascal

```

function solicitadadospdv(VP_Mensagem: PChar; var VO_Botao, VO_Dados: PChar): integer; cdecl;
var
  VL_btn: TMButton;

```

```

VL_I: integer;
VL_Tag: PChar;
VL_Dados: PChar;
VL_MenuVenda: TF_MenuVenda;
VL_Imagen:String;
begin
  Result := 0;
  VL_Tag := "";
  VL_Dados := "";
  F_MensagemCarregaTags(F_Mensagem, VP_Mensagem); // tópico 1.2.1(função auxiliar)
  VL_I := F_MensagemGetTag(F_Mensagem, '0033', VL_Dados); // captura de dados
  if VL_Dados = 'M' then // campo deve ser mascarado
    VL_MenuVenda.EDados.PasswordChar := '**'; // mudando campo para mascarado
  VL_MenuVenda.ShowModal; // mostra modal com as informações
  VO_Dados := StrAlloc(Length(VL_MenuVenda.EDados.Text) + 1);
  StrPCopy(VO_Dados, VL_MenuVenda.EDados.Text); // retorno do dado solicitado
  VO_Botao := StrAlloc(Length(VL_MenuVenda.V_Botao) + 1);
  StrPCopy(VO_Botao, VL_MenuVenda.V_Botao); // retorno botão selecionado pelo usuário
  VL_MenuVenda.Free;
end;

```

1.5.5 – Função solicita dados da transação

Esta função carrega os dados da transação solicitados pela operadora e trata o seu retorno sem interação com o operador, por isso ela deve testar e responder todos os dados presentes no tópico 1.6, além da chave da transação(tag 00F1) retornada no tópico 1.6.3

Se algum dado solicitado não for respondido pode haver a negação da transação pela operadora/bandeira/adquirênciia.

A função solicita dados da transação deve ter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Tags solicitadas no formato de mensageira (Tags conforme tópico 1.6, além da chave da transação tag 00F1)	PChar
Entrada/Saída	Dados de retorno das tags solicitadas no formato de mensageira	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Necessário usar as funções auxiliares(tópico 1.2.1) para tratar as solicitações ou desenvolver esse tratamento segundo a estrutura apontada no tópico 5 desse manual.

Exemplo em pascal:

```

function solicitadadostransacao(VP_Mensagem: PChar; var VO_Dados: PChar): integer; cdecl;
var
  VL_I: integer;
  VL_Tag: PChar;
  VL_Dados: PChar;
  VL_PChar: PChar;
  VL_Resposta: Pointer;
begin
  Result := 0;
  VL_Tag := "";
  VL_Dados := "";
  VL_PChar := "";
  VL_Resposta := nil;
  F_MensagemCreate(VL_Resposta); // criando uma mensageira
  F_MensagemAddComando(VL_Resposta, '00E1', 'R'); // comando de retorno
  F_MensagemCarregaTags(F_Mensagem, VP_Mensagem); // carregando os dados solicitados
  for vl_i := 1 to F_MensagemTagCount(F_Mensagem) do // lista de dados solicitados pela operadora
    begin
      F_MensagemGetTagIdx(F_Mensagem, VL_I, VL_Tag, VL_Dados); // tópico 1.2.1(função auxiliar)
      if VL_Tag = '0013' then // verifica se o valor foi solicitado
        F_MensagemAddTag(VL_Resposta, '0013', Pchar(Form_Principal.EvalorItens.Text)); // adiciona valor
        // verificar os outros dados conforme tópico 1.6, além da chave da transação
    end;
  F_MensagemTagAsString(VL_Resposta, VL_Pchar); // converte em string
  F_MensagemFree(VL_Resposta);
  VO_Dados := StrAlloc(Length(VL_PChar) + 1);
  StrPCopy(VO_Dados, VL_Pchar); // retorno dos dados solicitados
end;

```

1.5.6 – Função retorno de comandos diversos

Essa função recebe comandos variados solicitados pelo Open-Tef e bandeiras, o retorno pode ou não devolver dados de acordo com o comando.

Pode conter as seguintes tags de comandos:

1º retorno com erro (tag 0026):

Contém o número do erro(tópico 2.1) no dado do comando, a mensagem do erro pode ser lida através da função “mensagemerro” (tópico 1.6.4).

2º pedido para mostrar menu(tag 0018)

É preciso montar um menu de opções para o operador de caixa com as tags solicitadas, do mesmo modo da função mostra menu(tópico 1.5.2)

3º solicitação de tag (tag 0104)

As tags que podem ser solicitadas são:

versão do tef(tag 0005), é possível verificar a versão do tef através da função “versão” (tópico 1.8)

Para responder deve-se adicionar a tag 00F4 como comando contendo o dado ‘R’, além de adicionar as informações pedidas como tags.

4º atualização da lib tef(tag 010C)

Existem dois tipos de atualização, atualização obrigatória(tag 00FD) e atualização opcional(tag 010A), ao ser solicitada o dado dessa tag terá o valor “S“, assim saberemos qual for solicitada.

Para atualizar deve-se retornar a tag de comando 010C com o dado contendo o caminho para ser baixada a nova dll ou lib. Após o download ser concluído é necessário inicializar o frente de caixa com essa nova biblioteca, ou seja, deve pegar esse arquivo que foi baixado e substituir a dll ou lib.

A função deve conter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Tags solicitadas no formato de mensageira (Tags conforme tópico 1.6, além da chave da transação tag 00F1)	PChar
Entrada/Saída	Dados de retorno das tags solicitadas no formato de mensageira	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Necessário usar as funções auxiliares(tópico 1.2.1) para tratar as solicitações ou desenvolver esse tratamento segundo a estrutura apontada no tópico 5 desse manual.

Exemplo em pascal:

```
function Retorno(VP_DadosEntrada: PChar; var VO_DadosSaida: PChar): integer; cdecl;
var
  VL_Dados: PChar;
  VL_PChar: PChar;
  VL_Mensagem: Pointer;
```

```

VL_Erro: integer;
VL_DescricaoErro: PChar;
begin
  Result := 0;
  VL_Mensagem := nil;
  F_MensagemCreate(VL_Mensagem);
  VL_Dados := "";
  VL_DescricaoErro := "";
  VL_PChar := "";

  F_MensagemCarregaTags(VL_Mensagem, PChar(VP_DadosEntrada));

  F_MensagemComando(VL_Mensagem, VL_Dados);
  F_MensagemComandoDados(VL_Mensagem, VL_PChar);
  if VL_Dados = '0026' then // retorno com erro
  begin
    if VL_PChar = '96' then // desconectado
    begin
      F_Principal.lblStatusConexao.Caption := 'Desconectado';
      F_Principal.lblStatusConexao.Font.Color := clRed;
    end;

    F_Erro(StrToInt(VL_PChar), VL_DescricaoErro);
    ShowMessage('Erro: ' + VL_PChar + #13 + 'Descrição: ' + VL_DescricaoErro);
  end;
end;

```

1.5.7 – Função desaloca ponteiro da memória strdispose

Este procedimento desaloca da memória um ponteiro do tipo PChar e tem a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Procedimento (Função que desaloca da memória um PChar)	Ponteiro para a Função (strdispose)

Este procedimento não tem um retorno.

