

STK (Start Kit DARUMA)

Assiando os arquivos com a DarumaFramework.dll – Versão 1.0

Premissas:

1. Ter impressora de modelo FS600/ FS2100T (de versão 01.05.00 ou superior), FS700 ou MACH.
2. DarumaFramework.dll versão 4.16.1.0 ou superior
3. Executável de exemplo para testes.

Este STK divide-se em 6 partes:

1. Breve resumo sobre o algoritmo RSA;
2. Gerando uma chave privada;
3. Gerando Assinatura Digital (EAD);
 - 3.1. Para assinar manualmente;
 - 3.2. Para assinar automaticamente;
4. Obtendo a Chave Pública para validação de arquivos;
5. Validando Assinatura utilizando o aplicativo ECFc.

Importante: Sempre que você encontrar [XX], em colchetes, quer dizer que se trata de um caractere apenas, ou seja, é um chr.

1. Primeiramente vamos entender melhor o que vem a ser o Algoritmo RSA.

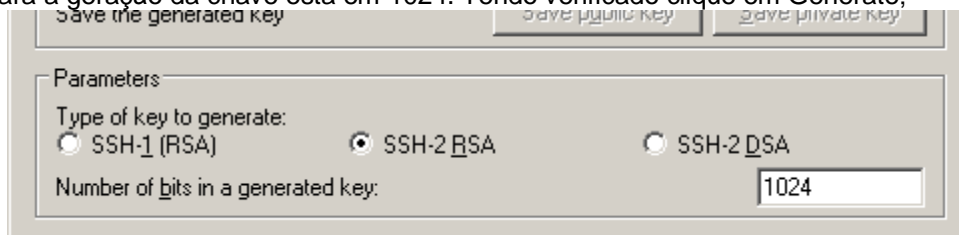
RSA é um algoritmo de criptografia de dados, que deve o seu nome a três professores do Instituto MIT (fundadores da atual empresa RSA Data Security, Inc.), Ron Rivest, Adi Shamir e Len Adleman, que inventaram este algoritmo — até a data (2008), a mais bem sucedida implementação de sistemas de chaves assimétricas, e fundamenta-se em teorias clássicas dos números. É considerado dos mais seguros, já que mandou por terra todas as tentativas de quebrá-lo. Foi também o primeiro algoritmo a possibilitar criptografia e assinatura digital, e uma das grandes inovações em criptografia de chave pública.

2. Gerando uma chave privada.

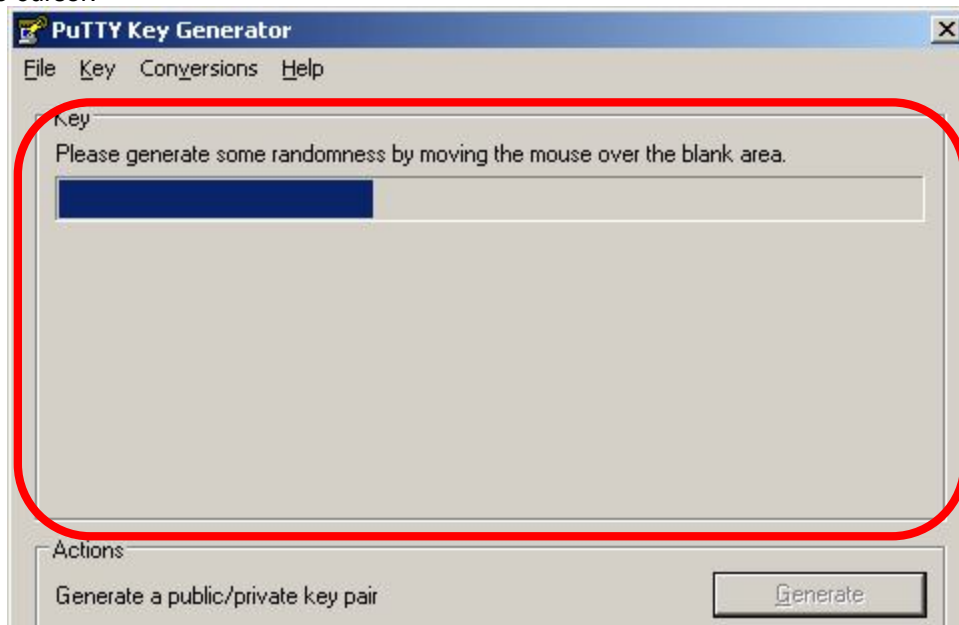
Existem vários softwares pagos ou free disponíveis para uma gerar chave privada, para o este teste utilizamos o puttygen.

