UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
Faculdade de Ciências Exatas (FACET)
Departamento de Computação

Iago Mateus Avila Fernandes
Lucas Alexsander Barbosa Cruz
Mariano Carlos Silva
Mateus Henrique Machado
Rafael Lucas Fernandes Soares

Segurança e Auditoria de Sistemas de Informação:

Certificação Digital e Chave Pública autoassinada com openssl

Diamantina - MG

2024

A navegação segura na web é uma preocupação crescente entre usuários e desenvolvedores de soluções digitais. Garantir que a troca de informações entre o navegador e o servidor seja feita de maneira segura é fundamental para a saúde do ambiente web e dos usuários dessas aplicações. Uma das formas de fazer isso é a partir do uso de certificados digitais, normalmente assinados por Autoridades Certificadoras que são tidas como confiáveis pelos fornecedores de aplicações web e navegadores.

Durante o desenvolvimento de sistemas, uma das formas de testar a aplicação desses certificados é a partir da implementação de Certificados Digitais Autoassinados. Assim, este tutorial tem como objetivo descrever os passos necessários para gerar um certificado para uma aplicação de domínio local em um ambiente Windows, o que inclui a geração de chaves privadas, públicas e de uma Autoridade Certificadora local. Para isso, faremos uso do OpenSSL, sendo.

1. Instalando o OpenSSL

O primeiro passo consiste na instalação da biblioteca OpenSSL em nossa máquina. Essa biblioteca, por padrão, vem associada ao Git. No entanto, muitas vezes a instalação não traz todas as ferramentas necessárias para o nosso objetivo. Assim, optamos pelo uso de um distribuidor terceiro com grau de confiabilidade entre desenvolvedores.

Primeiramente, navegue para a página do distribuidor a partir do link destacado abaixo. Uma vez que o acesso foi feito, procure pelo arquivo de instalação adequado para a sua máquina Windows. Como estamos usando um sistema 64 bits, o arquivo utilizado foi o Win64 OpenSSL v3.3.0 (versão mais atualizada até o momento de escrita deste tutorial).

```
https://slproweb.com/products/Win32OpenSSL.html
```

Com o executável em mãos, siga os passos para a instalação da biblioteca em sua máquina. Ao final da instalação, você deve encontrar a pasta OpenSSL-Win64 no seu diretório de arquivos e programas.

Após a instalação, adicione o OpenSSL às variáveis de ambiente do seu sistema. Ao final, a sua configuração deve ficar similar ao apresentado na figura a seguir. Você pode

verificar se a configuração foi feita corretamente tentando executar o comando `openssl version` no terminal da sua máquina. O resultado deverá ser a versão atualmente instalada.

