



POS PGWeb

Especificação de Integração

Versão 2.03 (29 mar 2018)

HISTÓRICO

Versão	Data	Autor	Descrição
1.00	17 jul2014	Vincent Bouchy	Primeira versão
1.01	4 ago 2014	Guilherme / Vincent	Liberação do primeiro kit de integração (pequenos ajustes)
1.02	5 dez 2014	Vincent Bouchy	Inclusão de funções de interface com o usuário
1.03	6 nov 2015	Guilherme	Alteração de códigos PTIRET_XXX.
1.04	20 set 2017	Guilherme	Criação da função PTI_ConnectionLoop. Criação de códigos PTIRET_XXX.
1.10	27 out 2017	Tuco Santana	Tradução de inglês para português.
1.11	15 nov 2017	Vincent Bouchy	Revisão.
1.12	23 nov 2017	Nishiyama	Inclusão do parâmetro pszPOS_Capabilities na função PTI_init. Criação de novos códigos PTIPAR_XXX. Inclusão de um novo código de status na função PTI_CheckStatus.
2.00	26 jan 2018	Rafael Massaia	Reestruturação das tags e valores de entrada e saída da biblioteca.
2.01	30 jan 2018	Rafael Massaia	Alteração dos valores das tags PWINFO_DATETIME e PWPTI_RESULT
2.02	07 fev 2018	Nishiyama	Inclusão da função PTI_PrnSymbolCode para impressão de QRCode e código de barra.
2.03	29 mar 2018	Guilherme	Inclusão das tags PWINFO_AUTRESPCODE, PWINFO_RCPTFULL, PWINFO_RCPTSHORT, PWINFO_INSTALLMDATE, PWINFO_LANGUAGE, PWINFO_TIPAMOUNT, PWINFO_BOARDINGTAX, PWINFO_CASHBACKAMT, PWINFO_DISCOUNTAMT, PWINFO_DUEAMNT, PWINFO_READJUSTEDAMNT, PWINFO_PRODUCTNAME, PWINFO_PNDVIRTMERCH, PWINFO_PNDAUTHSYST, PWINFO_PNDAUTEXTREF, PWINFO_PNDAUTLOCREF.

ÍNDICE

1. Sobre Este Documento.....	4
1.1. Escopo.....	4
1.2. Público	4
1.3. Contatos.....	4
1.4. Avisos.....	4
2. Introdução.....	5
3. Arquitetura.....	6
3.1. Integração direta com o <i>checkout</i>	6
3.2. Integração com o servidor de Automação Comercial	7
3.3. Terminal como <i>checkout</i>	7
4. Fluxo Transacional.....	9
5. API.....	11
5.1. Gerenciamento do terminal	11
PTI_Init.....	12
PTI_End	14
PTI_ConnectionLoop.....	15
PTI_CheckStatus	16
PTI_Disconnect	17
5.2. Operação sob controle da Automação Comercial.....	18
PTI_Display	19
PTI_WaitKey.....	20
PTI_ClearKey.....	22
PTI_GetData.....	23
PTI_StartMenu.....	25
PTI_AddMenuOption	26
PTI_ExecMenu	27
PTI_Beep.....	28
PTI_Print	29
PTI_PrnFeed.....	30
PTI_PrnSymbolCode	31
5.3. Pagamento.....	32
PTI_EFT_Start.....	34
PTI_EFT_AddParam	35
PTI_EFT_Exec	36
PTI_EFT_GetInfo	37
PTI_EFT_PrintReceipt.....	38
PTI_EFT_Confirm	39
5.4. Códigos de retorno	40
5.5. Redes adquirentes	41
5.6. TAG's de entrada e saída	42

1. Sobre Este Documento

1.1. Escopo

Este documento constitui a especificação técnica para integração de uma aplicação de Automação Comercial com a solução **POS PGWeb**.

1.2. Público

Este documento é essencialmente destinado a desenvolvedores de sistemas de Automação Comercial que desejem integrar seu sistema com a solução **POS PGWeb** para a realização de transações eletrônicas (TEF).

1.3. Contatos

O suporte ao desenvolvedor de sistema de Automação Comercial é prestado através do e-mail dev@ntk.com.br.

1.4. Avisos

As informações contidas neste documento estão sujeitas a alteração sem prévio aviso.

Pay&Go é uma marca registrada da **Erro! Nome de propriedade do documento desconhecido.**

Microsoft e *Windows* são marcas registradas pela Microsoft Corporation.

Copyright 2014-2018 © SETIS Automação e Sistemas Ltda. Todos os direitos reservados.

2. Introdução

A solução **POS PGWeb** é composta por um terminal de pagamento conectado à plataforma **Pay&Go Web** para realização de transações eletrônicas de pagamento ou de outros serviços disponibilizados pelas diversas entidades conectadas à plataforma.

O **POS PGWeb** pode operar de duas maneiras distintas:

- Autônomo: Toda a operação é realizada diretamente no terminal, sem acionamento externo, muito similarmente aos terminais fornecidos diretamente pelos adquirentes.
- Integrado com Automação Comercial: O terminal opera como terminal remoto escravo de uma aplicação de Automação Comercial, que o utiliza para captura de informações pertinentes à transação e aciona a operação de pagamento (ou outra) no momento desejado. A automação ainda pode decidir se e onde são impressos os comprovantes, além de agregar conteúdo a estes.

Este documento tem como propósito tratar exclusivamente deste segundo modo de operação, especificando o mecanismo de integração da Automação Comercial com o **POS PGWeb**.

3. Arquitetura

A figura a seguir ilustra a arquitetura da solução **POS PGWeb**:



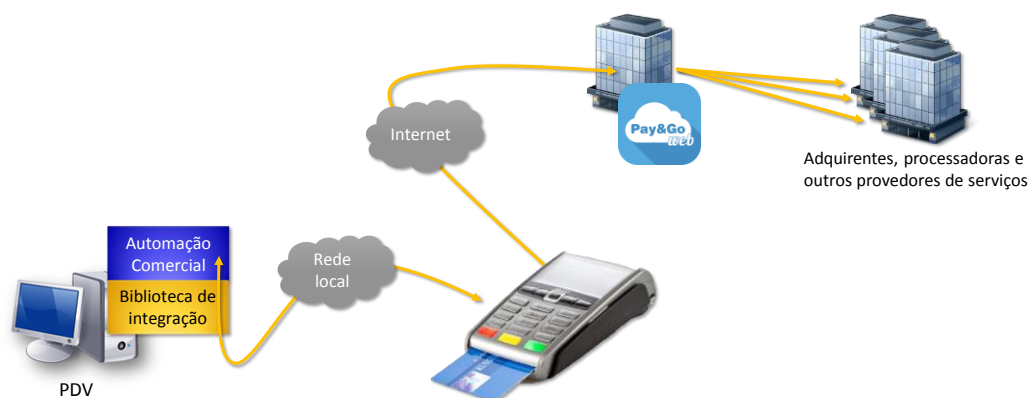
O **POS PGWeb** inclui uma camada de integração, responsável por:

- Comunicação com a Automação Comercial, através da rede local do estabelecimento ou internet;
- Recebimento de informações da transação enviadas pela Automação Comercial (essencialmente, tipo de transação, valor total e número de referência da transação);
- Envio dos resultados da transação para a Automação Comercial (aprova/negada, conteúdo do comprovante da transação, números de referências do emissor e adquirente, dados não sensíveis do cartão, entre outros).