2.1. Rode o puttygen a partir do site: <http://the.earth.li/~sgtatham/putty/latest/x86/puttygen.exe>

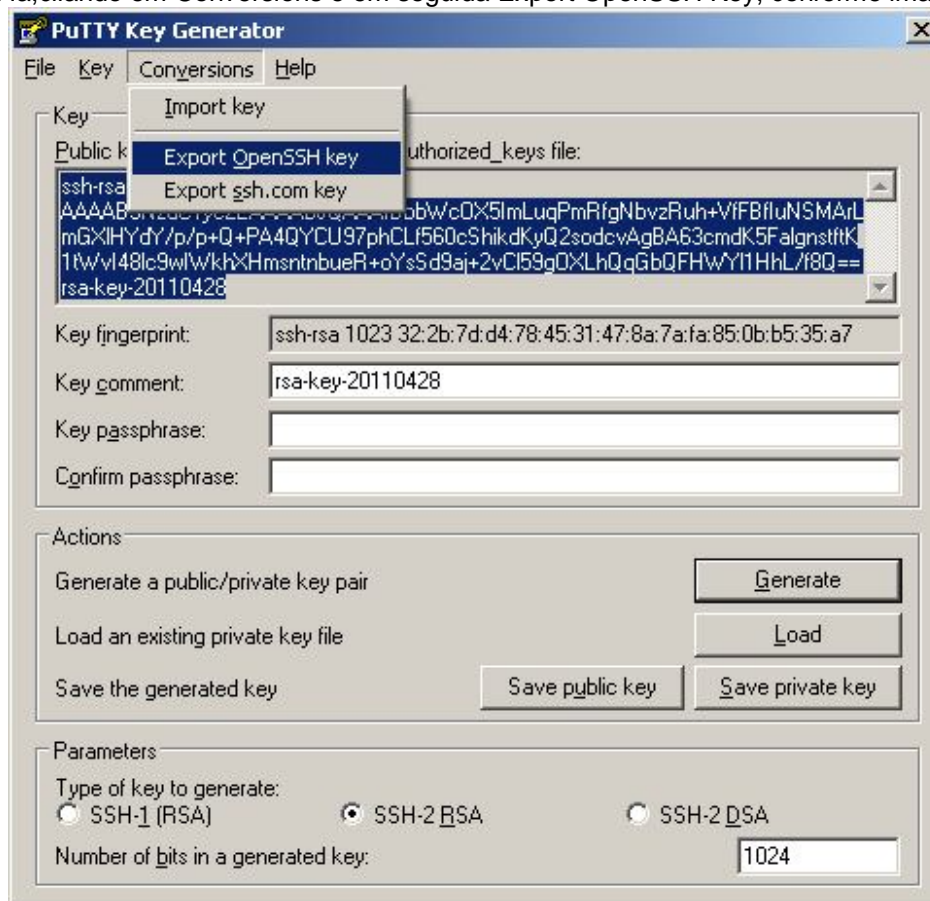
2.2. Antes de clicar no botão pra gerar a chave, confira de SSH-2 RSA esta selecionado e se o numero de bit's para a geração da chave esta em 1024. Tendo verificado clique em Generate;



Observação: após clicar no botão, para que seja gerada a chave, movimente o mouse sobre a tela do puttygen. Esse movimento é necessário, pois o algoritmo que gera a chave utiliza dentre outros cálculos, a posição X/Y do cursor.



