# **Assinatura Digital de Documentos – CEP**

# Descrição da solução

A assinatura digital de documentos poderá ser realizada através da utilização de uma aplicação Java Web Start que ficará disponível no site do CEP. Essa aplicação foi desenvolvida em Java utilizando componentes Swing para interface e feedback para o usuário.

Para utilização, é necessário o download do arquivo de extensão jnlp através da url (testes) <https://testeseguro7.petrobras.com.br/SCEP/Content/assinatura/assinaturaDocumentos.jnlp>. Sugere-se o download em uma pasta isolada (Figura 2) pois alguns arquivos serão carregados do servidor para a configuração do assinador. Se nenhum diretório for selecionado (Figura 1), o browser utilizará a pasta default de arquivos temporários, C:\Users\{chave}\AppData\Local\Temp.

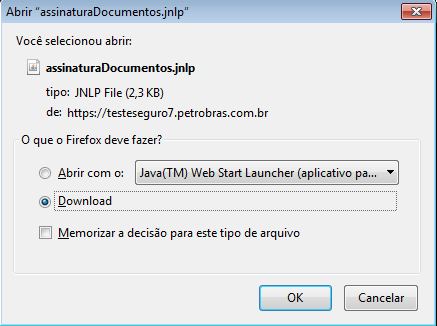
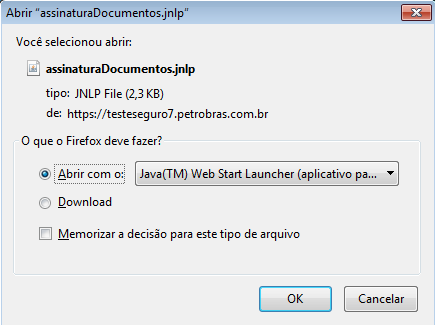


Figura 1-Diretório Não Selecionado

Figura 2- Diretório Selecionado

Antes da execução do arquivo jnlp, deve-se configurar o Painel de Controle Java para incluir a url acima como lista de exceções de Sites (Figura 3) por não ser um site com um certificado válido (Figura 4). A execução em produção em DMZ exige que o certificado seja válido para que este passo não seja necessário nas máquinas do usuário.

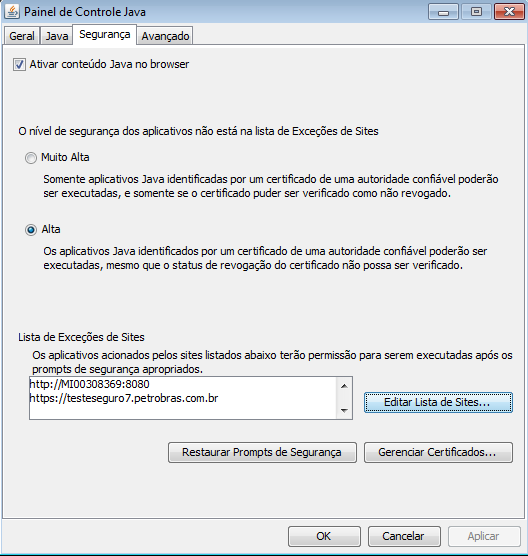
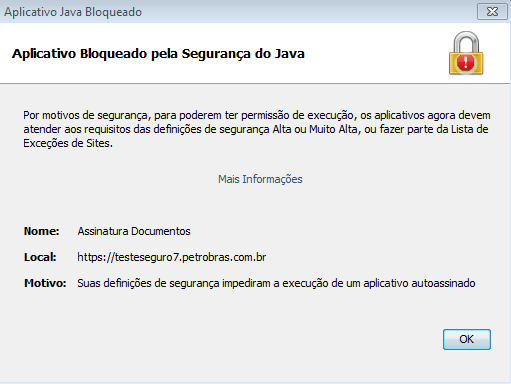


Figura 4 - Aplicativo Bloqueado

Figura 3 - Configuração da Lista de Exceção de Sites

Após configuração anterior, é necessário marcar o checkbox “Eu aceito o risco e desejo executar este aplicativo” (Figura 5), devido às restrições de segurança e devido ao acesso irrestrito na estação de execução. O acesso irrestrito é necessário pois a manipulação de pastas e arquivos é necessária para o processo de assinatura. Nenhum arquivo do sistema operacional é manipulado e apenas os arquivos envolvidos no processo de assinatura são criados ou alterados.