Para que a Automação Comercial possa comunicar com o terminal de pagamento, uma biblioteca de integração é fornecida na forma de uma DLL Microsoft Windows. Esta biblioteca permite integração para as três arquiteturas descritas abaixo.

3.1. Integração direta com o *checkout*

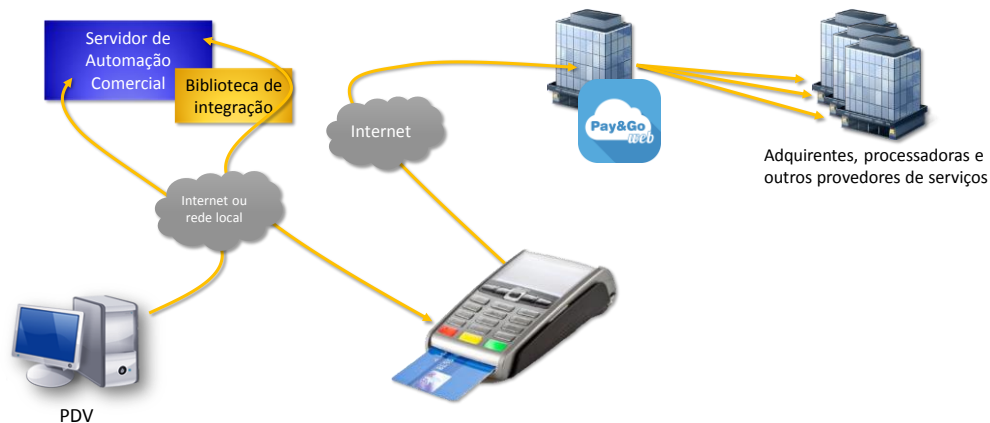
Nesta configuração um-para-um, a biblioteca de integração é utilizada diretamente pela aplicação de Automação Comercial executada no ponto de venda (PDV ou *checkout*).



3.2. Integração com o servidor de Automação Comercial

Nesta configuração, a biblioteca de integração é hospedada em um sistema servidor, que controla múltiplos pontos de venda. Este servidor pode estar instalado:

- Nas dependências do estabelecimento, para um estabelecimento com um número significativo de checkouts; ou
- Na nuvem, principalmente para aplicação de Automação Comercial Web.



3.3. Terminal como *checkout*

Nesta configuração, o terminal é utilizado como um *checkout*, propriamente dito. A biblioteca de integração é hospedada em um sistema servidor que controla múltiplos terminais com a finalidade de coletar dados que não sejam do cartão, e também para realizar transações com cartão.

Este modelo se aplica aos seguintes exemplos:

- Em um restaurante, para coletar pedidos e imprimir um recibo de pagamento prévio.
- Em um posto de combustível, para coletar o número da bomba e recuperar o valor a ser pago;
- Em soluções de *delivery*, com a finalidade de associar a transação ao pedido.

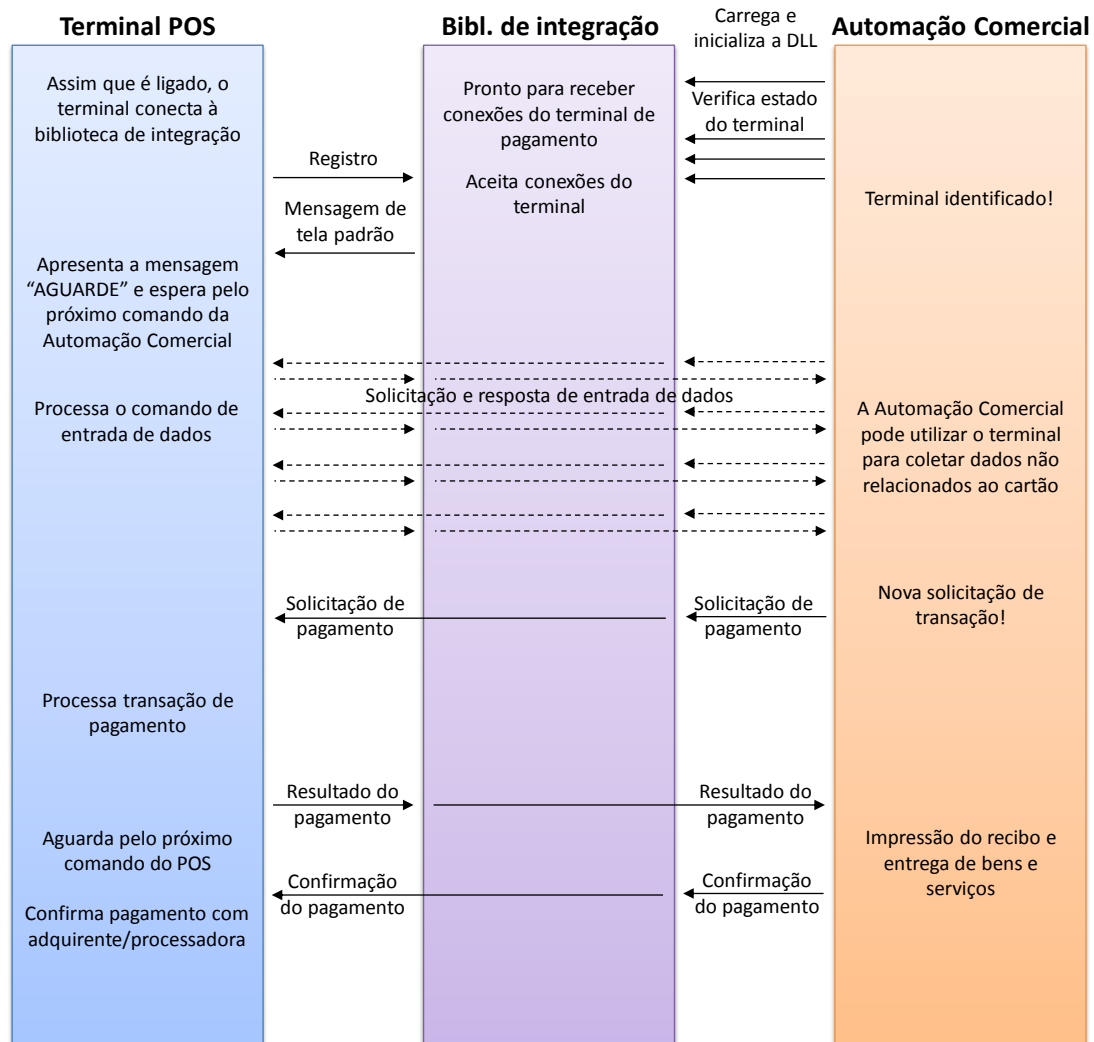
Este servidor pode ser instalado:

- Nas dependências do estabelecimento, tipicamente para operações *indoor*; ou
- Na nuvem, tipicamente para operações *outdoor*.



4. Fluxo Transacional

O diagrama abaixo ilustra a dinâmica de uma transação:



Um requerimento preliminar para transações ocorrerem é que tanto o terminal quanto a aplicação de Automação Comercial estejam ativos:

- Antes de qualquer terminal poder conectar, a aplicação de Automação Comercial deve carregar a biblioteca de integração, informando o número máximo de terminais a suportar.
- Tão logo o terminal é ligado, ele se conecta à biblioteca de integração e se torna disponível para transacionar.
- A qualquer momento, a Automação Comercial pode verificar o estado do terminal, se está on-line ou offline.
- A qualquer momento, a Automação Comercial pode desconectar um terminal.