Exemplo em pascal:

```

// estrutura
TStrDispose = procedure(VP_PChar: PChar); cdecl;

// procedimento
procedure StringDispose(VP_PChar: PChar); cdecl;
begin

```

```
if not Assigned(VP_PChar) then
  Exit;

if MemSize(VP_PChar) <= 0 then
  Exit;

  StrDispose(VP_PChar);
end;
```

1.6 – Processo de venda

Após ser feito o processo de conexão(tópico 1.4) e configuração com o Open-Tef, o PDV está liberado para realizar uma venda pelo Open-Tef, seguindo as etapas citadas abaixo.

Nessa versão os dados que o PDV deve manter para uma aprovação de venda e que poderão ser solicitados pelo Open-Tef de maneira automatizada, de acordo com a bandeira são:

Data da venda(tag 000C),
Hora da venda(tag 000D),
NSU da venda gerado pelo PDV(tag 000B),
Observação sobre a venda(tag 0040),
Desconto da venda(tag 00E6),
Valor da parcela (tag 0013),
Número de parcelas(tag 000F).

XML contendo os dados da venda, tendo como campos obrigatórios(tag 0017):

Código Produto, Descrição, Quantidade, Valor.
Operador do caixa(tag 0012)
Identificação/Número do caixa(tag 0011)
Número do cupom fiscal(tag 0010)

Link da validação da nota/cupom fiscal(tag 00E5)
 Valor total referente a produtos elegíveis no programa de vale-alimentação do PAT(tag 0014)
 Valor total referente a produtos elegíveis no programa de vale-refeição do PAT(tag 0015)
 Valor total referente a produtos elegíveis no programa de vale-cultura(tag 0016).

1.6.1 – Inicia uma transação – nome da função dentro da biblioteca “transacaocreate”

A função deve conter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a instância do OpenTef (ponteiro retornado pela biblioteca no tópico 1.4.1)	Ponteiro
Entrada	Tag de comando da transação (tag 000A para vendas)	PChar
Entrada	Identificador/Número do caixa	PChar
Entrada/Saída	Retorno do id único da transação (Informado pelo Open-Tef, esse ID é comum para toda referência a essa transação, gerada no início da transação, tópico 1.6.1)	PChar
Entrada	Tempo de espera (Tempo em que o sistema esperará o retorno do Open-Tef em milissegundos)	Inteiro

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_TransacaoCreate = function( VP_Tef: pointer; VP_Commando, VP_IdentificadorCaixa: PChar; var
VO_TransacaoID: PChar;           VP_TempoAguarda:   integer): integer; cdecl;
//aplicação
// exemplo de uma transação para aprovação de venda
VL_Erro := F_TransacaoCreate(F_Tef, PChar('000A'), VL_Caixa, VL_TransacaoID, 60000);
```

1.6.2 – Confirma transação

Após iniciar a transação (conforme o tópico 1.6.1) o sistema entrará em um loop esperando o retorno do status da transação (ver tabela no tópico 2.3) pelo Open-Tef e este processo é todo automatizado dentro do Open-Tef.

O Open-Tef fará todos os procedimentos necessários automaticamente para a análise da venda e retornará para o operador do PDV os passos a serem seguidos até a confirmação da venda.

1.6.3 – Consultar status da transação – nome da função dentro da biblioteca “transacaostatus”

Esta função deve conter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a instância do OpenTef (ponteiro retornado pela biblioteca no tópico 1.4.1)	Ponteiro
Entrada/Saída	Retorno do status da transação (Retornado pelo Open-Tef, esse código representa o status da transação conforme tópico 2.3)	Inteiro
Entrada/Saída	Retorno da chave da transação (Retornado pelo Open-Tef, essa chave é a identificadora da transação descrita no tópico 1.3.3)	PChar
Entrada	ID único da transação (Informado pelo Open-Tef, esse ID é comum para toda referência a essa transação, gerada no início da transação, tópico 1.6.1)	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_TransacaoStatus = function(VP_Tef: pointer; var VO_Status: integer; var VO_TransacaoChave: PChar;
VP_Transacaoid: PChar): integer; cdecl;

//aplicação
VL_Erro := F_TransacaoStatus(F_Tef, VL_Status, VL_TransacaoChave, VL_Transacaoid);
```

1.6.4 – Consultar descrição do erro – nome da função dentro da biblioteca “mensagemerro”

Esta função deve conter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Código do erro (Retornado pelo Open-Tef, código do erro conforme tópico 2.1)	Inteiro
Entrada/Saída	Retorno da descrição do erro (Retorno da descrição do código de erro passado)	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
//estrutura
F_Erro = function (VP_CodigoErro: Integer; var VO_RespostaMensagem: PChar): integer; cdecl;

//aplicação
F_Erro(VL_Erro, VL_DescricaoErro);
```

1.6.5 – Consultar descrição do erro da transação – nome da função dentro da biblioteca “transacaostatusdescricao”

Esta função deve conter a seguinte estrutura:

Parâmetros

Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a instância do OpenTef (ponteiro retornado pela biblioteca no tópico 1.4.1)	Ponteiro
Entrada/Saída	Descrição do erro (Descrição do erro da transação)	PChar
Entrada	ID único da transação (Informado pelo Open-Tef, esse ID é comum para toda referência a essa transação, gerada no início da transação, tópico 1.6.1)	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Se o parâmetro de retorno(descrição do erro) voltar vazio significa que nenhum erro foi encontrado na transação.

Exemplo em pascal:

```
//estrutura
F_TransacaoStatusDescricao = function(VP_Tef: pointer; var VO_Status: PChar; VP_Transacaoid: PChar): integer; cdecl;

//aplicação
VL_Erro := F_TransacaoStatusDescricao(F_Tef, VL_DescricaoErroTransacao, VL_Transacaoid);
```

1.6.6 – Captura dados do cliente

Após iniciar a transação (conforme o tópico 1.6.1) o sistema entrará em um loop esperando o retorno do status da transação (ver tabela no tópico 2.3) pelo Open-Tef e este processo é todo automatizado dentro do Open-Tef.

O Open-Tef realizará todos os processamentos necessários para a análise da venda e retornará para o operador do PDV os passos a serem seguidos, exemplo: Se for necessário capturar dados do cliente tal como senha, QR-Code, CPF na venda, o Open-Tef mandará uma mensagem para o operador do PDV solicitando a interação para a captura.

1.6.7 – Cancelamento de venda

Para solicitar o cancelamento de venda deve-se criar uma transação(descrito no tópico 1.6.1) com a tag de comando para o módulo(tag 00F4).

Após o cancelamento ser iniciado, o Open-Tef pedirá a chave da transação para a função solicita dados da transação(tópico 1.5.5).

1.6.8 – Desalocar transação da memória – nome do procedimento dentro da biblioteca “transacaofree”

Este procedimento deve conter a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada	Ponteiro para a instância do OpenTef (ponteiro retornado pela biblioteca no tópico 1.4.1)	Ponteiro
Entrada	ID único da transação (Informado pelo Open-Tef, esse ID é comum para toda referência a essa transação, gerada no início da transação, tópico 1.6.1)	PChar

Exemplo em pascal:

```
//estrutura
F_TransacaoFree = procedure (VP_Tef: pointer; VP_TransacaOID: PChar); cdecl;
//aplicação
F_TransacaoFree(F_Tef, VL_TransacaOID);
```

1.7 – Consultar saldo

Para consultar o saldo deve-se criar uma transação(conforme tópico 1.6.1) com a tag de comando para o módulo(tag 00F4). O restante do processo será feita de maneira automatizada pelo Open-Tef.

1.8 – Verificar versão do tef – nome da função dentro da biblioteca “versão”

Esta função consulta a versão do tef que está sendo utilizada no PDV e tem a seguinte estrutura:

Parâmetros		
Entrada/Saída	Descrição	Tipo de Dados
Entrada/Saída	Retorno da versão da biblioteca tef	PChar

O retorno da função devolve um inteiro, retorno diferente de 0 indica erro.

Exemplo em pascal:

```
// estrutura
function versao(var VO_Dados: PChar): integer; cdecl;
```

2 – Tabelas auxiliares

Abaixo segue as tabelas auxiliares para o uso do Open-Tef.

2.1 – Tabela de código de retorno (Tabela erro)

0	SUCESSO
1	ERRO
2	PORTA PINPAD NÃO INFORMADA
3	MODELO PINPAD NÃO INFORMADO
4	MODELO PINPAD NÃO SUPORTADO
5	FALHA NA CONEXÃO COM PINPAD
6	PINPAD RETORNOU FALHA(CONSULTA Nº DA FALHA NO RETORNO AUXILIAR)
7	DLL OU SO DO PINPAD NÃO ENCONTRADA
8	DLL OU SO DO PINPAD NÃO COMPATÍVEL(VERIFIQUE 32 OU 64 BITS)
9	HOST NÃO INFORMADO PARA O TEF
10	HOST NÃO COMPATÍVEL PARA O TEF
11	PORTA NÃO INFORMADA PARA O TEF
12	PORTA NÃO COMPATÍVEL PARA O TEF
13	VERSÃO_COMUNICAÇÃO NÃO INFORMADA PARA O TEF
14	VERSÃO_COMUNICAÇÃO NÃO COMPATÍVEL PARA O TEF
15	CHAVE NÃO INFORMADA PARA O TEF
16	CHAVE NÃO COMPATÍVEL PARA O TEF
17	VERSÃO DLL DO TEF NÃO COMPATÍVEL (SUGERE ATUALIZAÇÃO)

18	PACOTE COM A PRIMEIRA TAG DIFERENTE DE '0000'
19	ERRO O TAMANHO DO PACOTE NÃO PODE SER ZERO
20	ERRO O TAMANHO DO PACOTE DIFERENTE DO Tamanho DA MENSAGEM
21	ERRO OS DADOS DO PACOTE MAIOR QUE OS DADOS DA MENSAGEM
22	ERRO NÃO EXISTE PACOTE PARA A FUNÇÃO GET DA TAG
23	ERRO NÃO EXISTE PACOTE PARA A FUNÇÃO "TAGTOSTR"
24	NÃO FOI POSSÍVEL ENVIAR MENSAGEM
25	NÃO FOI POSSÍVEL CONECTAR AO SERVIDOR PARA ESCUTAR
26	NÃO FOI POSSÍVEL CONECTAR AO SERVIDOR ABERTURA SOCKET
27	ERRO NÃO EXISTE PACOTE PARA A FUNÇÃO ADD DA TAG
28	ERRO, PARÂMETRO NULO OU DIFERENTE DA ESTRUTURA DA TAG
29	ERRO TAG NÃO LOCALIZADA NO PACOTE DA FUNÇÃO GET DA TAG
30	DADOS DO PACOTE MAL FORMATADO OU INCOMPATÍVEL, IMPOSSÍVEL DE LER
31	CHAVE PUBLICA NÃO LOCALIZADA
32	PROBLEMA NAS TROCAS DE CHAVES NO CONECTAR
33	SOCKET DO SERVIDOR VINCULADO ESTA DESCONECTADO
34	LOGIN RECUSADO
35	SERVIDOR NÃO ESTA EM STATUS DE LOGADO
36	CHAVE NÃO LOCALIZADA PARA OPENTEF
37	SENHA DE ACESSO NÃO INFORMADO PARA O TEF
38	IP NÃO NÃO COMPATÍVEL PARA O TERMINAL
39	SENHA NÃO COMPATÍVEL PARA O TERMINAL
40	ERRO VALIDAÇÃO DO TERMINAL
41	CHAVE DO TERMINAL NÃO LOCALIZADA
42	TIPO DE PERMISSÃO NÃO INFORMADA PARA O TEF
43	REGISTRO EM DUPLICIDADE NO BANCO DE DADOS
44	ERRO NA INCLUSÃO DO REGISTRO
45	PERMISSÃO NÃO AUTORIZADO PARA ESTE PROCEDIMENTO
46	ERRO NA ALTERAÇÃO DO REGISTRO
47	DADOS INFORMADO NÃO PODE SER ZERO
48	TABELA SEM REGISTRO NO BANCO DE DADOS
49	DADOS CHAVES PARA O CADASTRO NÃO PODEM SOFRER ALTERAÇÃO
50	BIBLIOTECA DO PINPAD NÃO FOI POSSÍVEL CARREGAR
51	TIPO DE LEITURA DE CARTÃO NÃO SUPORTADO
52	CAMPO OBRIGATÓRIO NÃO PODE SER NULO
53	NÃO FOI POSSÍVEL ENVIAR A MENSAGEM DE RETORNO PARA O ID INFORMADO
54	NÃO FOI POSSÍVEL INICIALIZAR O MODULO
55	ERRO NA PESQUISA DA TABELA

56	NÃO FOI ENCONTRADO NENHUMA CONFIGURAÇÃO DE MENU DE CAIXA
57	ERRO NO FLUXO DAS TRANSAÇÕES DA TAG 0018
58	ERRO AO CRIAR A TRANSAÇÃO
59	ERRO AO PEGAR O STATUS DA TRANSAÇÃO
60	ERRO AO EXCLUIR UMA TRANSAÇÃO NO TEF_LIB
61	ERRO AO RECEBER A FUNÇÃO TRANSAÇÃO CREATE DO OPENTEF
62	ERRO NO PEDIDO DE CRIAÇÃO DE TRANSAÇÃO
63	BIN JÁ ESTAVA CADASTRADO PARA OUTRO MÓDULO
64	REGISTRO DE MÓDULO NÃO CARREGADO OU NÃO LOCALIZADO
65	NÃO FOI POSSÍVEL ATUALIZAR A TABELA DE BINS
66	CONEXÃO DO TEMPORIZADOR NÃO ENCONTRADA
67	A TRANSMISSÃO NÃO RESPONDIDA EM TEMPO HÁBIL
68	DADOS DA TRANSMISSÃO NÃO TEM COMANDO 00D1
69	A COMUNICAÇÃO FOI ABORTADA
