



Especificação Técnica – Interface com os meios de pagamento do SiTef Interface Java JCliSiTef

Versão 2.1

Especificação Técnica – Interface com os meios de pagamento do SiTef	Versão: 2.1
Interface Java JCliSiTef	Data: 04/04/2014

Índice

1. Introdução	1
2. Arquitetura	1
3. Métodos de acesso	2
3.1 get/set + chamada ao método sem parâmetros	2
3.2 Chamada de métodos parametrizados	5
4. Pré-requisitos	6

Especificação Técnica – Interface com os meios de pagamento do SiTef	Versão: 2.1
Interface Java JCliSiTef	Data: 04/04/2014

Interface Java JCliSiTef

1. Introdução

Este documento descreve a interface Java da CliSiTef que, devido às particularidades dessa linguagem e do padrão normalmente utilizado pelos programadores, possui forma de ativação e troca de informações um pouco diferente da descrita no documento principal da CliSiTef.

2. Arquitetura

A parte Java da interface é representada pela classe

```
br.com.softwareexpress.sitf.JCliSiTefI
```

Além da parte Java, existe uma biblioteca dinâmica que implementa a translação do mundo java para o ambiente nativo da máquina onde o aplicativo irá rodar (JNI). Esse módulo é implementado pela biblioteca jCliSiTefI.dll ou libjCliSiTefI.so dependendo se o ambiente é Windows ou Linux.

Finalmente, esta biblioteca JNI faz uso das bibliotecas originais da CliSiTef.

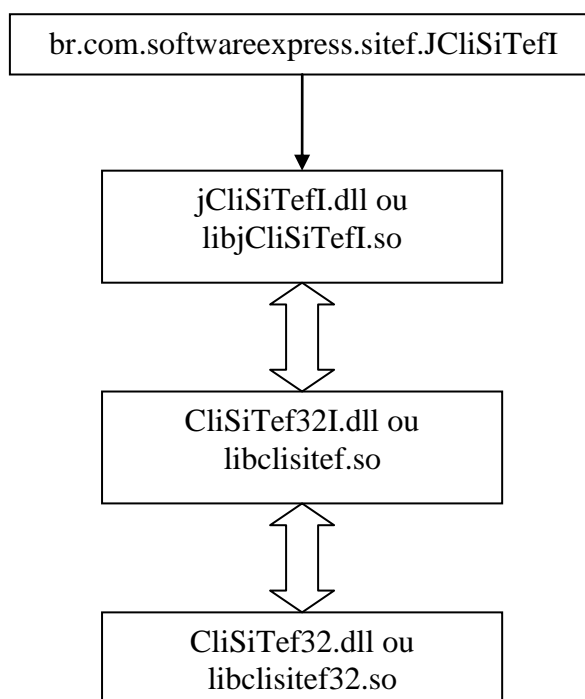


Figura 1 - Arquitetura geral para 32 bits (para 64bits é análogo).

Observação: a jCliSiTefI.dll **não** é uma biblioteca ActiveX, portanto não é necessário registrá-la no Windows, tampouco as bibliotecas CliSiTefI, CliSiTef ou libSEPPEmv.

Especificação Técnica – Interface com os meios de pagamento do SiTef	Versão: 2.1
Interface Java JCliSiTef	Data: 04/04/2014

3. Métodos de acesso

Existem duas formas de acessar os métodos da JCliSiTefI.

3.1 get/set + chamada ao método sem parâmetros

Por convenção, os acessos aos parâmetros de entrada e saída da interface são feitos através das funções:

```
<tipo> JCliSiTefI.get<parâmetro> ()

e

JCliSiTefI.set<parâmetro> (<tipo> <Valor>)
```

Assim, parâmetros de entrada das rotinas têm um “setter” associado, e parâmetros de saída tem um método “getter”.

Note que é possível que um mesmo parâmetro seja de entrada/saída, e neste caso possui os métodos get e set.