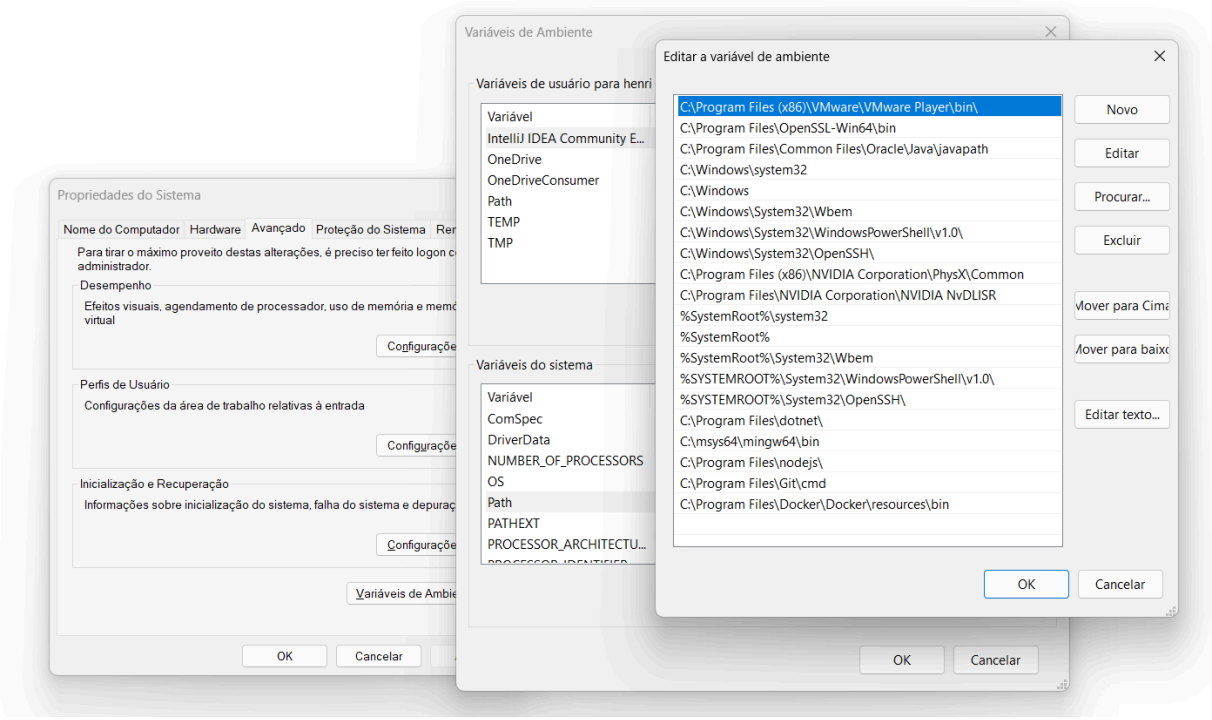

Figura 1: Adição do OpenSSL às variáveis de ambiente de uma máquina Windows

2. Criando um certificado SSL Root

Com a biblioteca instalada, o próximo passo consiste em criar um certificado *Secure Sockets Layer (SSL)* Root. Esse certificado será utilizado para assinar todos os outros certificados de domínio individuais, essencialmente tornando a sua máquina uma autoridade certificadora que iremos reconhecer como confiável localmente.

Para isso, primeiramente é necessário gerar uma chave `RSA-2048` que será salva no arquivo `rootCA.key`, e será usada como a chave privada para gerar certificado SSL root. Nesse momento será exigido que você cadastre uma senha para essa chave, então lembre-se de guardá-la bem pois será necessária em outras etapas.

Para gerar a chave RSA-2048, execute o comando abaixo no terminal da sua máquina:

```
openssl genrsa -des3 -out rootCA.key 2048
```

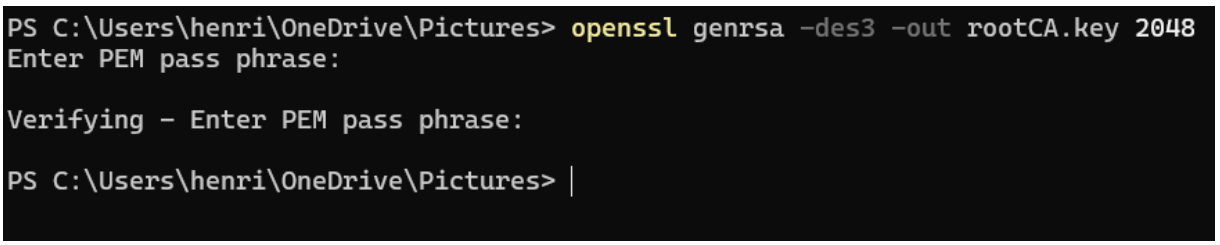

Figura 2: Geração da chave RSA-2048 no terminal cmd do windows

Uma vez que a chave foi gerada, ela poderá ser usada para criar o nosso certificado SSL root, que será armazenado no arquivo "`rootCA.pem`". O parâmetro `-days [NUM]` poderá ser usado para definir o prazo de validade deste certificado. Aqui usaremos o padrão de 1024 dias, mas você pode alterá-lo conforme sua necessidade.