70	Módulo config não carregado para incluir solicitação
71	NÃO FOI POSSÍVEL ATUALIZAR A TABELA DE MENUS
72	MENU JÁ ESTAVA CADASTRADO PARA OUTRO MÓDULO
73	MENU NÃO ESTÁ NA LISTA DE MENUS OFICIAIS HABILITADOS USE TAG INÍCIO "FF"
74	BIN inválido
75	Tag do botão do menu inválido
76	Texto do botão do menu inválido
77	pinpad_lib NÃO CARREGADO
78	COMANDO NÃO IMPLEMENTADO PELO PINPD
79	Não foi possível achar uma operadora para processar o cartão
80	Essa solicitação de aprovação não está em conformidade
81	ESSA CONEXÃO NÃO PERMITE APROVAÇÃO.
82	ERRO NA EXCLUSÃO DO REGISTRO
83	A CONEXÃO FOI DESCONECTADA INESPERADAMENTE
84	A OPERADORA/ADQUIRENTE EXIGE CRIPTOGRAFIA, MAS NÃO ENVIOU AS CHAVES
85	Erro de tempo em capturar a senha no pinpad
86	Erro ao cancelar uma transação
87	Erro na exclusão, TAG oficial exclusiva Open-Tef
88	TAG não é um número hexadecimal válido
89	Chave da transação não localizada
90	Autorização da venda não localizada
91	Código da loja não localizado
92	Desafio da chave não completado
93	ID do Terminal não informada para o OPENTEF

94	ID do Terminal não encontrado para o OPENTEF
95	Login negado pela Adquirente
96	CLIENTE DESCONECTADO DO SERVIDOR
97	Tarefa não encontrada para remoção
98	Chaves públicas não trocada
99	Conexão não encontrada
100	CONEXÃO NÃO ESTÁ ESTABELECIDA
101	Comando inválido para o OPENTEF
102	DADOS INFORMADOS DIFERENTE DOS DADOS DA TABELA NO BANCO DE DADOS
103	NÃO FOI POSSÍVEL CONECTAR AO SERVIDOR ABERTURA SOCKET COM EXCEÇÃO
104	NÃO FOI ENCONTRADA OS DADOS DO ARQUIVO
105	Erro insperado no servidor
106	Identificação NÃO LOCALIZADA
107	Identificação não informada
108	Loja já pertence a outra multi-loja
109	Chave de comunicação não informada para uma transação criptografa
110	O Modulo não foi carregado
111	O Modulo não está logado
112	A biblioteca Tef já foi inicializada
113	A função de retorno para o cliente não foi informada
114	A função de solicita dados do pdv não foi informada
115	A função de solicita dados da transação não foi informada
116	A função de impressão não foi informada
117	A função de mostra menu não foi informada
118	A função de mensagem ao operador não foi informada
119	ERRO AO PEGAR A TAG DA TRANSAÇÃO
120	Não é possível buscar essa tag pois ela é protegida
121	Não é possível realizar a conciliação, pois a versão está desatualizada
122	ESSA CONEXÃO NÃO PERMITE CONCILIAÇÃO
123	Não é possível enviar mais de uma conciliação para a mesma operadora
124	Erro insperado na lib do pinpad
125	Ponteiro do tef informado está nulo
126	Ponteiro do mcom informado está nulo
127	A função strDispose não foi informada

2.2 – TABELA DE TAGS

TAG	TAG DE DEFINIÇÃO	TIPO DE TAG	TIPO DE DADOS
0000	PACOTE	INÍCIO DO PACOTE	
0001	LOGIN	COMANDO	Comando sem Dados
0002	CHAVE DE IDENTIFICAÇÃO	DADOS	Texto
0003	HOST	DADOS	Texto
0004	PORTA	DADOS	Numérico
0005	VERSÃO DA DLL	DADOS	Texto
0006	VERSÃO DA MENSAGEM	DADOS	Texto
0007	SOLICITA CHAVE DE COMUNICAÇÃO	COMANDO	Texto
0008	CHAVE PUBLICA MODULO (hex)	DADOS	Texto
0009	CHAVE SIMÉTRICA(AES)	DADOS	Texto
000A	SOLICITA APROVAÇÃO DA TRANSAÇÃO	COMANDO	ID da transação gerada pelo PDV opcional, para controle interno
000B	NUMERO NSU	DADOS	<i>Texto(obrigatório)</i>
000C	DATA DA VENDA	DADOS	<i>Data(obrigatório)</i>
000D	HORA DA VENDA	DADOS	<i>Hora(obrigatório)</i>
000E	VALOR DA PARCELA	DADOS	<i>Numérico(obrigatório)</i>
000F	NUMERO PARCELA	DADOS	<i>Numérico(obrigatório)</i>
0010	CUPOM FISCAL	DADOS	Texto
0011	NUMERO CAIXA	DADOS	Numérico
0012	OPERADOR CAIXA	DADOS	Texto
0013	VALOR TOTAL DOS ITENS	DADOS	Numérico
0014	VALOR TOTAL ALIMENTAÇÃO	DADOS	Numérico
0015	VALOR TOTAL REFEIÇÃO	DADOS	Numérico
0016	VALOR TOTAL VALE-CULTURA	DADOS	Numérico
0017	XML DO CUPOM FISCAL	DADOS	Texto
0018	MENU	COMANDO	LADO OPENTEF S=PDV SOLICITANDO MENU S<>PDV INFORMANDO MENU SELECIONADO LADO PDV
0019	MENU - CARTÃO DIGITADO	MENU_PDV	Texto - Mensagem padrão (Cartão digitado)
001A	MENU - CARTÃO ALIMENTAÇÃO DIGITADO	MENU_PDV	Texto - Mensagem padrão (Cartão-Alimentação digitado)