2.3. Após a progress bar ser preenchida, com o movimento do mouse, a chave será gerada e é hora de exportá-la, clicando em Conversions e em seguida Export OpenSSH Key, conforme imagem abaixo:



Ao clicar em Export OpenSSH Key, uma pergunta será feita, conforme imagem abaixo, clique em Yes para continuar e escolha o caminho e nome para o seu arquivo .key.

3. Gerando Assinatura Digital (EAD)

3.1. Gerando Assinatura Digital e Assinando Manualmente o arquivo

Iremos utilizar o comando `rAssinarRSA_ECF_Daruma`, passando como parâmetros três variáveis do tipo string. Uma para informar o caminho do arquivo .key, outra para indicar o caminho do arquivo a ser assinado e a última por referência para receber o conteúdo da assinatura digital. Conforme imagens a seguir:

```

procedure TFR_FISCAL_AssinarRSA_ECF_Daruma.BT_ENVIARClick(Sender: TObject);
var
    Str_EAD, Str_CaminhoArqAssinar, Str_CaminhoChavePublica: string;
begin
    SetLength (Str_EAD,256);
    Str_CaminhoArqAssinar:= Edt_CaminhoArqAssinado.text;
    Str_CaminhoChavePublica:= Edt_CaminhoChavePrivada.text;
    Int_Retorno := rAssinarRSA_ECF_Daruma(Str_CaminhoArqAssinar,Str_CaminhoChavePubl:
    MM_EAD.Lines.Text := '';
    MM_EAD.Lines.Text := Trim(Str_EAD);
    FR_MenuImpressoraFiscal_Principal.DarumaFramework_Mostrar_Retorno(Int_Retorno);
end;
  
```

Imagem 1 – Exemplo de código em Delphi.