A correta configuração do servidor e a assinatura do arquivo .jar executado pelo aplicativo com um certificado válido impede a exibição da informação de risco à segurança exibida abaixo.

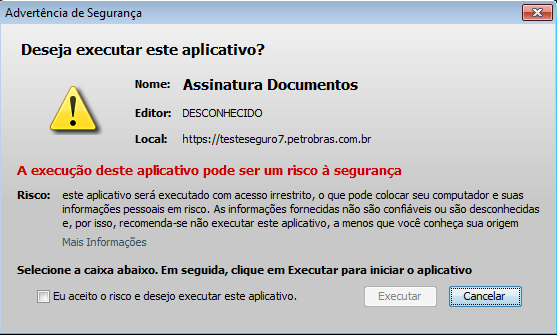


Figura 5 - Seleção Aceito o Risco

O processo de assinatura desta solução é composto pelas seguintes etapas:

1. **Configuração da aplicação**

Nesta etapa (Figura 6), os arquivos necessários para a execução são carregados do CEP para a estação de trabalho do usuário no diretório escolhido ou diretório padrão (Figura 1 e Figura 2). A estrutura dos arquivos pode ser visualizada na Figura 7.

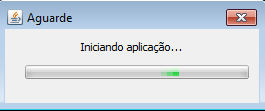


Figura 6 - Configurando a aplicação



Figura 7 - Carga dos Arquivos de configuração do Aplicativo

**cert –** arquivos contendo toda a cadeia de certificados que serão carregados pela aplicação respeitando a ICP-Brasil (Infraestrutura de Chaves Públicas Brasileira). A sua atualização é esporádica, porém necessária de acordo com novas cadeias de certificados.

**crl –** lista de certificados revogados por algum motivo, antes da data de expiração do mesmo. A sua atualização é esporádica, porém necessária de acordo com as novas revogações.

**logs –** diretório contendo o arquivo de log da aplicação.

**policy-signature –** diretório contendo osarquivos com informações referentes à codificação dos certificados.

**tempAssinaturaArquivosCep –** diretório temporário onde os arquivos a serem assinados serão carregados do servidor além dos arquivos .p7s, representando a assinatura dos mesmos.

**assinaturaDocumentos.jnlp** – arquivo baixado do site para execução do aplicativo.

**general.properties** – arquivo contendo informações para configuração da aplicação.

**license.xml** – arquivo contendo informações referente à licença da API de assinatura.

1. **Download dos arquivos para assinatura**

Nesta etapa, a aplicação irá realizar a carga dos arquivos para assinatura no diretório **tempAssinaturaArquivosCep**. A cada execução, o conteúdo do mesmo é removido e os novos arquivos carregados são transferidos para este diretório.

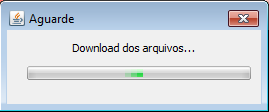


Figura 8 - Download dos Arquivos para Assinatura

1. **Busca dos certificados**

O aplicativo irá buscar no sistema operacional todos certificados instalados (certificados A1) ou disponíveis (certificados A3).

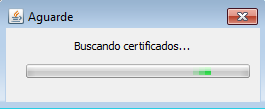


Figura 9 - Busca dos Certificados

1. **Seleção do certificado para assinatura**

A seleção de um certificado válido é necessária para realizar a assinatura dos documentos. Caso um certificado não seja selecionado ou seja inválido, o processo de assinatura é interrompido.

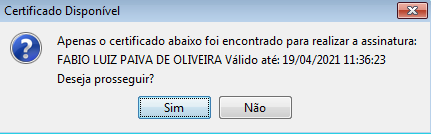


Figura 10 - Seleção do Certificado para Assinatura

1. **Assinatura dos documentos**

Os documentos carregados no item 2 (Download dos arquivos para assinatura) são assinados através do certificado selecionado no item 4 (Seleção do certificado para assinatura). Será necessário informar o PIN do certificado para certificados do tipo A3.