001B	MENU - CARTÃO REFEIÇÃO DIGITADO	MENU_PDV	Texto - Mensagem Padrão (Cartão Refeição digitado)
001C	MENU - CARTÃO VALE CULTURA	MENU_PDV	Texto -

	DIGITADO		Mensagem padrão (Cartão Vale-cultura digitado)
001D	MENU - LEITURA DO CARTÃO NO PINPAD	MENU_PDV	Texto - Mensagem padrão (Cartão)
001E	MENU - CARTÃO ALIMENTAÇÃO NO PINPAD	MENU_PDV	Texto - Mensagem padrão (Cartão Alimentação)
001F	MENU - CARTÃO REFEIÇÃO NO PINPAD	MENU_PDV	Texto - Mensagem padrão (Cartão Refeição)
0020	MENU - CARTÃO VALE CULTURA NO PINPAD	MENU_PDV	Texto - Mensagem padrão (Cartão Vale Cultura)
0021	PEDIDO DE CONEXÃO(TROCA DE CHAVE RSA)	COMANDO	Comando sem Dados
0022	SOLICITA DADOS DA IDENTIFICAÇÃO	COMANDO	TEXTO
0023	CHAVE PUBLICA EXPOENTE (MÓDULOS, TEF...)	DADOS	TEXTO
0024	CHAVE PUBLICA MODULO(HEX) (MÓDULOS, TEF...)	DADOS	TEXTO
0025	ENVIO DE CHAVE SIMÉTRICA CONEXÃO ACEITA	COMANDO	Comando sem Dados
0026	RESPOSTA COM ERRO, PROBLEMA NA EXECUÇÃO DO COMANDO	COMANDO	VER TABELA DE CÓDIGO DE RETORNO
0027	CHAVE PUBLICA EXPOENTE	DADOS	Texto
0028	CHAVE PUBLICA TAMANHO (MÓDULOS, TEF ...)	DADOS	TEXTO
0029	CHAVE SIMÉTRICA(AES) CRIPTOGRAFADA	DADOS	TEXTO
002A	SOLICITAÇÃO DE DADOS NO PDV	COMANDO	TEXTO - Mensagem a exibir ("")
002B	MODULO EXCLUIR	COMANDO	Comando sem Dados
002C	MENSAGEM A SER MOSTRADA NO PDV	COMANDO	TEXTO
002D	TEXTO A SER IMPRESSO	COMANDO	IMPRESSÃO
002E	IMAGEM A SER MOSTRADA NO PDV "QR-CODE"	DADOS	DADOS
002F	BOTÃO "OK"	DADOS	TEXTO - TEXTO DO BOTÃO
0030	BOTÃO "CANCELAR"	DADOS	TEXTO - TEXTO DO BOTÃO
0031	BOTÃO "NÃO"	DADOS	TEXTO - TEXTO DO BOTÃO
0032	BOTÃO "SIM"	DADOS	TEXTO - TEXTO DO BOTÃO

0033	DADOS CAPTURADO DO PDV	DADOS	TEXTO
0034	ID ÚNICA DA TRANSAÇÃO	DADOS	TEXTO
0035	SENHA DO TERMINAL	DADOS	TEXTO
0036	BIN OU IIN	TEXTO	IIN OU BIN
0037	TIPO DE PERMISSÃO DO TERMINAL	DADOS	NUMÉRICO
0038	ABORTA	COMANDO	ABORTA O COMANDO OU TRANSAÇÃO EM ANDAMENTO
0039	LOJA INCLUIR	COMANDO	Comando sem Dados
003A	TABELA MODULO_CONF	DADOS	TEXTO
003B	CHAVE DO PDV	DADOS	CAMPO CHAVE NA TABELA PDV
003C	LOJA_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
003D			
003E	TABELA LOJA	DADOS	TEXTO
003F	LOJA ALTERAR	COMANDO	Comando sem Dados
0040	OBSERVAÇÃO SOBRE A VENDA	DADOS	TEXTO
0041	CHAVE DO MODULO_CONF	DADOS	CAMPO CHAVE NA TABELA MODULO_CONF
0042	TRANSAÇÃO	COMANDO	Contem os dados da transação
0043	PDV_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
0044	PDV INCLUIR	COMANDO	Comando sem Dados
0045	VALIDA CHAVE DO PDV	COMANDO	Comando sem Dados
0046	CÓDIGO VINDO PINPAD	COMANDO	CÓDIGO VINDO DO PINPAD(POR MODELO ENTÃO NÃO TRATADO)
0047	MOSTRA MENSAGEM PINPAD	COMANDO	Mensagem
0048	PINPAD SOLICITA TARJA CARTÃO	COMANDO	0= MAG 1=FUTURO MAG,CHIP,CLS
0049	EXCEÇÃO NO PINPAD	COMANDO	CÓDIGO ERRO PINPAD
004A	MENSAGEM DE ERRO DO PINPAD	DADOS	TEXTO
004B	PDV ALTERAR	COMANDO	Comando Sem Dados
004C	ID DE UM COMANDO	DADOS	NUMÉRICO
004D	RETORNO DO COMANDO EXECUTADO	DADOS	NUMÉRICO 0= SUCESSO 0<>= ERRO (VER CÓDIGO TABELA ERRO)
004E	TARJA MAGNÉTICA TK1	DADOS	TEXTO
004F	TARJA MAGNÉTICA TK2	DADOS	TEXTO
0050	TARJA MAGNÉTICA TK3	DADOS	TEXTO

0051	TEMPO DE ESPERA DO COMANDO	DADOS	NUMÉRICO EM MILISSEGUNDOS
0052	TAG INCLUIR	COMANDO	INCLUIR REGISTRO NA TABELA TAG
0053	PINPAD INCLUIR	COMANDO	Comando sem Dados
0054	PINPAD_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
0055	PINPAD ALTERAR	COMANDO	Comando sem Dados
0056	CONFIGURADOR_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
0057	CONFIGURADOR INCLUIR	COMANDO	Comando sem Dados
0058	CONFIGURADOR ALTERAR	COMANDO	Comando sem Dados
0059	VALIDA CHAVE DO CONFIGURADOR	COMANDO	Comando sem Dados
005A	PINPAD SOLICITA SENHA	COMANDO	TEXTO
005B	ÍNDICE MK NO PINPAD	DADOS	NUMÉRICO
005C	MENSAGEM PINPAD	DADOS	TEXTO MAX 32
005D	NUMERO MÍNIMO DA SENHA NO PINPAD	DADOS	NUMÉRICO
005E	NUMERO MÁXIMO DA SENHA NO PINPAD	DADOS	NUMÉRICO
005F	WK CRIPTOGRAFADA	DADOS	TEXTO
0060	SENHA CRIPTOGRAFADA	DADOS	TEXTO
0061	SENHA	DADOS	TEXTO
0062	PAN MASCARADO	DADOS	TEXTO
0063	ACEITA TODAS FUNÇÕES DE MODULO	DADOS	BOOLEANO
0064	MULTILOJA INCLUIR	COMANDO	Comando sem Dados