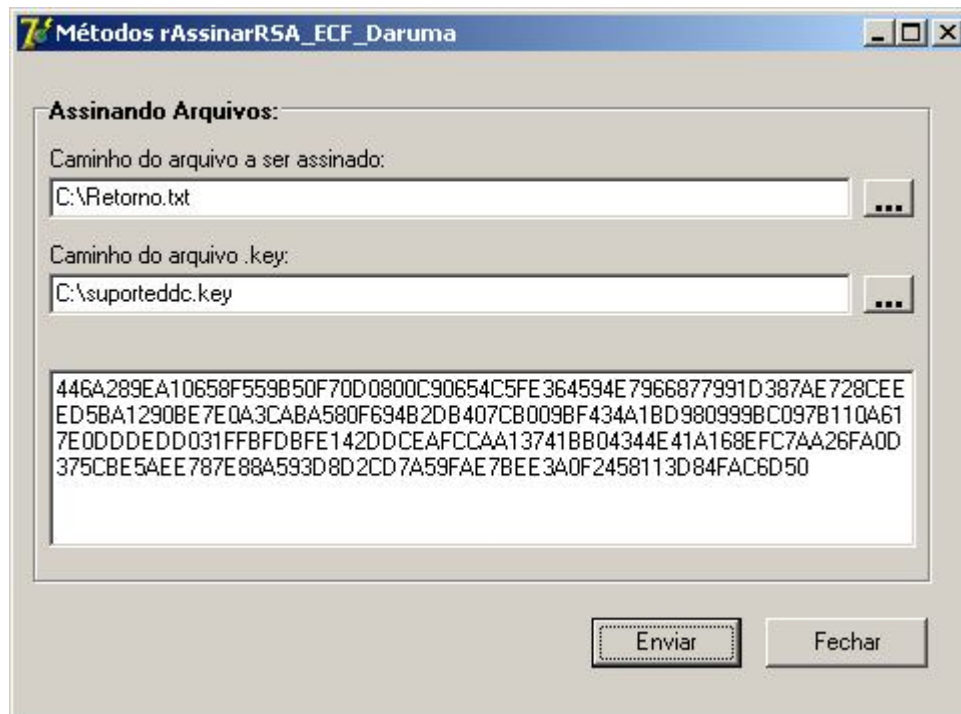
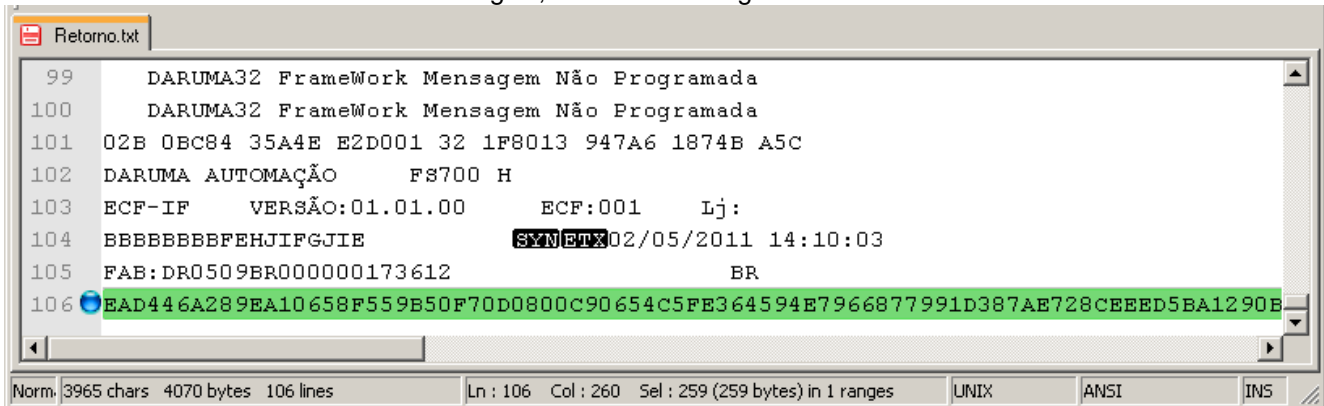


Imagem 2 – Variável contendo o valor do local + nome do arquivo a ser assinado,
Variável contendo o valor do local + nome do arquivo.key,
Variável contendo o valor da Assinatura digital do arquivo informado.

Após obter assinatura digital do arquivo informado, você deverá inserir na última linha do arquivo o Registro “EAD” + o valor da variável Assinatura Digital, conforme a imagem abaixo:



3.2. Gerando Assinatura Digital e Assinando Automaticamente o arquivo.

Neste caso iremos utilizar o comando eRSAAssinarArquivo_ECF_Daruma, passando como parâmetros apenas duas variáveis do tipo string. Uma para informar o caminho do arquivo .key, outra para indicar o caminho do arquivo a ser assinado. O método já irá gerar a Assinatura e inclui-la automaticamente no final do arquivo. Conforme imagens a seguir:

```

procedure TFR_FISCAL_eRSAAssinarArquivo_ECF_Daruma.BT_ENVIARClick(
  Sender: TObject);
var
  Str_EAD, Str_CaminhoArqAssinar, Str_CaminhoChavePublica: string;
begin
  Str_CaminhoArqAssinar:= Edt_CaminhoArqAssinado.text;
  Str_CaminhoChavePublica:= Edt_CaminhoChavePrivada.text;
  Int_Retorno := eRSAAssinarArquivo_ECF_Daruma(Str_CaminhoArqAssinar,Str_CaminhoChavePublica);
  FR_MenuImpressoraFiscal_Principal.DarumaFramework_Mostrar_Retorno(Int_Retorno);
end;
  
```

Imagem 1 – Exemplo de código em Delphi.