O terminal é escravo da Automação Comercial e, desde que esteja on-line, continua aguardando novos comandos. Tão logo ele receba novos comandos, processa-os até sua conclusão ou falha e então retorna todas as informações resultantes para a Automação Comercial. Neste momento, a Automação Comercial deve também:

- **Caso a transação falhe:** refazer ou solicitar um método de pagamento alternativo;
- **Caso a transação seja bem-sucedida:** atualizar seus registros com a informação do pagamento, imprimir o recibo, entregar a mercadoria, e informar o estado final da transação para a biblioteca de integração.

5. API

Este capítulo descreve a interface de programação fornecida pela biblioteca de integração para a aplicação de Automação Comercial.

As funções da API podem ser divididas em três grupos distintos, conforme capítulos a seguir.

5.1. Gerenciamento do terminal

Estas funções administrativas permitem que a Automação Comercial:

- Inicialize e encerre o uso da biblioteca de integração;
- Verifique e altere o estado (on-line ou offline) dos terminais de pagamento.

Função	Descrição
PTI_Init	Inicializa a biblioteca de integração
PTI_End	Encerra o uso da biblioteca de integração
PTI_CheckStatus	Verifica o estado de determinado terminal
PTI_Disconnect	Desconecta determinado terminal

PTI_Init

Esta função configura a biblioteca de integração e deve ser a primeira a ser chamada pela Automação Comercial. A biblioteca de integração somente aceitará conexões do terminal de pagamento após sua chamada.

```
void PTI_Init ( LPCSTR pszPOS_Company, LPCSTR pszPOS_Version, LPCSTR
                pszPOS_Capabilities, LPCSTR pszDataFolder, UINT16 uiTCP_Port,
                UINT16 uiMaxTerminals, LPCSTR pszWaitMsg, UINT16 uiAutoDiscSec,
                PINT16 piRet)
```

Entrada:

- pszPOS_Company** Nome da empresa de Automação Comercial (final-nulo, até 40 caracteres e sem acentuação). Por exemplo, "KND SISTEMAS LTDA."
- pszPOS_Version** Nome e versão da aplicação de Automação Comercial (final-nulo, até 40 caracteres e sem acentuação). Por exemplo, "SUPERVENDAS v1.01".
- pszPOS_Capabilities** Capacidades da Automação (soma dos valores abaixo):
- 1: funcionalidade de troco/saque;
 - 2: funcionalidade de desconto;
 - 4: valor fixo, sempre incluir;
 - 8: impressão das vias diferenciadas do comprovante para Cliente/Estabelecimento;
 - 16: impressão do cupom reduzido.
 - 32: utilização de saldo total do voucher para abatimento do valor da compra.
- pszDataFolder** Caminho completo do diretório para armazenar dados e logs da biblioteca de integração.
Observação: O usuário do sistema operacional onde é executada a aplicação de Automação Comercial deve ter permissão de gravação nesse diretório.
- uiTCP_Port** Porta TCP à qual todos os terminais irão conectar.
Observação: esta porta deve estar habilitada para o recebimento de conexões através de qualquer firewall que estiver no caminho entre a aplicação de Automação Comercial e o terminal de POS.
- uiMaxTerminals** Número máximo de conexões simultâneas de terminais.
- pszWaitMsg** Mensagem a ser apresentada na tela de qualquer terminal imediatamente após se conectar. Veja **PTI_Display** para informações de formatação.
- uiAutoDiscSec** Tempo de ociosidade em segundos após o qual o terminal deve se desconectar da Automação Comercial quando opera sem alimentação externa, ou zero para nunca desconectar. Veja **PTI_Disconnect** para informações adicionais.

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK Operação bem-sucedida.
PTIRET_INVPARAM Parâmetro inválido informado à função.

PTIRET_SOCKETERRErro ao iniciar a escuta da porta TCP informada.
PTIRET_WRITEERRErro no uso do diretório informado.

PTI_End

Esta função deve ser a última função chamada pela Automação Comercial, quando finalizada ou antes de descarregar a biblioteca de integração.

Neste momento, a biblioteca de integração libera todos recursos alocados (portas TCP, processos, memória, etc.).

```
void PTI_End (void)
```

Entrada:

nenhuma

Saída:

nenhuma

Retorno:

nenhum

PTI_ConnectionLoop

Esta função permite que a Automação Comercial verifique quando um novo terminal se conectou e, se PTIRET_NEWCONN é retornado, recupere informações adicionais do equipamento.

```
void PTI_ConnectionLoop ( LPSTR pszTerminalId, LPSTR pszModel, LPSTR pszMAC,  
                          LPSTR pszSerNo, PINT16 piRet)
```

Entrada:

nenhuma

Saída:

pszTerminalId	Identificador único do terminal (final nulo, até 20 caracteres).
pszModel	Modelo do terminal (final nulo, até 20 caracteres).
pszMAC	Endereço MAC do terminal (final nulo, formato "XX:XX:XX:XX:XX:XX").
pszSerNo	Número serial do terminal (final nulo, até 25 caracteres).

Retorno (piRet):

PTIRET_NEWCONN Novo terminal conectado.
PTIRET_NONEWCONN Sem novas conexões recebidas.

PTI_CheckStatus

Esta função permite que a Automação Comercial verifique o status (on-line ou offline) de determinado terminal de pagamento e recupere informações adicionais do equipamento.

Cada terminal de pagamento recebe um único identificador lógico, que é configurado quando o terminal é instalado. Se a Automação Comercial controla mais de um terminal, ela deve ter registro de todos os identificadores e suas localizações, com a finalidade de poder enviar comandos para o terminal desejado.

```
void PTI_CheckStatus ( LPSTR pszTerminalId, PINT16 piStatus, LPSTR pszModel,
                      LPSTR pszMAC, LPSTR pszSerNo, PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo). Pode ser vazio se o número máximo de terminais suportado (informado em PTI_Init) for 1.

Saída:

pszTerminalId Identificador único do terminal (final nulo, até 20 caracteres).
piStatus Status do terminal, como descrito abaixo.
pszModel Modelo do terminal (final nulo até 20 caracteres).
pszMAC Endereço MAC do terminal (final nulo, formato "XX:XX:XX:XX:XX:XX")
pszSerNo Número de série do terminal (final nulo, até 25 caracteres).

Retorno (piRet):

PTIRET_OK Operação bem-sucedida.

Lista de possíveis status:

Nome	Valor	Descrição
PTISTAT_IDLE	0	Terminal está on-line e aguardando por comandos.
PTISTAT_BUSY	1	Terminal está on-line, porém ocupado processando um comando.
PTISTAT_NOCONN	2	Terminal está offline.
PTISTAT_WAITRECON	3	Terminal está off-line. A transação continua sendo executada e após sua finalização, o terminal tentará efetuar a reconexão automaticamente.

PTI_Disconnect

Esta função permite que a Automação Comercial desconecte um terminal de pagamento e o coloque em modo offline, seja imediatamente ou após algum tempo funcionando sem alimentação externa.

Para terminais móveis, permanecer on-line aumenta consideravelmente o consumo da bateria. Por este motivo é recomendado que a Automação Comercial defina um valor diferente de zero para o parâmetro **uiAutoDiscSec** de **PTI_Init**, ou chame essa função assim que o terminal conectar.

Após o terminal ficar offline, tão logo uma tecla é pressionada, este se conecta automaticamente novamente à Automação Comercial.