0065	MULTILOJA_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
0066	MULTILOJA EXCLUIR	COMANDO	Comando sem Dados
0067	MULTILOJA ALTERAR	COMANDO	Comando sem Dados
0068	MODULO PADRÃO	DADOS	TEXTO
0069	LOJA EXCLUIR	COMANDO	Comando sem Dados
006A	PINPAD EXCLUIR	COMANDO	Comando sem Dados
006B	PDV EXCLUIR	COMANDO	Comando sem Dados
006C	MODULO_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
006D	MODULO_CONF_FUNCAO_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
006E	TAG_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
006F	ADQUIRENTE_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
0070	PESQUISA VARIAS TABELAS	COMANDO	Comando Sem Dados (Carrega

			tabelas em Lote no Banco)
0071	VALIDA CHAVE MODULO_CONF	COMANDO	Comando sem Dados
0072	MODULO_CONF ALTERAR	COMANDO	Comando sem Dados
0073	MODULO_CONF INCLUIR	COMANDO	Comando sem Dados
0074	MODULO INCLUIR	COMANDO	Comando sem Dados
0075	MODULO ALTERAR	COMANDO	Comando sem Dados
0076	BIN_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
0077	BIN INCLUIR	COMANDO	Comando sem Dados
0078	BIN EXCLUIR	COMANDO	Comando sem Dados
0079	MODULO_CONF_FUNCAO INCLUIR	COMANDO	Comando sem Dados
007A	CRIA UMA TRANSAÇÃO	COMANDO	S=SOLICITANDO R=RETORNO
007B	MODULO_CONF_ID	DADOS	NUMÉRICO
007C	DATA DA TRANSAÇÃO NO SERVIDOR OPEN TEF	DADOS	NUMÉRICO
007D	MENSAGEM ESTRUTURADA	DADOS	MENSAGEM DO TIPO TMENSAGEM ESTRUTURADA
007E	MODULO_FUNCAO INCLUIR	COMANDO	Comando sem Dados
007F	MODULO_FUNCAO EXCLUIR	COMANDO	Comando sem Dados
0080	TABELA MULTILOJA	DADOS	TEXTO
0081	TABELA TAG	DADOS	TEXTO
0082	TABELA ADQUIRENTE	DADOS	TEXTO
0083	TABELA BIN	DADOS	TEXTO
0084	MULTILOJA_MODULO_CONF_FUNCAO HABILITADO	DADOS	BOOLEANO
0085	MODULO_CONF_FUNCAO ALTERAR	COMANDO	Comando sem Dados
0086	MODULO_CONF_FUNCAO HABILITADO	DADOS	BOOLEANO
0087	LOJA_MODULO_CONF_FUNCAO EXCLUIR	COMANDO	Comando sem Dados
0088	MULTILOJA_MODULO_CONF_FUNCAO EXCLUIR	COMANDO	Comando sem Dados
0089	LOJA_MODULO_CONF_FUNCAO HABILITADO	DADOS	BOOLEANO
008A	MODULO_CONF_FUNCAO EXCLUIR	COMANDO	Comando sem Dados
008B	MODULO_FUNC_ID	DADOS	NUMÉRICO
008C	INCLUI OU ATUALIZA TAG	COMANDO	Comando sem Dados
008D	TABELA PINPAD	DADOS	TEXTO
008E	TABELA PDV	DADOS	TEXTO
008F	TABELA CONFIGURADOR	DADOS	TEXTO
0090	TABELA MODULO	DADOS	TEXTO

0091	DOCUMENTO	DADOS	CPF = PESSOA FÍSICA CNPJ=PESSOA JURÍDICA
0092	TABELA MODULO_FUNCAO	DADOS	TEXTO
0093	TABELA MODULO_CONF_FUNCAO	DADOS	TEXTO
0094	TABELA MULTLOJA_MODULO_CONF	DADOS	TEXTO
0095	TABELA PESQUISA_MODULO	DADOS	TEXTO
0096	MULTILOJA_MODULO_CONF INCLUIR	COMANDO	Comando sem Dados
0097	MULTILOJA_MODULO_CONF_ID	DADOS	NUMÉRICO
0098	TABELA LOJA_MODULO_CONF_FUNCAO	DADOS	TEXTO
0099	MULTILOJA_MODULO_CONF EXCLUIR	COMANDO	Comando sem Dados
009A	MULTILOJA_MODULO_CONF ALTERAR	COMANDO	Comando sem Dados
009B	LOJA_MODULO_CONF_FUNCAO INCLUIR	COMANDO	Comando sem Dados
009C	TABELA MULTILOJA_FUNCAO	DADOS	TEXTO
009D	MULTILOJA_FUNCAO INCLUIR	COMANDO	Comando sem Dados
009E	MULTILOJA_FUNCAO_ID	DADOS	NUMÉRICO
009F	MULTILOJA_FUNCAO ALTERAR	COMANDO	Comando sem Dados
00A0	MULTILOJA_FUNCAO EXCLUIR	COMANDO	Comando sem Dados
00A1	MULTILOJA_FUNCAO HABILITADO	DADOS	BOOLEANO
00A2	TERMINAL TIPO	DADOS	PDV, CONFIGURADOR...
00A3	TERMINAL ID	DADOS	NUMÉRICO
00A4	TRANSAÇÃO STATUS	DADOS	NUMÉRICO
00A5	CNPJ	DADOS	TEXTO
00A6	ID ÚNICO GERADO PELO PDV NA TRANSAÇÃO	DADOS	NUMÉRICO
00A7	TABELA LOJA_MODULO_CONF	DADOS	TEXTO
00A8	LOJA_FUNCAO HABILITADO	DADOS	BOOLEANO
00A9	TABELA LOJA_FUNCAO	DADOS	TEXTO
00AA	LOJA_FUNCAO INCLUIR	COMANDO	Comando sem Dados
00AB	LOJA_FUNCAO_ID	DADOS	NUMÉRICO
00AC	LOJA_FUNCAO ALTERAR	COMANDO	Comando sem Dados
00AD	LOJA_FUNCAO EXCLUIR	COMANDO	Comando sem Dados
00AE	LOJA_MODULO_CONF INCLUIR	COMANDO	Comando sem Dados
00AF	LOJA_MODULO_CONF_ID	DADOS	NUMÉRICO
00B0	MULTILOJA_MODULO_CONF_FUNCAO ALTERAR	COMANDO	Comando sem Dados
00B1	LOJA_MODULO_CONF ALTERAR	COMANDO	Comando sem Dados
00B2	LOJA_MODULO_CONF EXCLUIR	COMANDO	Comando sem Dados
00B3	PINPAD_FUNCAO HABILITADO	DADOS	BOOLEANO
00B4	TABELA PINPAD_FUNCAO	DADOS	TEXTO