No mesmo diretório que o arquivo rootCA.key foi gerado, execute o seguinte comando no seu terminal:

```
openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem
```

Ao ser executado, será necessário que você forneça a senha da chave gerada no passo anterior e, depois, preencha uma série de informações adicionais, como o país, estado e organização. Caso você queira deixar tudo vazio, somente dê enter. A figura a seguir mostra o resultado esperado no seu terminal.

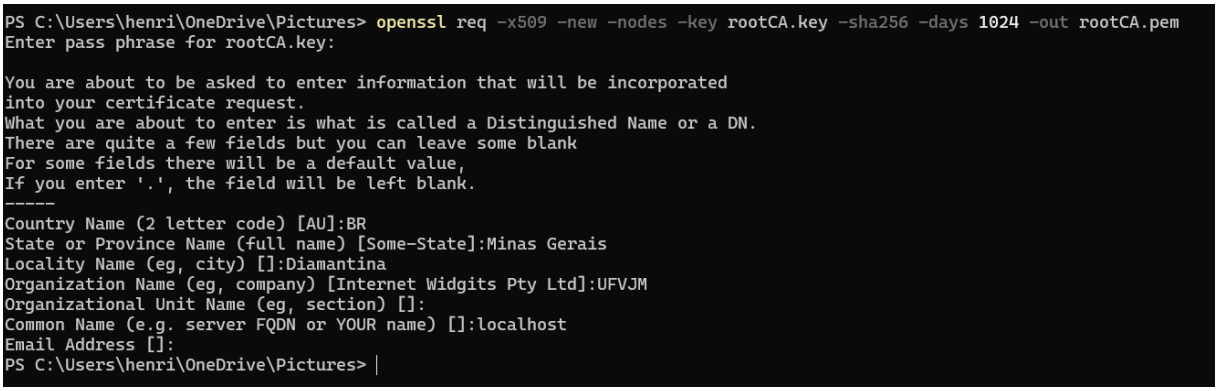

Figura 3: Geração do certificado SSL root no terminal cmd do windows

Com o certificado SSL root em mãos, precisaremos indicar para a nossa máquina que ela deve confiar nele, antes que possamos utilizá-lo para assinar outros certificados de domínio.

No windows, execute o comando `Win + R`, digite `mmc` e pressione `Enter`. Essas ações irão tornar visível o seu Console de Gerenciamento da Microsoft. Uma vez aberto, siga os passos a seguir:

1. No `Console de Gerenciamento`, clique em `Arquivo > Adicionar/Remover Snap-in`

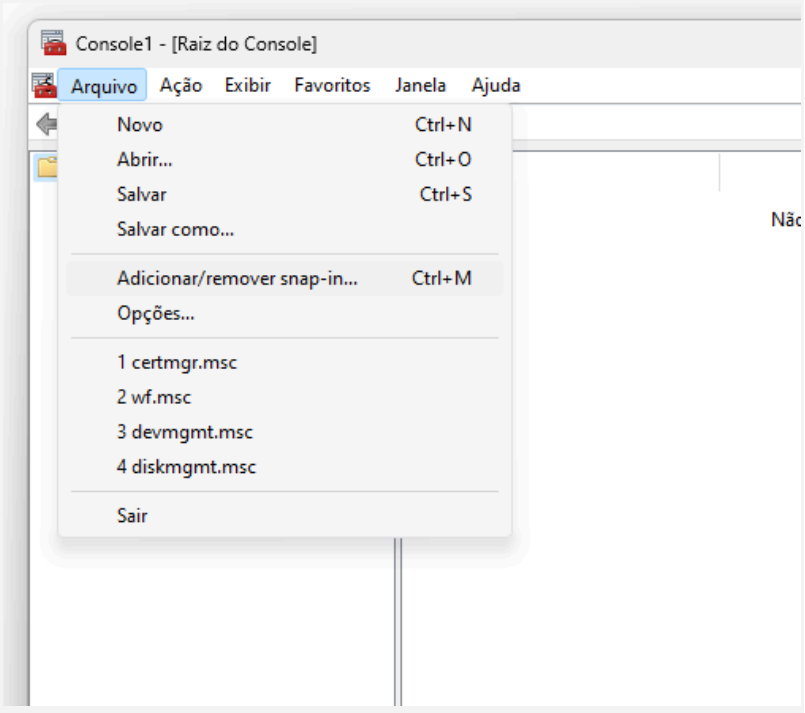

Figura 4: Console de Gerenciamento

2. Na janela `Adicionar ou Remover Snap/ins`, selecione `Certificados e clique em Adicionar`

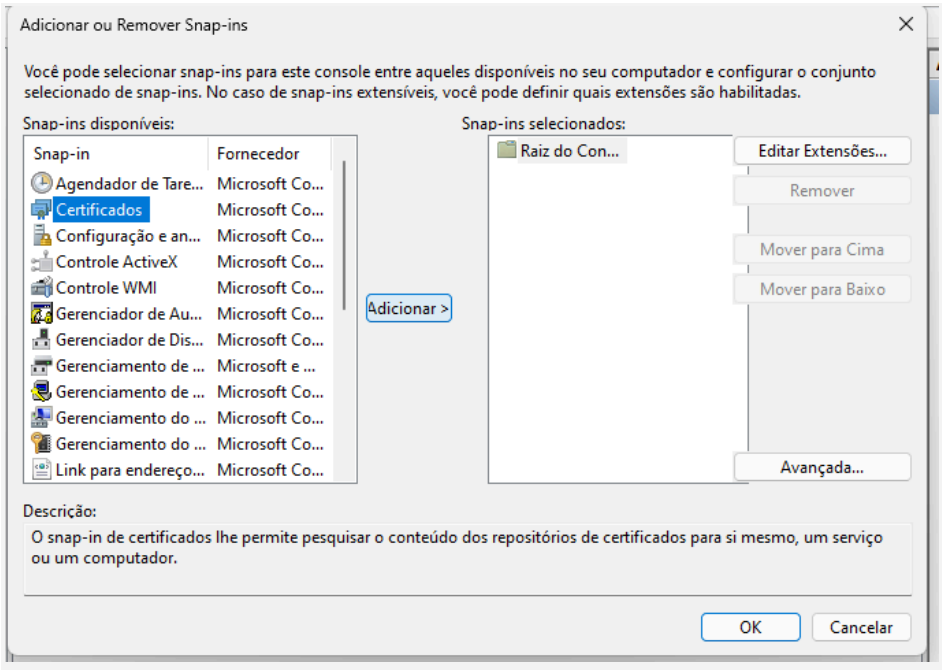

Figura 5: Adicionando Snap/in de certificado

3. Escolha a opção `Conta do Computador` e clique em `Avançar`

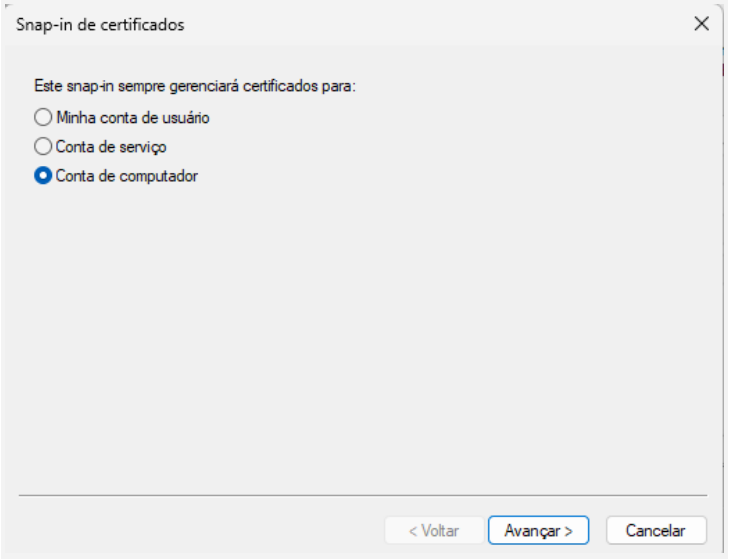