```
void PTI_Disconnect (LPCSTR pszTerminalId, UINT16 uiPwrDelay, PINT16 piRet)
```

Entrada:

pszTerminalId	Identificador único do terminal (final nulo).
uiPwrDelay	Se igual a zero, desconecta imediatamente o terminal, independentemente de sua fonte de energia. Se diferente de zero, representa o número máximo de segundos durante os quais o terminal permanecerá on-line enquanto estiver operando sem alimentação externa. O terminal não ficará offline enquanto estiver conectado a uma fonte de alimentação externa. Este valor sobrescreve o parâmetro uiAutoDiscSec de PTI_Init para este terminal específico.

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK	Operação bem-sucedida.
PTIRET_NOCONN	O terminal está offline.
PTIRET_BUSY.....	O terminal está ocupado processando outro comando.

5.2. Operação sob controle da Automação Comercial

Estas funções permitem à Automação Comercial utilizar a tela, o teclado e a impressora do terminal de forma remota para interagir com o usuário antes e depois da realização da transação de pagamento. O terminal deve previamente estar conectado ao servidor de Automação Comercial.

Função	Descrição
PTI_Display	Exibe texto na tela do terminal
PTI_WaitKey	Aguarda uma tecla ser pressionada no terminal
PTI_ClearKey	Limpa o buffer do teclado
PTI_GetData	Realiza a captura de dados no terminal
PTI_StartMenu	Inicia a definição de um menu para seleção pelo usuário
PTI_AddMenuOption	Adiciona um item ao menu
PTI_ExecMenu	Apresenta o menu ao usuário e aguarda a seleção
PTI_Beep	Emite um aviso sonoro no terminal
PTI_Print	Imprime texto na impressora do terminal
PTI_PrnFeed	Avança o papel na impressora

Nota: Caracteres acentuados não devem ser utilizados em tela ou impressora. Apenas caracteres na faixa entre 32 (20h, espaço) e 122 (7Ah, 'z') da tabela ASCII devem ser utilizados.

PTI_Display

Esta função apresenta uma mensagem na tela do terminal e retorna imediatamente.

A mensagem é apresentada a partir do canto superior esquerdo da tela, sendo 20 caracteres por linha, com quebra de linha identificada pelo caractere '\r' (retorno ao início da linha, código ASCII 13). Caracteres que ultrapassem as 20 colunas ou o número máximo de linhas são descartados. O número máximo de linhas suportado pode variar dependendo do modelo do terminal, entretanto o mínimo de quatro linhas é sempre suportado.

```
void PTI_Display (LPCSTR pszTerminalId, LPCSTR pszMsg, PINT16 piRet)
```

Entrada:

pszTerminalId	Identificador único do terminal (final nulo).
pszMsg	Mensagem a ser apresentada na tela do terminal (final nulo).

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK	Operação bem-sucedida.
PTIRET_INVPARAM	Parâmetro inválido passado à função.
PTIRET_NOCONN	O terminal está offline.
PTIRET_BUSY	O terminal está ocupado processando outro comando.

PTI_WaitKey

Esta função aguarda o pressionar de uma tecla no terminal e apenas retorna após uma tecla ser pressionada ou quando o tempo de espera se esgotar.

Importante: Esta função somente deve ser utilizada para captura isolada de teclas, não devendo ser sucessivamente chamada para captura de dados de entrada. Para este propósito, **PTI_GetData** deve ser utilizado.

```
void PTI_WaitKey (LPCSTR pszTerminalId, UINT16 uiTimeOutSec, PINT16 piKey,
                 PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo).
uiTimeOutSec Tempo de espera do usuário, em segundos. Se igual a zero, a função retorna imediatamente, somente informando que uma tecla foi pressionada caso tenha sido feito antes da chamada à função. (Captura de tecla buferizada.)

Saída:

piKey Identificador da tecla que foi pressionada, de acordo com a tabela abaixo (somente se o retorno da função for **PTIRET_OK**).

Retorno (piRet):

PTIRET_OK Operação bem-sucedida, uma tecla foi pressionada.
 PTIRET_NOCONN O terminal está offline.
 PTIRET_BUSY O terminal está ocupado processando outro comando.
 PTIRET_TIMEOUT Nenhuma tecla foi pressionada durante o período de tempo especificado.

Definição de tecla:

Nome	Valor (dec.)	Valor (hex.)	Valor (ASCII)	Tecla
PTIKEY_BACKSP	8	08	BS	[LIMPA] (amarelo)
PTIKEY_OK	13	0D	CR	[OK] (verde)
PTIKEY_CANCEL	27	1B	ESC	[CANCELA] (vermelho)
PTIKEY_HASH	35	23	#	Numeral
PTIKEY_00	37	25	%	Zero duplo
PTIKEY_ALPHA	38	26	&	[ALFA]
PTIKEY_STAR	42	2A	*	Asterisco
PTIKEY_DOT	46	2E	.	Ponto
PTIKEY_0	48	30	0	Numérico [0]
PTIKEY_1	49	31	1	Numérico [1]
PTIKEY_2	50	32	2	Numérico [2]
PTIKEY_3	51	33	3	Numérico [3]

PTIKEY_4	52	34	4	Numérico [4]
PTIKEY_5	53	35	5	Numérico [5]
PTIKEY_6	54	36	6	Numérico [6]
PTIKEY_7	55	37	7	Numérico [7]
PTIKEY_8	56	38	8	Numérico [8]
PTIKEY_9	57	39	9	Numérico [9]
PTIKEY_FUNC0	97	61	a	Tecla de função adicional
PTIKEY_FUNC1	98	62	b	Tecla de função adicional
PTIKEY_FUNC2	99	63	c	Tecla de função adicional
PTIKEY_FUNC3	100	64	d	Tecla de função adicional
PTIKEY_FUNC4	101	65	e	Tecla de função adicional
PTIKEY_FUNC5	102	66	f	Tecla de função adicional
PTIKEY_FUNC6	103	67	g	Tecla de função adicional
PTIKEY_FUNC7	104	68	h	Tecla de função adicional
PTIKEY_FUNC8	105	69	i	Tecla de função adicional
PTIKEY_FUNC9	106	6A	j	Tecla de função adicional
PTIKEY_FUNC10	107	6B	k	Tecla de função adicional
...

PTI_ClearKey

Esta função limpa o buffer de teclas pressionadas, para que a próxima chamada da função não considere qualquer tecla previamente pressionada.

Esta função retorna imediatamente.

```
void PTI_ClearKey (LPCSTR pszTerminalId, PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo).

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK Operação bem-sucedida.

PTIRET_NOCONN O terminal está offline.

PTIRET_BUSY O terminal está ocupado processando outro comando.

PTI_GetData

Esta função realiza a captura de um único dado em um terminal previamente conectado.

Esta função é bloqueante e somente retorna após a captura de dado ser bem-sucedida ou falhar.