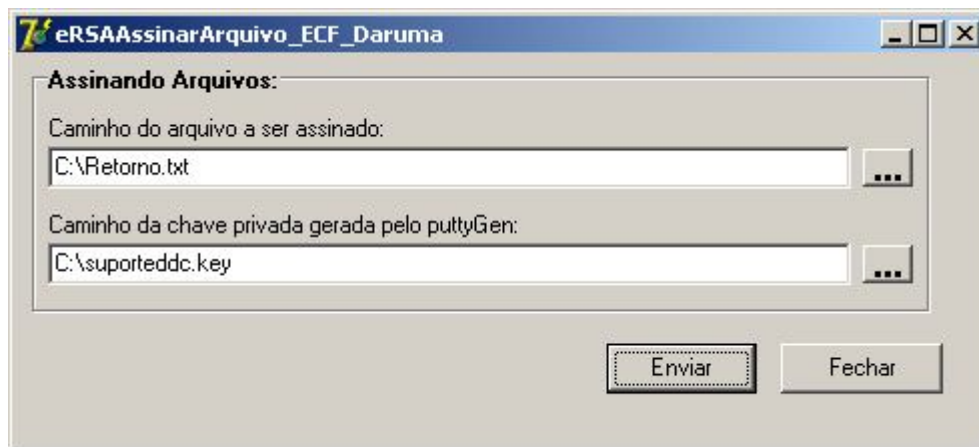


Imagem 2 – Variável contendo o valor do local + nome do arquivo a ser assinado,
Variável contendo o valor do local + nome do arquivo.key,
Variável contendo o valor da Assinatura digital do arquivo informado.

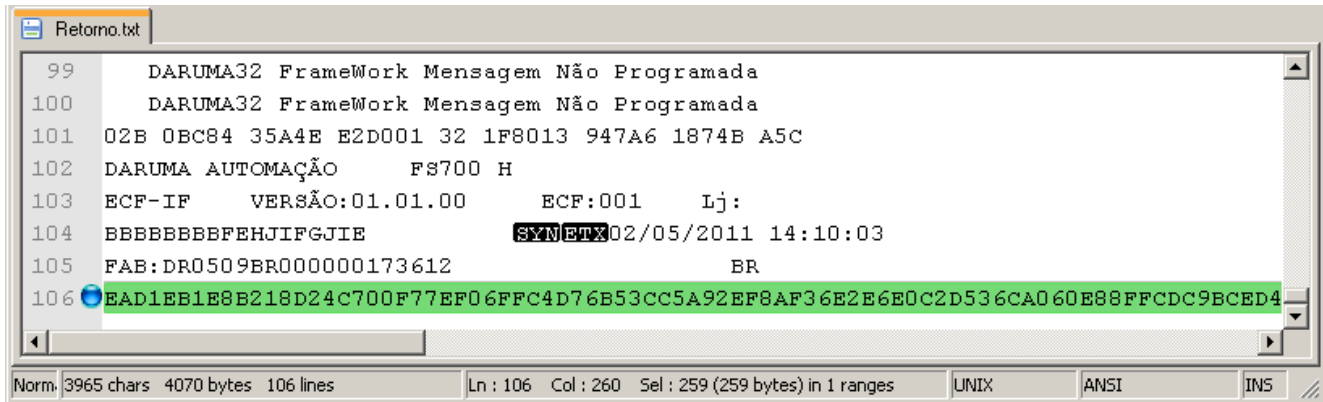


Imagem 3 – Arquivo já assinado automaticamente pelo método.