Uma vez chamados os métodos “set” para fornecer os parâmetros necessários a execução da função, o método que corresponde a função escolhida deve ser chamado através do comando:

```
sts = JCliSiTefI.<método> ();
```

Onde <método> corresponde a qualquer uma das funções descritas no documento principal da CliSiTef.

Por exemplo, considere a função de configuração cujo protótipo em C é a seguinte:

```
long ConfiguraIntSiTefInterativo (char *EnderecoSiTef, char *CodigoLoja,
char *NumeroTerminal);
```

sua chamada correspondente fica:

```
int S;
JCliSiTefI CliSiTef = new JCliSiTefI ();

CliSiTef.setEnderecoSiTef ("192.168.1.1");
CliSiTef.setCodigoLoja ("00000000");
CliSiTef.setNumeroTerminal ("SE000001");
S = CliSiTef.configuraIntSiTefInterativo ();
if (S != 0)
{
    System.out.println ("Erro " + S + " na ConfiguraIntSiTefInterativo");
    System.exit (1);
}
```

Especificação Técnica – Interface com os meios de pagamento do SiTef	Versão: 2.1
Interface Java JCLiSiTef	Data: 04/04/2014

A tabela a seguir descreve o nome de todas as variáveis acessíveis via set/get e que, embora ligeiramente diferentes das descritas no documento original de interface da CliSiTef, é fácil fazer a correlação entre elas de forma que o programador saiba qual utilizar em cada situação.

Parâmetro	Tipo	Métodos existentes	
		get	set
Buffer	String	X	X
Campo	String	X	---
ChaveSecreta	String	---	X
ChaveSeguranca	String	X	X
ChaveTrnCancelamento	String	---	X
CodigoAutorizacao	String	---	X
CodigoEmBarras	String	---	X
CodigoLoja	String	---	X
CodigoProduto	String	---	X
CodigoResposta	short	X	---
CodigoServico	String	---	X
ConfiguraResultado	short	---	X
Confirma	short	---	X
ContinuaNavegacao	int	---	X
DadosCaptura	String	---	X
DadosContas	String	X	X
DadosDesfazimento	String	X	X
DadosEntrada	byte []	---	X
DataFiscal	String	---	X
DataFiscalOriginal	String	---	X
DadosRx	byte []	X	---
DadosSaida	String	X	---
DadosServico	String	X	---
DadosTransacao	String	---	X
DadosTx	byte []	---	X
Delimitador	short	---	X
EnderecoSiTef	String	---	X
eSenha	short	---	X
FuncaoSiTef	short	---	X
Horario	String	---	X
IdentificacaoPdvOriginal	String	---	X
IndiceParametro	short	---	X
IndiceProduto	short	---	X
Modalidade	int	---	X
MsgDisplay	String	---	X
NomeArquivo	String	---	X
NumeroCuponFiscal	String	---	X
NumeroCuponFiscalOriginal	String	---	X
NumeroDocumentoOriginal	String	---	X
NumeroProdutos	short	---	X
NumeroTerminal	String	---	X
OffsetCartao	short	---	X

Especificação Técnica – Interface com os meios de pagamento do SiTef	Versão: 2.1
Interface Java JCliSiTef	Data: 04/04/2014

Operador	String	---	X
Parametro	String	---	X
ParametrosAdicionais	String	---	X
ParametroCartao	short	---	X
Produtos	String	---	X
ProximoComando	int	X	---
Quantidade	short	---	X
RedeDestino	short	---	X
Senha	String	X	---
SenhaPinPad	String	---	X
Supervisor	String	---	X
TamanhoMaximo	int	X	X
TamanhoMinimo	int	X	X
TamDadosEntrada	int	---	X
TamDadosTx	short	---	X
TamMaxDadosRx	short	---	X
TamMaxDadosServico	short	---	X
TempoEsperaRx	short	---	X
Timeout	short	---	X
TipoCampo	int	X	---
TipoCancelamento	short	---	X
TipoCodigoEmBarras	String	X	X
TipoConsulta	int	---	X
TipoOperacao	int	---	X
TipoTransacao	short	---	X
TipoTrilha1	String	X	X
TipoTrilha2	String	X	X
TotalGeral	String	---	X
Totalizador	String	---	X
Trilha1	String	X	X
Trilha2	String	X	X
Trilha3	String	X	X
Valor	String	---	X
ValorTotalBonus	String	---	X
ValorTotalCupon	String	---	X
VersaoCliSiTef	String	X	---
VersaoCliSiTefI	String	X	---