```
void PTI_GetData (LPCSTR pszTerminalId, LPCSTR pszPrompt, LPCSTR pszFormat,
                 UINT16 uiLenMin, UINT16 uiLenMax, BOOL fFromLeft,
                 BOOL fAlpha, BOOL fMask, UINT16 uiTimeOutSec, LPSTR pszData,
                 UINT16 uiCaptureLine, PINT16 piRet)
```

Entrada:

pszTerminalId	Identificador único do terminal (final nulo).
pszPrompt	Mensagem de texto com final nulo a ser apresentada ao usuário, descrevendo a informação a ser solicitada. Utilize '\r' (código ASCII 13) para quebra de linha. Por exemplo: "VALOR DO SERVICO:".
pszFormat	Máscara de formatação com final nulo. Utilize '@' (arroba) para as posições de caracteres editáveis. Por exemplo: "@@.@@.@@,@" para um valor em centavos. Deve ser nulo (NULL) ou vazio para captura direta sem formatação.
uiLenMin	Número mínimo de caracteres.
uiLenMax	Número máximo de caracteres.
fFromLeft	TRUE (1) para iniciar a digitação da esquerda; FALSE (0) para iniciar a digitação da direita.
fAlpha	TRUE (1) para habilitar a entrada de caracteres não numéricos; FALSE (0) para permitir apenas caracteres numéricos. Nota: como a digitação de caracteres não numéricos em muitos terminais não é amigável, recomenda-se evitar o uso desse recurso sempre que possível.
fMask	TRUE (1) para mascarar os caracteres digitados com asterisco (tipicamente, para digitação de senha); FALSE (0) para mostrar os caracteres digitados.
uiTimeOutSec	Tempo máximo entre cada tecla pressionada, em segundos.
pszData	Valor inicial para um dado a ser editado com final nulo.
uiCaptureLine	Índice da linha da tela (iniciando em 1) onde a informação digitada deve ser apresentada. Caso a legenda da mensagem também for apresentada nessa linha, a informação digitada será exibida logo após a legenda; senão, será exibida iniciando na primeira coluna.

Saída:

pszData	Informação digitada com final nulo (somente caso a função retorne PTIRET_OK).
----------------	---

Retorno (piRet):

PTIRET_OK	Captura de dado bem-sucedida.
PTIRET_INVPARAM	Parâmetro inválido passado à função.
PTIRET_NOCONN	O terminal está offline.
PTIRET_BUSY	O terminal está ocupado processando outro comando.
PTIRET_TIMEOUT	Nenhuma tecla foi pressionada no tempo especificado.

PTIRET_CANCELUsuário pressionou a tecla [CANCELA].
PTIRET_SECURITYERRA função foi rejeitada por questões de segurança.

PTI_StartMenu

Esta função inicia a construção de um menu de opção para seleção pelo usuário.

Esta função retorna imediatamente.

```
void PTI_StartMenu (LPCSTR pszTerminalId, PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo).

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK Criação do menu iniciada.

PTIRET_NOCONN O terminal está offline.

PTIRET_BUSY O terminal está ocupado processando outro comando.

PTI_AddMenuOption

Esta função adiciona uma opção ao menu que foi criado através de **PTI_StartMenu**.

Esta função retorna imediatamente.

```
void PTI_AddMenuOption (LPCSTR pszTerminalId, LPCSTR pszOption, PINT16 piRet)
```

Entrada:

pszTerminalId	Identificador único do terminal (final nulo).
pszOption	Mensagem de texto com final nulo que descreve a opção a ser exibida no terminal (máximo: 18 caracteres).

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK	A opção foi adicionada ao menu.
PTIRET_INVPARAM	Parâmetro inválido passado à função.
PTIRET_NOCONN	O terminal está offline.
PTIRET_BUSY	O terminal está ocupado processando outro comando.

PTI_ExecMenu

Esta função exibe o menu de opções que foi criado através de **PTI_StartMenu** e **PTI_AddMenuOption** e identifica a seleção feita pelo usuário.

Esta função é bloqueante e somente retorna após a seleção de uma opção ou a ocorrência de um erro.

```
void PTI_ExecMenu ( LPCSTR pszTerminalId, LPCSTR pszPrompt, UINT16 uiTimeOutSec,  
                  PUINT16 puiSelection, PINT16 piRet)
```

Entrada:

pszTerminalId	Identificador único do terminal (final nulo).
pszPrompt	Mensagem de texto com final nulo a ser apresentada ao usuário no topo do menu (máximo: 20 caracteres). Por exemplo: "SELECIONE UMA OPCAO:". Caso NULL ou vazio, o menu é exibido a partir da primeira linha da tela.
uiTimeOutSec	Tempo máximo entre duas teclas pressionadas, em segundos.
puiSelection	Índice (iniciado em zero) da opção que deve estar pré-selecionada quando o menu for inicialmente exibido, fazendo com que esta opção seja selecionada se o usuário simplesmente pressionar [OK]. Caso puiSelection não seja uma opção válida, nenhuma é pré-selecionada.

Saída:

puiSelection	Índice (iniciado em zero) da opção que foi selecionada pelo usuário (somente se a função retornar PGWRET_OK).
---------------------	---

Retorno (piRet):

PTIRET_OK	Seleção do menu bem-sucedida.
PTIRET_INVPARAM	Parâmetro inválido passado à função.
PTIRET_NOCONN	O terminal está offline.
PTIRET_BUSY	O terminal está ocupado processando outro comando.
PTIRET_TIMEOUT	Nenhuma tecla foi pressionada durante o tempo especificado.
PTIRET_CANCEL	Usuário pressionou a tecla [CANCELAR].

PTI_Beep

Esta função emite um aviso sonoro no terminal.

Esta função retorna imediatamente.

```
void PTI_Beep (LPCSTR pszTerminalId, INT16 iType, PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo).
iType Tipo de aviso sonoro, de acordo com a tabela abaixo.

Saída:

nenhuma

Retorno (piRet):

PTIRET_OKOperação bem-sucedida.
PTIRET_INVPARAMParâmetro inválido passado à função.
PTIRET_NOCONNO terminal está offline.
PTIRET_BUSY.....O terminal está ocupado processando outro comando.

Tipos de aviso sonoro:

Nome	Valor	Descrição
PTIBEEP_OK	0	Sucesso
PTIBEEP_WARNING	1	Alerta
PTIBEEP_ERROR	2	Erro

PTI_Print

Esta função imprime uma ou mais linhas de texto na impressora do terminal e retorna imediatamente.

Até 40 caracteres por linha podem ser impressos, com quebras de linha identificadas pelo caractere '\r' (código ASCII 13). Caracteres além das 40 colunas serão descartados.

Um caractere de controle na primeira posição de uma linha indica a mudança da fonte do caractere utilizada para o texto da linha inteira. Caso o primeiro caractere de uma linha não é um caractere de controle, a fonte padrão é utilizada. Os caracteres de controle suportados são:

Caractere de controle	Código ASCII do caractere	Efeito
'\v'	11	Dobra a largura da fonte, consequentemente o número de colunas suportado é dividido por dois.

Caso ocorra erro durante a impressão, o código de erro correspondente é retornado por **PTI_PrnFeed**. **PTI_PrnFeed** deve ser chamada após uma ou mais chamadas a **PTI_Print**.

```
void PTI_Print (LPCSTR pszTerminalId, LPCSTR pszText, PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo).
pszText Texto a ser impresso (final nulo).

Saída:

nenhuma

Retorno (piRet):

PTIRET_OKOperação bem-sucedida.
PTIRET_INVPARAMParâmetro inválido passado à função.
PTIRET_NOCONNO terminal está offline.
PTIRET_BUSY.....O terminal está ocupado processando outro comando.
PTIRET_NOTSUPPORTEDFunção não suportada pelo terminal.

PTI_PrnFeed

Esta função avança algumas linhas do papel da impressora, para permitir que o usuário destaque o recibo.

```
void PTI_PrnFeed (LPCSTR pszTerminalId, PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo).

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK Operação bem-sucedida.
PTIRET_INVPARAM Parâmetro inválido passado à função.
PTIRET_NOCONN O terminal está offline.
PTIRET_BUSY O terminal está ocupado processando outro comando.
PTIRET_PRINTERR Erro na impressora.
PTIRET_NOPAPER Impressora sem papel.
PTIRET_NOTSUPPORTED Função não suportada pelo terminal.

PTI_PrnSymbolCode

Esta função imprime um código de barras ou QR code na impressora do terminal.