4. Obtendo a Chave Pública para validação de arquivos

Para obtermos a chave pública utilizaremos o método `rRSACHavePublica_ECF_Daruma`, passando como parâmetros três variáveis, sendo as duas ultimas por referencia, a primeira para informar o caminho e nome do arquivo .key, uma para obter o valor de “n” (modulo publico) e outra para armazenar o valor de “e” (expoente publico), conforme as imagens, 1, e 2 abaixo:

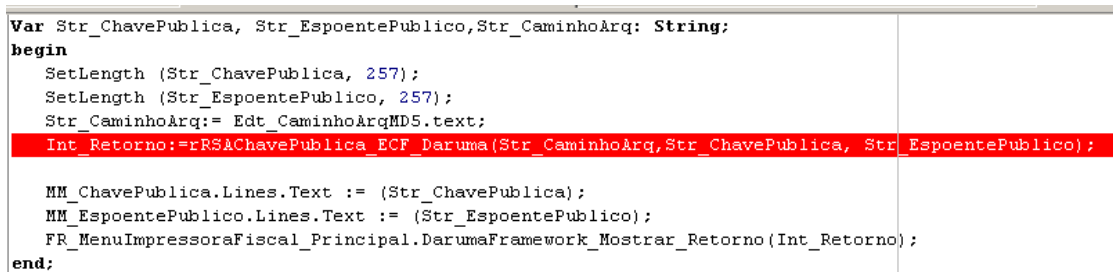


Imagem 1 – Exemplo de código em Delphi.

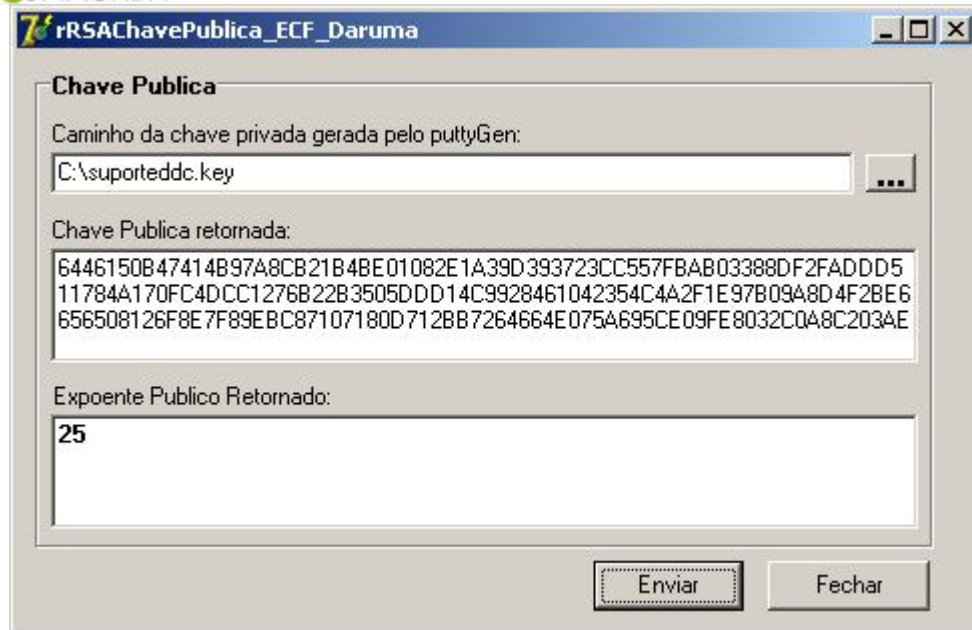


Imagem 2 – Variável contendo o valor de “n” (modulo publico, com 256 bytes)
Variável contendo o valor de “e” (expoente publico, com 2 bytes)

5. Validando Assinatura utilizando o aplicativo ECFc

Conforme previsto no ATO COTEPE/ICMS N°09 – Altera o Anexo VIII do ATO COTEPE/ICMS n°06/08, faz-se necessário a validação dos documentos assinados digitalmente e para isso é necessário utilizar o aplicativo eECFc.