00B5	PINPAD_FUNCAO INCLUIR	COMANDO	Comando sem Dados
00B6	PINPAD_FUNCAO ALTERAR	COMANDO	Comando sem Dados
00B7	PINPAD_FUNCAO_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
00B8	PINPAD_FUNCAO EXCLUIR	COMANDO	Comando sem Dados
00B9	MODULO_FUNCAO ALTERAR	COMANDO	Comando sem Dados
00BA	MODULO_CONF EXCLUIR	COMANDO	Comando sem Dados
00BB	CONFIGURADOR EXCLUIR	COMANDO	Comando sem Dados
00BC	LOJA_MODULO_CONF_FUNCAO_ID	DADOS	NUMÉRICO
00BD	MULTILOJA_MODULO_CONF_FUNCAO_ID	DADOS	NUMÉRICO – 0 INDICA TODOS OS REGISTROS
00BE	PINPAD_MODULO ALTERAR	COMANDO	Comando sem Dados
00BF	LOJA_MODULO_CONF_FUNCAO ALTERAR	COMANDO	Comando sem Dados
00C0	PDV_FUNCAO HABILITADO	DADOS	BOOLEANO
00C1	TABELA PDV_FUNCAO	DADOS	TEXTO
00C2	PDV_FUNCAO INCLUIR	COMANDO	Comando sem Dados
00C3	PDV_FUNCAO_ID	DADOS	NUMÉRICO
00C4	PDV_FUNCAO ALTERAR	COMANDO	Comando sem Dados
00C5	PDV_FUNCAO EXCLUIR	COMANDO	Comando sem Dados
00C6	MODULO_FUNCAO HABILITADO	DADOS	BOOLEANO
00C7	TABELA PDV_MODULO	DADOS	TEXTO
00C8	PDV_MODULO INCLUIR	COMANDO	Comando sem Dados
00C9	PDV_MODULO_ID	DADOS	NUMÉRICO
00CA	MULTILOJA_MODULO_CONF_FUNCAO INCLUIR	COMANDO	Comando sem Dados
00CB	PDV_MODULO ALTERAR	COMANDO	Comando sem Dados
00CC	PDV_MODULO EXCLUIR	COMANDO	Comando sem Dados
00CD	SOLICITA TABELA DE BINS	COMANDO	Comando sem Dados
00CE	BIN OU IIN	DADOS	EM FORMATO DE TEXTO UM BIN POR LINHA
00CF	SOLICITA TABELA DE MENUS VENDA	COMANDO	Comando sem Dados
00D0	TABELA MULTILOJA_MODULO_CONF_FUNCAO	DADOS	TEXTO
00D1	COMUNICAÇÃO DE SOCKET	COMANDO	TEXTO
00D2	DADOS TRANSMITIDO NO SOCKET	DADOS	TEXTO
00D3	MENU – CANCELAMENTO DE TRANSAÇÃO	MENU-OPERACIONAL	TEXTO
00D4	SOLICITA TABELA DE MENUS OPERACIONAL	COMANDO	TEXTO
00D5	BOTÃO SELECIONADO	DADOS	TAG DO BOTÃO

00D6	ID DE CONEXÃO DO SOCKET DO CLIENTE	DADOS	TEXTO
00D7	ID DA TRANSMISSÃO DO SOCKET DO CLIENTE	DADOS	TEXTO
00D8	ID DA TAREFA NO OPEN TEF	DADOS	TEXTO
00D9	PAN	DADOS	TEXTO
00DA	TEXTO A SER MOSTRADO NO PDV	DADOS	TEXTO
00DB	TAG ALTERAR	COMANDO	Comando sem Dados
00DC	TAG EXCLUIR	COMANDO	Comando sem Dados
00DD	BOTOES	COMANDO	CONTÉM LISTA DE BOTOES
00DE	ADQUIRENTE INCLUIR	COMANDO	Comando sem Dados
00DF	ADQUIRENTE ALTERAR	COMANDO	Comando sem Dados
00E0	ADQUIRENTE EXCLUIR	COMANDO	Comando sem Dados
00E1	SOLICITAÇÃO DE DADOS DA TRANSAÇÃO	COMANDO	DADOS SOLICITADOS PELA OPERADORA PARA APROVAÇÃO REFERENTE A VENDA
00E2	ECO NA CONEXÃO DO SOCKET	COMANDO	TEXTO
00E3	TRANSAÇÃO CRIPTOGRAFADA	COMANDO	ESTRUTURA DE UMA MENSAGEM
00E4	CHAVE PUBLICA TAMANHO	COMANDO	TEXTO
00E5	LINK VALIDAÇÃO NOTA FISCAL	DADOS	TEXTO
00E6	VALOR DO DESCONTO	DADOS	NUMÉRICO
00E7	PAN CARTÃO DIGITADO (PAN BIN + CARTÃO)	DADOS	TEXTO
00E8	SENHA DIGITADA	DADOS	TEXTO
00E9	CPF	DADOS	TEXTO
00F0	MENU - CONSULTAR SALDO	MENU_OPERACIONAL	Texto - Mensagem padrão (Cartão digitado)
00F1	CHAVE DA TRANSAÇÃO	DADOS	ESTRUTURA DE UMA MENSAGEM
00F2	MODULO	DADOS	TEXTO
00F3	SOLICITA SALDO	COMANDO	TEXTO
00F4	COMANDO PARA SER EXECUTADO NO MODULO	COMANDO	TEXTO
00F5	MENU OPERACIONAL MOSTRAR	COMANDO	TEXTO
00F6	SOLICITA CANCELAMENTO DE VENDA	COMANDO	TEXTO
00F7	AUTORIZAÇÃO DA VENDA	DADOS	TEXTO
00F8	MENSAGEM VINDA DE FORA DO OPEN TEF	DADOS	TEXTO
00F9	CÓDIGO DA LOJA	DADOS	TEXTO
00FA	INCLUIR LOJA NO GERENCIAMENTO DA MULTILOJA	COMANDO	TEXTO
00FB	EXCLUIR LOJA NO GERENCIAMENTO DA	COMANDO	TEXTO

	MULTILOJA		
00FC	PINPAD MOSTRA IMAGEM	PINPAD_FUNC	TEXTO
00FD	OBRIGATÓRIO ATUALIZAR O TEF	DADOS	TEXTO
00FE	IMAGEM A SER MOSTRADA NO PINPAD	DADOS	TEXTO
00FF	IMAGEM MOSTRADA NO PINPAD COM EXITO	DADOS	BOOLEANO
0100	NOME DO CLIENTE	DADOS	TEXTO
0101	FIRMA DO CLIENTE	DADOS	TEXTO
0102	COMERCIO CREDENCIADO	DADOS	TEXTO
0103	SOLICITA DESAFIO DE CHAVE	COMANDO	TEXTO
0104	SOLICITA TAGS	COMANDO	TEXTO
0105	COMANDO PARA SER EXECUTADO NOS TERMINAIS CLIENTES	COMANDO	TEXTO
0106	DESAFIO DE CHAVE	DADOS	TEXTO
0107	CÓDIGO DO PDV	DADOS	TEXTO
0108	TERMINAL DE IDENTIFICAÇÃO	DADOS	TEXTO
0109	IDENTIFICAÇÃO DO ADQUIRENTE	DADOS	TEXTO
010A	OPCIONAL ATUALIZAR O TEF	DADOS	TEXTO
010B	DADOS DO ARQUIVO	DADOS	TEXTO
010C	SOLICITA ATUALIZAÇÃO DO TEF	COMANDO	TEXTO