```
void PTI_PrnSymbolCode (LPCSTR pszTerminalId, LPCSTR pszMsg, INT16 iSymbology,  
                        PINT16 piRet);
```

Entradas:

pszTerminalId	Identificador único do terminal (final nulo).
pszMsg	Código a ser impresso.
iSymbology	Tipo de código, conforme tabela abaixo.

Saídas:

não há.

Retorno (piRet):

PTIRET_OK	Operação bem sucedida.
PTIRET_INVPARAM	Parâmetro inválido passado à função.
PTIRET_INTERNALERR.....	Erro interno da biblioteca de integração.
PTIRET_NOCONN	O terminal está offline.
PTIRET_BUSY.....	O terminal está ocupado processando outro comando.

Tabela de tipos de código:

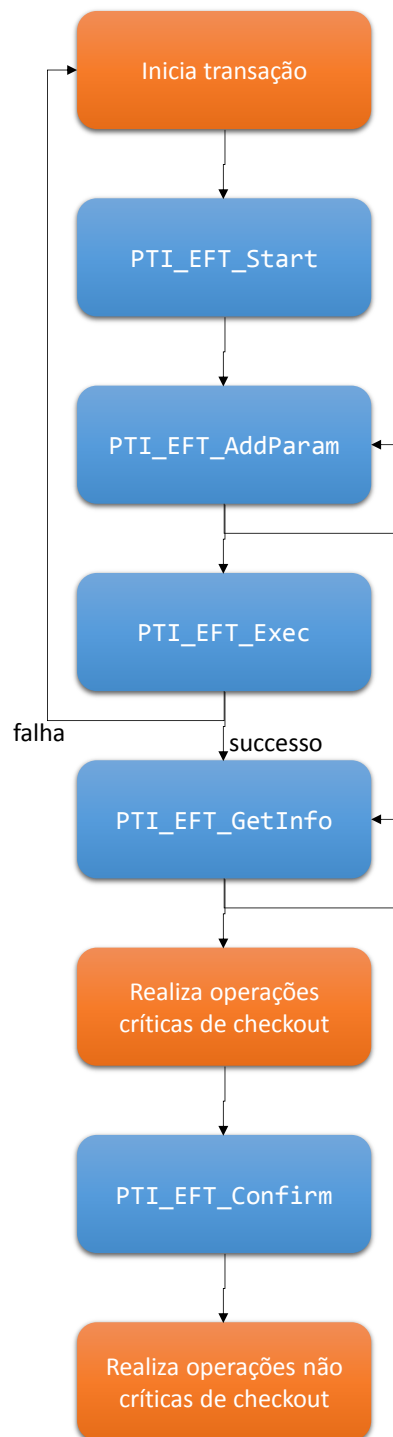
Nome	Valor	Descrição
CODESYMB_128	2	Código de barras padrão 128. Pode-se utilizar aproximadamente 31 caracteres alfanuméricos.
CODESYMB_ITF	3	Código de barras padrão ITF. Pode-se utilizar aproximadamente 30 caracteres alfanuméricos.
CODESYMB_QRCODE	4	QR Code. Com aceitação de aproximadamente 600 caracteres alfanuméricos.

5.3. Pagamento

Estas funções permitem que a Automação Comercial realize um pagamento através do terminal, ou qualquer outro tipo de transação suportado pelo terminal.

Função	Descrição
PTI_EFT_Start	Inicia uma nova transação.
PTI_EFT_AddParam	Define um parâmetro para a próxima transação.
PTI_EFT_Exec	Realiza uma transação.
PTI_EFT_GetInfo	Recupera os dados resultantes da última transação realizada.
PTI_EFT_PrintReceipt	Imprime os recibos da última transação realizada na impressora do terminal.
PTI_EFT_Confirm	Confirma ou desfaz a última transação realizada.

Abaixo está o diagrama que ilustra a sequência inteira das chamadas para uma transação completa:



PTI_EFT_Start

A Automação Comercial deve chamar esta função para iniciar qualquer nova transação. Esta função retorna imediatamente.

```
void PTI_EFT_Start (LPCSTR pszTerminalId, INT16 iOper, PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo).
iOper Tipo de transação, de acordo a tabela abaixo.

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK Operação bem-sucedida.
PTIRET_INVPARAM Parâmetro inválido passado à função.
PTIRET_NOCONN O terminal está offline.
PTIRET_BUSY O terminal está ocupado processando outro comando.

Lista dos tipos de transações:

Nome	Valor	Descrição
PWOPER_SALE	21h	Pagamento de mercadorias ou serviços.
PWOPER_ADMIN	20h	Qualquer transação que não seja um pagamento (estorno, pré-autorização, consulta, relatório, reimpressão de recibo, etc.).
PWOPER_SALEVOID	22h	Estorna uma transação de venda que foi previamente realizada e confirmada.

PTI_EFT_AddParam

A Automação Comercial deve chamar esta função iterativamente após **PTI_EFT_Start** para definir todos os parâmetros disponíveis para a transação.

Esta função retorna imediatamente.

```
void PTI_EFT_AddParam ( LPCSTR pszTerminalId, INT16 iParam, LPCSTR pszValue,  
                        PINT16 piRet)
```

Entrada:

pszTerminalId	Identificador único do terminal (final nulo).
iParam	Identificador do parâmetro, de acordo com o capítulo " TAG's de entrada e saída ".
pszValue	Valor do parâmetro (final nulo).

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK	Operação bem-sucedida.
PTIRET_INVPARAM	Parâmetro inválido passado à função.
PTIRET_NOCONN	O terminal está offline.
PTIRET_BUSY.....	O terminal está ocupado processando outro comando.

PTI_EFT_Exec

Esta função efetua de fato a transação, utilizando os parâmetros que foram previamente definidos através de **PTI_EFT_AddParam**.

Esta função é bloqueante, e somente retorna após a conclusão (ou falha) da transação.

```
void PTI_EFT_Exec (LPCSTR pszTerminalId, PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo).

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK Operação bem-sucedida (para venda, significa transação aprovada).
PTIRET_INVPARAM Parâmetro inválido passado à função.
PTIRET_NOCONN O terminal está offline.
PTIRET_BUSY O terminal está ocupado processando outro comando.
PTIRET_EFTERR A transação foi realizada, entretanto falhou.

PTI_EFT_GetInfo

A Automação Comercial deve chamar esta função iterativamente para recuperar os dados relativos à transação que foi realizada (com ou sem sucesso) pelo terminal.

Esta função retorna imediatamente.