Figura 6: Adicionando snap-in de certificado

4. Escolha a opção `Computador Local` e clique em `Concluir`

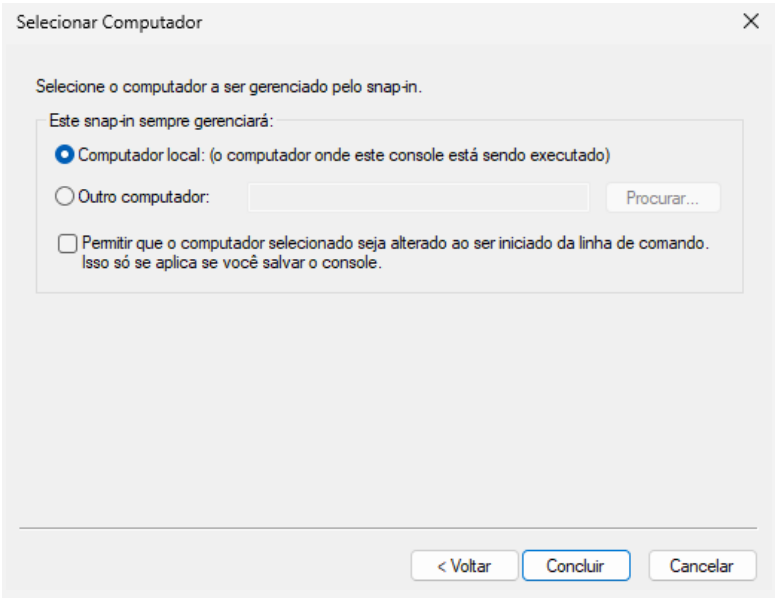

Figura 7: Adicionando snap-in de certificado

5. Clique em `OK` para fechar a janela
6. No painel esquerdo, expanda `Certificados (Local Computer)`. Clique com o botão direito do mouse em `Autoridades de Certificação Raiz Confiáveis`, aponte para `Todas a Tarefas` e, em seguida, clique em `Importar`