010D	SOLICITA ARQUIVO PARA O OPENTEF	COMANDO	TEXTO
010E	CHAVE DE COMUNICAÇÃO (AES) CRIPTOGRAFADA (MÓDULOS, TEF ...) S	DADOS	TEXTO
010F	CHAVE DE COMUNICAÇÃO (AES) CRIPTOGRAFADA (MÓDULOS, TEF ...) R	DADOS	TEXTO
0110	TIPO DA CONEXÃO	DADOS	TEXTO S = SISTEMA C= CAIXA
0111	SOLICITA CHAVE PÚBLICA	COMANDO	TEXTO
0112	MENU - PAGAMENTO PIX	MENU_PDV	TEXTO
0113	SOLICITA CONCILIAÇÃO	COMANDO	TEXTO
0114	VERSÃO DA CONCILIAÇÃO	DADOS	TEXTO
0115	CAMPOS DA CONCILIAÇÃO	DADOS	TEXTO
0116	FILTRO DA CONCILIAÇÃO	DADOS	TEXTO
0117	DADOS DA CONCILIAÇÃO	DADOS	TEXTO
0118	NOME DA TABELA	COMANDO	TEXTO

0119	QUANTIDADE DE LINHA NA TABELA	DADOS	NUMÉRICO
011A	PAN (DIGITADO PELO PINPAD)	DADOS	TEXTO
011B	PINPAD MODELO	DADOS	TEXTO

2.3 – TABELA DE STATUS DE TRANSAÇÃO

CÓDIGO	STATUS DA TRANSAÇÃO
0	EFETIVADA
1	NEGADA
2	CANCELADA
3	PROCESSANDO
4	AGUARDANDO COMANDO
5	NÃO LOCALIZADA
6	INICIALIZADA
7	ERRO
8	ABORTADA
9	AGUARDANDO DADOS DO PDV

2.4 – TABELA DE STATUS DA CONEXÃO

CÓDIGO	STATUS DA TRANSAÇÃO
0	NÃO INICIALIZADO
1	DESCONECTADO
2	LINK
3	CHAVEADO
4	CHAVEADO ASSINADO
5	LOGADO

3 – MENSAGEIRIA

3.1 – ESTRUTURA DO PACOTE

Nome	Tipo	Tamanho
Tag(0000)	Hex	4
QT de Dígitos do tamanho (0..F)máximo 15 dígitos	Hex	1
Tamanho até 9999999999999999	Numérico	Variável
Dados(Tags concatenadas)	String	Variável

3.2 – TIPOS DE DADOS

Tipo	Definição
Texto	ASCII
Inteiro	Número sem separador decimal
Numérico	Ponto como separador decimal
Data	DD/MM/AAAA
Hora	HH:MM:SS
Dados	Hexadecimal
Impressão/ Mensagem	<p>
 = pula linha</p> <p>! = exceção para texto</p> <p>Uso o comando "
" para pular linha na impressão e o comando "!" para exceção em um texto veja o exemplo a baixo:</p> <p>Exemplo 01: Quero duas palavras "Texto" e "Texto2" uma em cada linha.</p> <p>Comando: " Texto
Texto2 "</p> <p>Impresso sai assim:</p> <p>Texto Texto2</p>

	<p>Exemplo 02: Quero duas palavras “Texto
” e “Texto2” uma em cada linha.</p> <p>Comando: ” Texto!

Texto2 “</p> <p>Impresso sai assim:</p> <p>Texto
 Texto2</p>
	<p>Exemplo 03: Quero duas palavras “Texto!
” e “Texto2” uma em cada linha.</p> <p>Comando: ” Texto!!

Texto2 “</p> <p>Impresso sai assim:</p> <p>Texto!
 Texto2</p>
Booleano	F=Falso ou T=Verdadeiro

3.2.1 – TAGS PADRONIZADAS

Tags oficiais usadas no OpenTef contém o intervalo de 0000 a FEFF

Tags proprietárias para ser usadas em projetos particulares terão o intervalo FF00 a FFFF

3.3 – PACOTES

As solicitações são divididas por pacotes individuais.

A primeira Tag é sempre indicativa para o tamanho do pacote(Máximo 15 dígitos(xF) 999 terabytes) **TIPO PACOTE**

A segunda Tag é destinada a alguma requisição ou função **TIPO COMANDO**
As Tags subsequentes são parâmetros da segunda Tag **TIPO DADOS**

3.3.1 – EXEMPLO DE PACOTE

LOGIN

TAG	TAG DEFINIÇÃO	TIPO
0000	PACOTE	
0001	LOGIN DO FRENTE DE CAIXA	Comando sem Dados
0002	CHAVE DO FRENTE DE CAIXA	Texto
0003	HOST	Texto
0004	PORTA	Numérico
0005	VERSÃO DA DLL “Tef_lib”	Texto
0006	VERSÃO DA MENSAGEM	Texto
0007	SOLICITA CHAVE SIMÉTRICA	Comando sem Dados
0008	CHAVE PUBLICA (hex)	Texto
0009	CHAVE SIMÉTRICA CRIPTOGRAFADA	Texto

Exemplo de montagem de Tags

“0000 2 86 0001 0 0002 2 20 12345678901234567890 0003 1 9 127.0.0.1 0004 1 4 1000 0005 1 5 1.2.3 0006 1 5 1.2.3”

TAG =0000 = PACOTE

QT = 2

TM =86

DADOS=00010000222012345678901234567890000319127.0.0.10004141000005151.2.30006151.2.2

TAG =0001=LOGIN

QT = 0

TM =

DADOS =

TAG =0002=Tag de Dados

QT = 2

TM = 20

DADOS = “12345678901234567890”

TAG =0003=Tag de Dados

QT = 1

TM = 9

DADOS = “127.0.0.1”

TAG =0004=Tag de Dados

QT = 1
TM = 4
DADOS = 1000

TAG =0005=Tag de Dados
QT = 1
TM = 5
DADOS = “1.2.3“

TAG =0006=Tag de Dados
QT = 1
TM = 5
DADOS = “1.2.2“

4 – VERSIONAMENTO

CÓDIGO	DESCRIÇÃO	DETALHAMENTO
1	FUNCIONALIDADE	Indica a funcionalidade do sistema
2	CORREÇÃO	Correções no código existe a necessidade de atualização
3	MELHORIAS	Melhorias no código sugere atualização

Exemplo

1.2.3

- 1 = Tipo
2 = Versão (atualização obrigatório)
3 = Compilação (atualização sugerida não obrigatória)