```
void PTI_EFT_GetInfo ( LPCSTR pszTerminalId, INT16 iInfo, UINT16 uiBufLen,  
                      LPSTR pszValue, PINT16 piRet)
```

Entrada:

pszTerminalId	Identificador único do terminal (final nulo).
iInfo	Identificador da informação a ser obtida, conforme o capítulo " TAG's de entrada e saída ".
uiBufLen	Tamanho (em bytes) do buffer referenciado pelo ponteiro pszValue .

Saída:

pszValue	Informação recuperada (final nulo).
-----------------	-------------------------------------

Retorno (piRet):

PTIRET_OK	Operação bem-sucedida, informação retornada.
PTIRET_INVPARAM	Parâmetro inválido passado à função.
PTIRET_BUFOVRFLW	O tamanho do dado é maior que uiBufLen .
PTIRET_NOCONN	O terminal está offline.
PTIRET_BUSY	O terminal está ocupado processando outro comando.
PTIRET_NODATA	Informação não disponível.

PTI_EFT_PrintReceipt

Esta função faz com que o terminal imprima o comprovante da última transação realizada.

A Automação Comercial pode optar por:

- Utilizar esta função para imprimir uma ou ambas as vias (estabelecimento e/ou portador do cartão) do comprovante de pagamento;
- Recuperar o conteúdo do comprovante através de **PTI_EFT_GetInfo** e:
 - Imprimir uma ou ambas as vias em uma impressora dedicada;
 - Enviar a cópia do portador do cartão por e-mail ou outro tipo de mensageria.

Nota: a via do estabelecimento deve sempre ser impressa quando PWINFO_CHOLDVERIF (recuperado através de **PTI_EFT_GetInfo**) indicar que a assinatura do portador do cartão é requerida.

```
void PTI_EFT_PrintReceipt (LPCSTR pszTerminalId, INT16 iCopies, PINT16 piRet)
```

Entrada:

pszTerminalId	Identificador único do terminal (final nulo).
iCopies	Soma dos valores da tabela abaixo.

Saída:

pszValue	Informação recuperada (final nulo).
-----------------	-------------------------------------

Retorno (piRet):

PTIRET_OK	Bem-sucedida, impressão iniciada.
PTIRET_INVPARAM	Parâmetro inválido passado à função.
PTIRET_NOCONN	O terminal está offline.
PTIRET_BUSY	O terminal está ocupado processando outro comando.
PTIRET_NODATA	Não há recibo a ser impresso.
PTIRET_PRINTERR	Erro na impressora.
PTIRET_NOPAPER	Impressora sem papel.

Identificadores da cópia do recibo:

Nome	Valor	Descrição
PTIPRN_MERCHANT	1	Via do estabelecimento
PTIPRN_CHOLDER	2	Via do portador do cartão

PTI_EFT_Confirm

Qualquer transação financeira bem-sucedida (**PTI_EFT_Exec** retorna **PTIRET_OK**) deve ser confirmada pela Automação Comercial através desta função para assegurar a integridade da transação entre todas as partes (Automação Comercial e registro fiscais, terminal, adquirente, emissor e portador do cartão).

Múltiplas transações podem ser realizadas simultaneamente por diversos terminais, entretanto, para cada terminal, a transação deve ser confirmada antes de outra ser iniciada. Em cada momento, somente pode haver no máximo uma única transação pendente para cada terminal.

Para minimizar cenários de desfazimento, é recomendável que a Automação Comercial confirme a transação tão logo seja possível. Caso **PTI_EFT_Exec** retorne **PTIRET_OK** e a Automação Comercial não confirmar a transação imediatamente, esta deve ser armazenada em memória não volátil (arquivo) com todas as informações necessárias para confirmar ou desfazer a transação em caso de queda de energia que ocorra após esse momento.

Eventos que podem levar a um desfazimento da transação são:

- Falha na impressora (quando a assinatura do portador do cartão for requerida);
- Mercadoria não pode ser entregue (mecanismo do dispensador falha ou equivalente);
- Falta de energia (portador do cartão utilizou um método de pagamento alternativo antes da volta da energia).