1ºPasso: Baixe o eECFc caso ainda não tenha:

http://www.desenvolvedoresdaruma.com.br/home/downloads/Site_2011/Utilitarios/eECFc.zip

2º Passo: Preencher o arquivo XML com o nome da empresa desenvolvedora, a chave pública, contendo o módulo e o expoente público obtidos através do método rRSACHavePublica_ECF_Daruma (passo anterior). O arquivo XML deverá ser salvo dentro da pasta SHouse do aplicativo eECFc.

```

1  <?xml version="1.0" ?>
2
3  <empresa_desenvolvedora>
4
5      <nome>DARUMA</nome>
6
7      <chave>
8          <modulo>
9              6446150B47414B97A8CB21B4BE01082E1A39D393723CC557FBAB03388DF2FADDD511784A170FC4DCC
10             1276B22B3505DDD14C9928461042354C4A2F1E97B09A8D4F2BE6656508126F8E7F89EBC87107180D7
11             12BB7264664E075A695CE09FE8032C0A8C203AE63D8F21738D2DA7B8959E9779F15C259A341CF54F4
12             4A5D0681B5E8D</modulo>
9          <expoente_publico>25</expoente_publico>
10         </chave>
11
12 </empresa_desenvolvedora>
  
```

Obs: O Módulo deve ser preenchido em uma única linha, aqui quebrei as linhas só para mostrar como deve ser realizado o preenchimento. O nome da empresa desenvolvedora deve ser idêntico ao nome do arquivo.

3º Passo: Validar Chave Pública, iremos utilizar o aplicativo eECFc. Clique no botão Validar Assinatura PAF-ECF, irá abrir uma tela para que você selecione o arquivo XML e depois o arquivo assinado para ser validado. Como pode ver nas imagens a seguir:

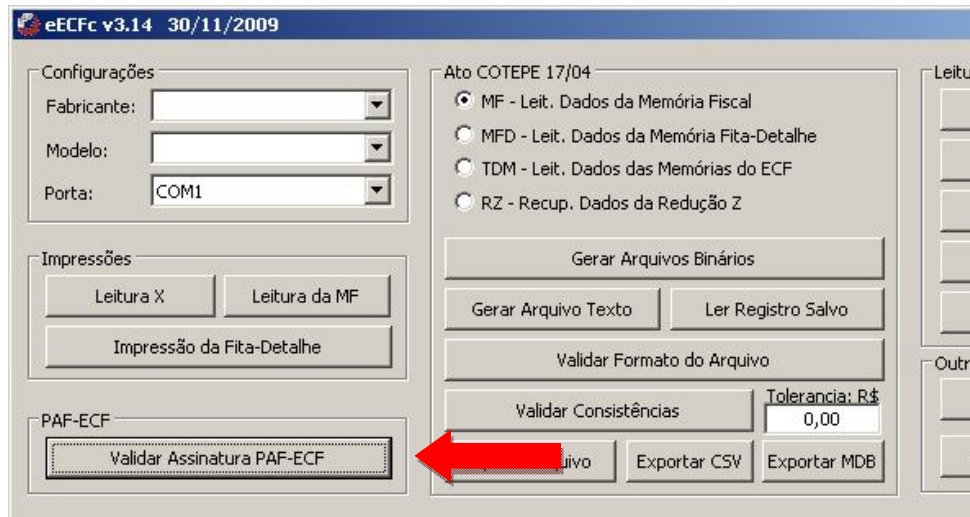


Imagem 1 – Opção utilizada para validar assinaturas do PAF-ECF.

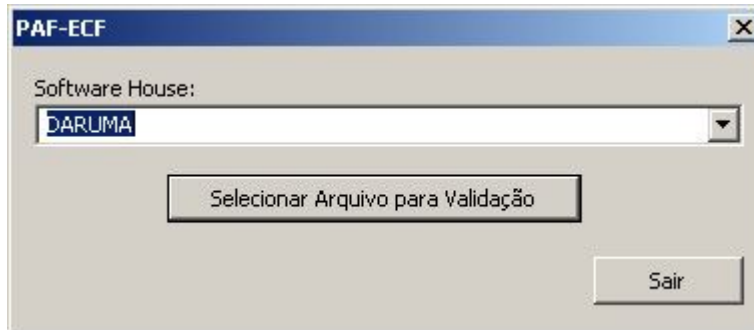


Imagem 2 – Irá aparecer os Arquivos XML existentes no diretório Shouse, ecolha o que você configurou.