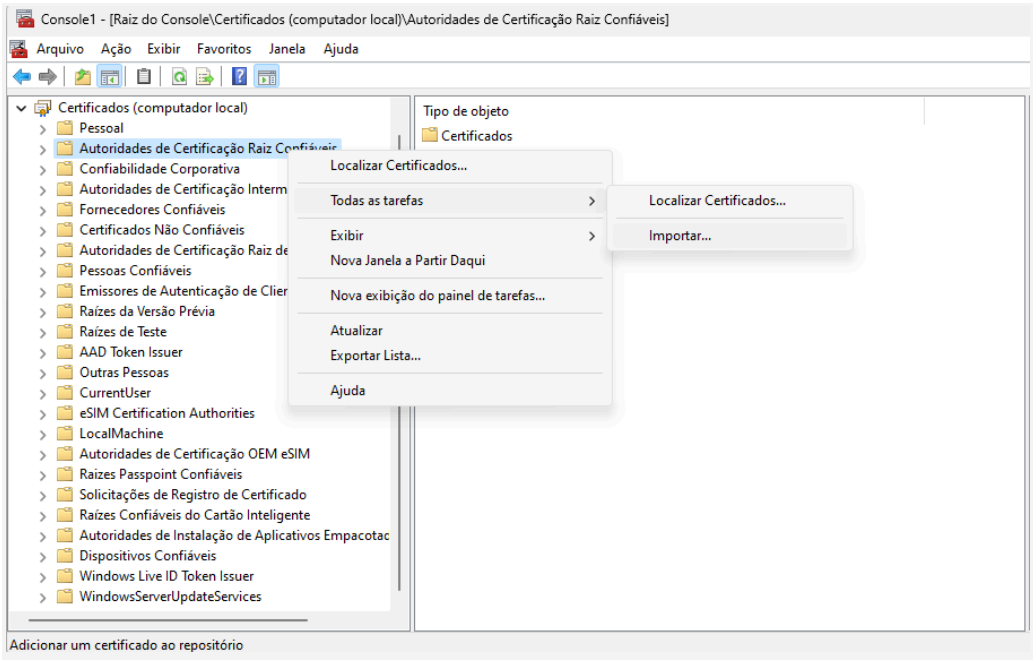

Figura 7: Importando certificado

7. No `Assistente de Importação de Certificados`, clique em `Avançar`

8. Clique em `Procurar`, navegue até o diretório onde o certificado foi gerado e selecione o arquivo `rootCA.pem` (se ele não estiver sendo exibido, mude a configuração de exibição para todos os arquivos no canto inferior direito).

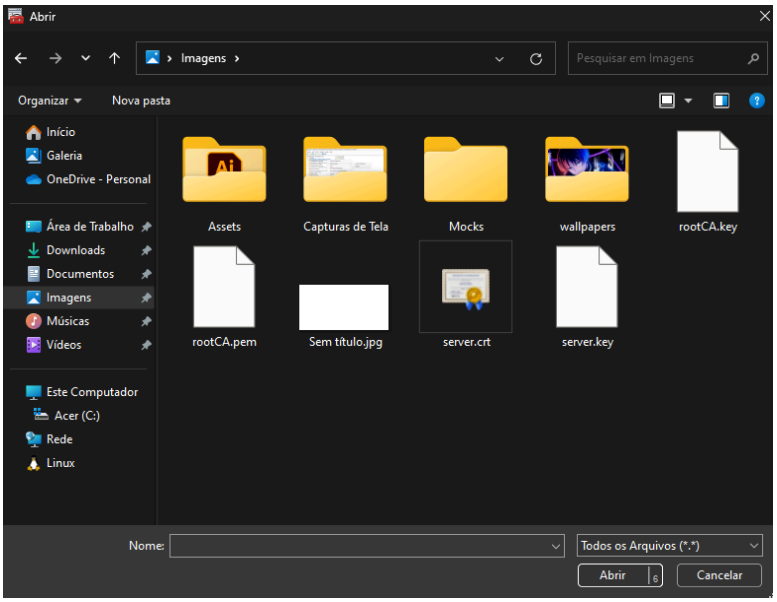

Figura 8: Importando certificado

9. Clique em `Avançar`, selecione `Colocar todos os certificados no ‘repositório a seguir` e verifique se `Autoridades de Certificação Raiz Confiáveis` está selecionado

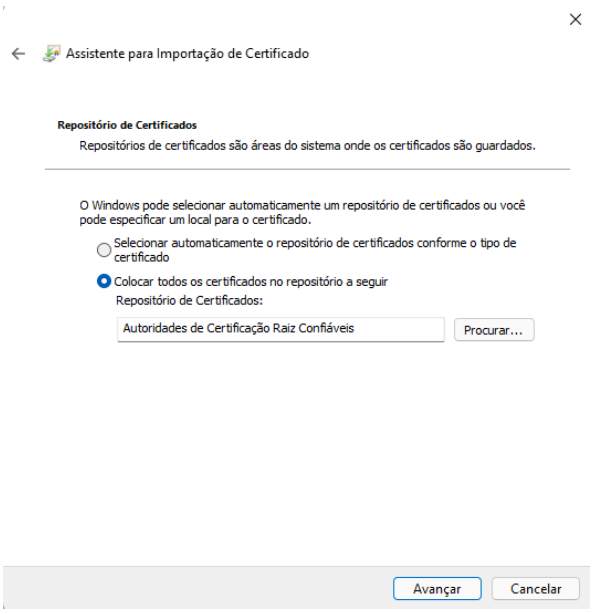

Figura 9: Importando certificado

10. Clique em `Avançar` e depois em `Concluir`

11. Caso você receba algum aviso de segurança, selecione a opção `Sim` para adicionar o certificado ao repositório
12. Clique em `OK` para fechar a caixa de diálogo de conclusão do assistente

3. Gerando um certificado SSL do domínio

Com a nossa máquina se comportando como uma autoridade certificadora, podemos emitir um certificado especificamente para o nosso ambiente local de desenvolvimento `localhost`.