```
void PTI_EFT_Confirm (LPCSTR pszTerminalId, INT16 iStatus, PINT16 piRet)
```

Entrada:

pszTerminalId Identificador único do terminal (final nulo).
iStatus Status final da transação, conforme detalhado abaixo.

Saída:

nenhuma

Retorno (piRet):

PTIRET_OK Confirmação realizada.
PTIRET_INVPARAM Parâmetro inválido passado à função.

Lista de possíveis status final para a transação:

Nome	Valor	Descrição
PTICNF_SUCCESS	1	Transação confirmada.
PTICNF_PRINTERR	2	Erro na impressora, desfazer a transação.
PTICNF_DISPFail	3	Erro com o mecanismo dispensador, desfazer a transação.
PTICNF_OTHERERR	4	Outro erro, desfazer a transação.

5.4. Códigos de retorno

A tabela a seguir define os códigos de retorno existentes para todas as funções.

Nome	Valor	Descrição
PTIRET_OK	0	Operação bem-sucedida.
PTIRET_INVPARAM	-2001	Parâmetro inválido informado à função.
PTIRET_NOCONN	-2002	O terminal está offline.
PTIRET_BUSY	-2003	O terminal está ocupado processando outro comando.
PTIRET_TIMEOUT	-2004	Usuário falhou ao pressionar uma tecla durante o tempo especificado.
PTIRET_CANCEL	-2005	Usuário pressionou a tecla [CANCELAR].
PTIRET_NODATA	2006	Informação requerida não disponível.
PTIRET_BUFOVRFLW	-2007	Dados maiores que o tamanho do buffer fornecido.
PTIRET_SOCKETERR	-2008	Impossibilitado de iniciar escuta das portas TCP especificadas.
PTIRET_WRITEERR	-2009	Impossibilitado de utilizar o diretório especificado.
PTIRET_EFERR	-2010	A operação financeira foi completada, porém falhou.
PTIRET_INTERNALERR	-2011	Erro interno da biblioteca de integração.
PTIRET_PROTOCOLERR	-2012	Erro de comunicação entre a biblioteca de integração e o terminal.
PTIRET_SECURITYERR	-2013	A função falhou por questões de segurança.
PTIRET_PRINTERR	-2014	Erro na impressora.
PTIRET_NOPAPER	-2015	Impressora sem papel.
PTIRET_NEWCONN	-2016	Novo terminal conectado.
PTIRET_NONEWCONN	-2017	Sem recebimento de novas conexões.
PTIRET_NOTSUPPORTED	-2057	Função não suportada pelo terminal.
PTIRET_CRYPTERR	-2058	Erro na criptografia de dados (comunicação entre a biblioteca de integração e o terminal).

5.5. Redes adquirentes

A tabela a seguir define os códigos das redes adquirentes que devem ser utilizados pelas automações comercial.

Código	Descrição
BIN	First Data
CIELO	Cielo
CONDUCTOR	Conductor
CREDSYSTEM	CredSystem
CTF	CTF
DMCARD	DM Card
FDCORBAN	First Data (Correspondente Bancário)
FILLIP	Fillip
GAX	Gax
GETNET	Getnet / Santander
GLOBAL	Global Payments
INFOCARDS	Infocards
REDE	Rede
RV	RV
STONE	Stone
TICKETLOG	TicketLog
VCMAIS	VC+
VERO	Banrisul / Vero

5.6. TAG's de entrada e saída

A tabela a seguir define todas as tag's de entrada e saída de integração

Nome	Valor	Tamanho	Entrada	Saída	Descrição
PWINFO_OPERATION	02h	b1		X	Transação que foi realizada: "00" – sem definição "01" – Venda (pagamento) "02" – Administrativa (geral) "04" – Estorno de venda
PWINFO_MERCHCNPJCPF	1Ch	ans..14	X	X	CNPJ (ou CPF) do Estabelecimento / Ponto de captura
PWINFO_TOTAMNT	25h	n..12	X	X	Valor da transação, em centavos. Este parâmetro é mandatório para transações de venda (PTITRN_SALE) e de estorno (PTITRN_SALEVOID).
PWINFO_CURRENCY	26h	b..2	X	X	Código da moeda, conforme o padrão internacional ISO4217 (986 para Real, 840 para Dólar americano). Este parâmetro é requerido sempre que PWINFO_TOTAMNT for informado.
PWINFO_FISCALREF	28h	ans..12	X	X	Número da fatura (final nulo). Este parâmetro é opcional.
PWINFO_CARDTYPE	29h	n..3	X	X	Modalidade da transação do cartão: 1: crédito; 2: débito; 4: voucher; 8: private label; 16: frota; 128: outros.
PWINFO_PRODUCTNAME	2Ah	ans..20		X	Nome/tipo do produto utilizado, na nomenclatura do Provedor.
PWINFO_REQNUM	32h	n..10		X	Identificador único da transação (gerado pelo terminal).

PWINFO_AUTHSYST	35h	ans..20	X	X	Nome da adquirente/processadora que a transação será/foi processada. Código da Rede Adquirente, conforme descrito no item 5.5. Caso este campo seja preenchido, a transação será realizada diretamente na rede adquirente especificada.
PWINFO_VIRTMERCH	36h	n..9		X	Identificador da afiliação utilizada para o sistema de gerenciamento do terminal.
PWINFO_AUTMERCHID	38h	ans..50		X	Identificador do estabelecimento para a adquirente.
PWINFO_FINTYPE	3Bh	n..3	X	X	Modalidade de financiamento da transação: 1: à vista; 2: parcelado pelo Emissor; 4: parcelado pelo Estabelecimento; 8: pré-datado; 16: crédito emissor; 32: Prédado parcelado.
PWINFO_INSTALLMENTS	3Ch	b4	X	X	Quantidade de parcelas, para transações parceladas
PWINFO_INSTALLMDATE	3Dh	6	X	X	Data de vencimento do pré-datado, ou da primeira parcela. Formato "DDMMMAA".
PWINFO_AUTRESPCODE	47h	ans..3		X	Caso a transação chegue ao sistema autorizador, esse é o código de resposta do mesmo (bit39 da mensagem ISO8583).
PWINFO_RESULTMSG	42h	ans..80		X	Mensagem de texto que descreve o resultado da transação (sucesso ou falha).
PWINFO_AUTLOCREF	44h	ans..50		X	Identificador único da transação (gerado pelo sistema de gerenciamento do terminal).

PWINFO_AUTEXTREF	45h	ans..50		X	Identificador único da transação (gerado pela adquirente/processadora).
PWINFO_AUTHCODE	46h	ans..6		X	Código de autorização (gerado pelo emissor).
PWINFO_DISCOUNTAMT	49h	ans..12	x	x	Valor do desconto concedido pelo Provedor, considerando PWINFO_CURREXP , já deduzido em PWINFO_TOTAMNT .
PWINFO_CASHBACKAMT	4Ah	ans..12	x	x	Valor do saque/troco, considerando PWINFO_CURREXP , já incluído em PWINFO_TOTAMNT .
PWINFO_CARDNAME	4Bh	ans..12		X	Nome do cartão ou emissor.
PWINFO_BOARDINGTAX	4Dh	ans..12	X	X	Valor da taxa de embarque, considerando PWINFO_CURREXP , já incluído em PWINFO_TOTAMNT .
PWINFO_TIPAMOUNT	4Eh	ans..12	X	X	Valor da taxa de serviço (gorjeta), considerando PWINFO_CURREXP , já incluído em PWINFO_TOTAMNT .
PWINFO_RCPTMERCH	53h	ans..32000		X	Comprovante – via do estabelecimento
PWINFO_RCPTCHOLDER	54h	ans..32000		X	Comprovante – via do portador
PWINFO_RCPTFULL	52h	ans..32000		X	Comprovante – via completa
PWINFO_RCPTCHSHORT	55h	ans..32000		X	Comprovante – via reduzida
PWINFO_TRNORIGDATE	57h	n6	X		Data da transação original. Este campo é utilizado para transações de cancelamento. Formato DDMMAA.
PWINFO_TRNORIGNSU	58h	ans..50	X		Número de referência da transação original (atribuído pela adquirente/processadora). Este parâmetro é mandatório para transações de estorno (PTITRN_SALEVOID).

PWINFO_TRNORIGAUTH	62h	ans..6	X		Código de autorização da transação original. Este campo é utilizado para transações de cancelamento.
PWINFO_LANGUAGE	6Ch	1	X		Idioma a ser utilizado para a interface com o cliente: 0: Português 1: Inglês 2: Espanhol
PWINFO_DATETIME	31h	n14		X	Horário do servidor PGWEB.
PWINFO_TRNORIGTIME	73h	n6	X		Horário da transação original. Este campo é utilizado para transações de cancelamento. Formato HHMMSS.
PWINFO_CARDENTMODE	C0h	b4		X	Modo de entrada do cartão: 1: número do cartão digitado 2: tarja magnética 4: chip com contato EMV 16: fallback para tarja magnética 32: chip sem contato simulando tarja magnética 64: chip sem contato EMV 128: indica que a transação atual é oriunda de um fallback (flag enviado do servidor para o ponto de captura). 256: <i>fallback</i> de tarja para digitado
PWINFO_CARDPARCPAN	C8h	ans..19		X	Número do cartão mascarado.
PWINFO_CHOLDVERIF	CFh	b1		X	Verificação do portador do cartão, soma de: 1: assinatura 2: verificação offline da senha 4: senha offline bloqueada durante a transação 8: verificação on-line da senha

PWINFO_MERCHADDDATA1	F0h	ans..127	X		Número de referência da transação atribuído pela Automação Comercial. Caso fornecido, este número será incluído no histórico de dados da transação e encaminhado à adquirente/processadora, se suportado. Este parâmetro é opcional.
PWINFO_MERCHADDDATA2	F1h	ans..127	X		Dados adicionais específicos do negócio. Caso fornecido, será incluso no histórico de dados da transação, por exemplo para referências cruzadas. Este parâmetro é opcional.
PWINFO_MERCHADDDATA3	F2h	ans..127	X		Dados adicionais específicos do negócio. Caso fornecido, será incluso no histórico de dados da transação, por exemplo para referências cruzadas. Este parâmetro é opcional.
PWINFO_MERCHADDDATA4	F3h	ans..127	X		Dados adicionais específicos do negócio. Caso fornecido, será incluso no histórico de dados da transação, por exemplo para referências cruzadas. Este parâmetro é opcional.
PWINFO_PNDAUTHSYST	7F05h	ans..20		X	Nome do provedor para o qual existe uma transação pendente.
PWINFO_PNDVIRTMERCH	7F06h	n..9		X	Identificador do Estabelecimento para o qual existe uma transação pendente.
PWINFO_PNDAUTLOCREF	7F08h	n..50		X	Referência para a infraestrutura Erro! Nome de propriedade do documento desconhecido. da transação que está pendente.
PWINFO_PNDAUTEXTREF	7F09h	ans..50		X	Referência para o Provedor da transação que está pendente.

PWINFO_DUEAMNT	BF06h	ans..12		X	Valor devido pelo usuário, considerando PWINFO_CURREXP, já deduzido em PWINFO_TOTAMNT.
PWINFO_READJUSTEDAMNT	BF09h	ans..12		X	Valor total da transação reajustado, este campo será utilizado caso o autorizador, por alguma regra de negócio específica dele, resolva alterar o valor total que foi solicitado para a transação.