4º Passo – Agora você deverá selecionar o arquivo assinado para validação, clicando no botão “Selecionar Arquivo para Validação”.

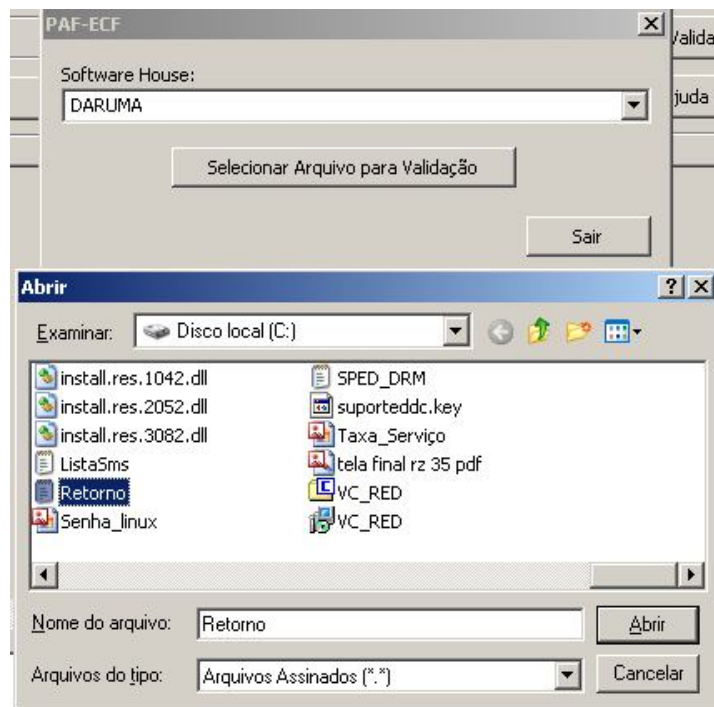


Imagem 1 – Selecionando Arquivo assinado para validação.

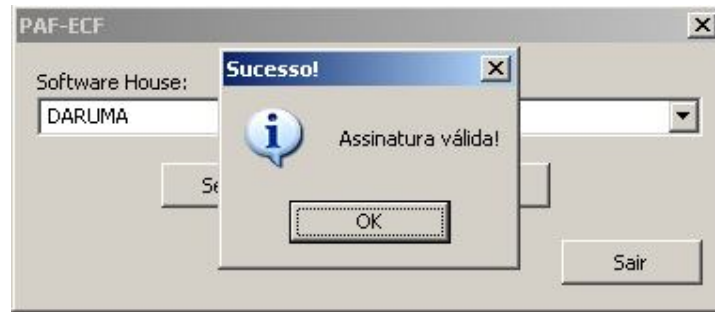


Imagem 2 – Arquivo Validado com Sucesso.

Se tiver alguma dúvida e/ou dificuldade, entre em contato com a nossa equipe de suporte ao desenvolvedor.

Telefone:

Suporte ao Desenvolvedor 0800 770 3320

E-mails:

suporte@daruma.com.br , desenvolvedores.suporte@daruma.com.br ,
desenvolvedores.daruma@daruma.com.br , daruma.desenvolvedores@daruma.com.br ,
suporte.ddc@daruma.com.br , ddc.suporte@daruma.com.br , suporte.desenvolvedores@daruma.com.br ,
suporte.alexandre@daruma.com.br , claudenir@daruma.com.br

Skypes:

suporte_daruma, desenvolvedores_suporte_daruma, suporte_desenvolvedores_daruma,
desenvolvedores_daruma, daruma.desenvolvedores, suporte_ddc_daruma, ddc_suporte_daruma,
daruma_suporte_alexandre, claudenir_andrade