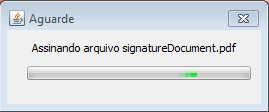


Figura 11 - Assinatura dos Documentos

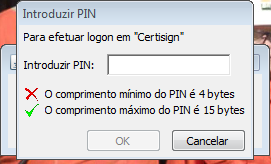


Figura 12 - Solicitação do PIN

1. **Upload dos arquivos para o CEP**

Após a assinatura dos documentos, os mesmos serão enviados através de um serviço disponível pelo CEP para uma pasta no NAS.

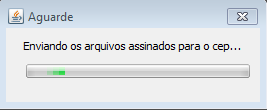


Figura 13 - Upload para o CEP

1. **Feedback com o resultado da assinatura**

Após a finalização da assinatura ou a ocorrência de algum erro, uma mensagem é exibida ao usuário com as informações sobre o ocorrido.

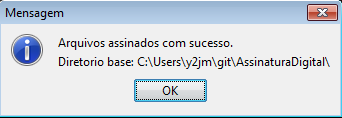


Figura 14 - Mensagem Sucesso

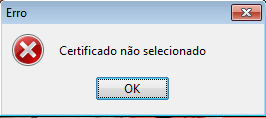


Figura 15 - Mensagem Erro

# Informações sobre implementação

Uma das premissas do desenvolvimento desta solução é a não utilização do pacote sun.security devido à falta deste pacote em algumas versões do Java. Dessa forma, a solução pode ser executada em máquinas com JRE instalada a partir da versão 1.6.

Todo o processo é realizado com as classes listadas na figura abaixo:

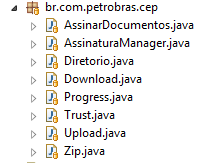


Figura 13 - Classes

**AssinarDocumentos** – Classe principal, executada pelo arquivo jnlp. Não contém código referente à nenhuma etapa, sendo apenas responsável pela orquestração e feedback do usuário em relação às etapas do processo.

**AssinaturaManager –** Classe responsável pela busca de certificados e assinatura de documentos. Utiliza a API certisigner-1.12.0.22.jar.

**Diretorio –** Classe responsável pela manipulação de diretórios.

**Download –** Classe responsável pelo download de arquivos via protocolo HTTP/HTTPS.

**Progress –** Classe para abstração do componente Swing JProgressBar a fim de facilitar sua utilização.

**Trust –** Classe para “*trust*” do site do CEP em requisições HTTPS por se tratar de um servidor cujo certificado não se encontra válido.

**Upload –** Classe responsável pelo upload de arquivos para uma URL via protocolo HTTP/HTTPS.

**Zip –** Classe responsável pelo zip/*unzip* de arquivos.

O Jar de execução resultante das classes acima e as bibliotecas envolvidas nesta solução (Figura 14) devem ser disponibilizadas assinadas para execução. Para o ambiente de teste, as mesmas podem ser auto assinadas pelas linhas de comando abaixo:

Geração do keystore:

**keytool -genkeypair -alias {alias} -keyalg RSA -dname "CN={CN},OU={OU},O={O},L={L},ST={ST},C={C}" -keypass {password} -storepass {password}**

Ex.: **keytool -genkeypair -alias tomcat -keyalg RSA -dname "CN=tomcat,OU=Petrobras,O=Softtek,L=Rio de Janeiro,ST=RJ,C=BR" -keypass teste123 -storepass teste123**

Assinatura do arquivo:

**jarsigner assinaturaDocumentos.jar {alias} -storepass {senha}**

Ex.: **jarsigner assinaturaDocumentos.jar fabio -storepass teste123**

No ambiente de produção, os mesmos devem ser assinados por um certificado digital válido.

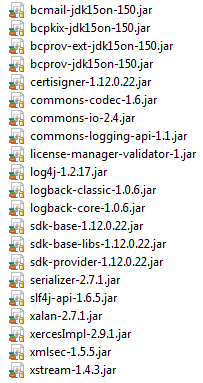


Figura 14 - Libs utilizadas

A aplicação pode ser parametrizada caso necessário através da alteração das propriedades existentes no arquivo .jnlp.

jnlp.url.arquivo.configuracao – Url do arquivo .zip de onde será feito o download contendo os arquivos de configuração da aplicação (Figura 6).

jnlp.url.arquivo.assinatura – Url do arquivo .zip contendo os arquivos a serem assinados pela aplicação (Figura 8).

jnlp.diretorio.temp.arquivos – Diretório onde ficarão os arquivos temporários utilizados pela aplicação.