Observação: para simplificar o trabalho do programador, todas as funções que possuem como parâmetro o tipo short podem ser acionadas passando-se um parâmetro do tipo int, pois existe o método correspondente para esse tratamento.

Especificação Técnica – Interface com os meios de pagamento do SiTef	Versão: 2.1
Interface Java JCliSiTef	Data: 04/04/2014

3.2 Chamada de métodos parametrizados

Com o crescente aumento de rotinas, e consequentemente o número de variáveis, criou-se uma segunda interface onde é possível passar os parâmetros de entrada, sendo que os parâmetros de saída continuam sendo obtidos através dos métodos Getters descritos acima.

Exemplo:

```
long ConfiguraIntSiTefInterativo (char *EnderecoSiTef, char *CodigoLoja,
char *NumeroTerminal)
```

pode ser chamada por:

```
int i;
JCliSiTefI CliSiTef = new JCliSiTefI ();

i = CliSiTef.configuraIntSiTefInterativo ("192.168.1.1","00000000","SE000001");
if (i != 0) {
    System.out.println ("Erro " + i + " na ConfiguraIntSiTefInterativo");
    System.exit (1);
}
```

Especificação Técnica – Interface com os meios de pagamento do SiTef	Versão: 2.1
Interface Java JCLiSiTef	Data: 04/04/2014

4. Pré-requisitos

Deve-se ter em mente que o JNI deve atender quatro requisitos básicos:

- 1) A versão do Java deve ser compatível com a arquitetura das bibliotecas. Isto é, se o Java for 32 bits, as bibliotecas devem ser obrigatoriamente de 32 bits.

Note que é possível instalar um Java 32 bits em uma máquina com S.O. de 64 bits. O que vale é a arquitetura do processo raiz, isto é, a Java Virtual Machine (JVM); neste caso, as bibliotecas devem ser de 32 bits.

- 2) O Java deve ser capaz de encontrar a classe do pacote jar. Normalmente isto é feito indicando-se no CLASSPATH o pacote .jar.
- 3) O Java deve ser capaz de encontrar a biblioteca JNI.

No Windows, normalmente basta colocar a biblioteca em uma pasta do PATH.

No Linux, normalmente é usado a variável interna java.library.path. Uma forma de se fazer pela linha de comando é:

```
java -Djava.library.path=../linux64 -classpath ../LibSiTef.jar:.
ExemploLibSiTef
```

- 4) A biblioteca JNI deve ser capaz de carregar as DLL's finais.

No Windows, normalmente basta colocar as DLL's em uma pasta do PATH, ou na mesma pasta da biblioteca JNI.

No Linux, pode ser usado a variável de ambiente LD_LIBRARY_PATH, ou através do ldconfig.

Especificação Técnica – Interface com os meios de pagamento do SiTef	Versão: 2.1
Interface Java JCliSiTef	Data: 04/04/2014

Histórico das Revisões

Data	Versão	Descrição	Autor
27/10/2010	1.0	Preenchimento inicial, baseado na edição 0.06	Alexandre Hamada
04/02/2011	1.1	Inclusão da documentação da applet. Inclusão de histórico do documento.	Alexandre Hamada
21/03/2014	1.2	Referências da applet movidas para o documento “Interface Applet CliSiTef”	Alexandre Hamada
02/04/2013	2.0	Atualização da nomenclatura das classes “obsoletas”.	Alexandre Hamada
04/04/2013	2.1	Inclusão do item pré-requisitos	Alexandre Hamada