> **Atenção!!** Certifique-se de estar no mesmo diretório que os arquivos de chave privada e certificado root estão, uma vez que precisaremos acessá-los para gerarmos nosso certificado de domínio e chave pública.

Primeiramente, crie um arquivo de configuração chamado `server.csr.cnf` e insira as configurações a seguir em seu conteúdo. Lembre-se de preencher com os dados corretos.

```
[req]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn

[dn]
C=BR
ST=[Seu estado]
L=[Sua cidade]
O=[Sua organização]
OU= [Sua organização]
emailAddress=dev@gmail.com
CN = localhost
```

***OBSERVAÇÃO**: Lembre-se de deixar a extensão do arquivo como .cnf. Se você o criou num bloco de textos, ele provavelmente terá a extensão .txt por padrão. No windows, exiba as extensões dos arquivos, clique para renomear e apague o ".txt" que aparece no final. Uma janela de confirmação será exibida. Aceite a modificação.*

Feito isso, crie um arquivo `v3.ext` com o seguinte conteúdo:

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment,
dataEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = localhost
```

***OBSERVAÇÃO**: Lembre-se de deixar a extensão do arquivo como .ext. Se você o criou num bloco de textos, ele provavelmente terá a extensão .txt por padrão. No windows, exiba as extensões dos arquivos, clique para renomear e apague o ".txt" que aparece no final. Uma janela de confirmação será exibida. Aceite a modificação.*

Agora que temos os arquivos de configuração necessários, iremos criar uma chave de certificado. Para isso, execute a linha de comando a seguir no seu terminal:

```
openssl req -new -sha256 -nodes -out server.csr -newkey rsa:2048 -keyout
server.key -config server.csr.cnf
```

Figura 10: Gerando chave de certificado pelo terminal cmd do Windows

Com a chave criada, iremos solicitar a assinatura de um certificado que será feito pelo certificado root que criamos. Esse processo gerará um arquivo `server.crt` que representa o nosso certificado de domínio. Para isso, execute o comando a seguir no seu terminal e insira a senha da sua chave privada gerada nos primeiros passos deste tutorial.

```
openssl x509 -req -in server.csr -CA rootCA.pem -CAkey rootCA.key
-CAcreateserial -out server.crt -days 500 -sha256 -extfile v3.ext
```

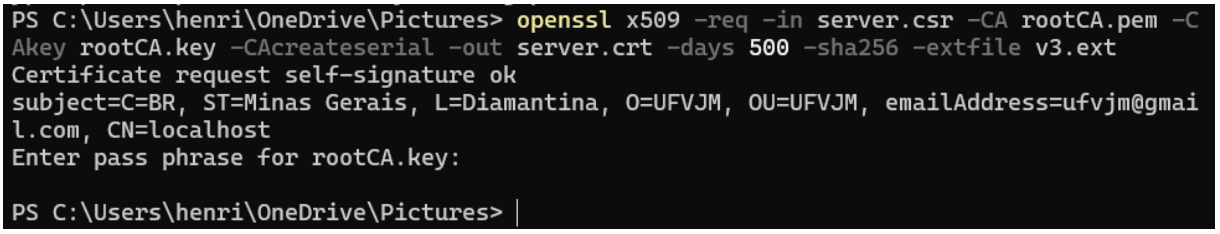

Figura 11: Requisição de assinatura do certificado

4. Utilizando seu certificado de domínio

Com a execução dos passos anteriores, você terá em seu diretório os arquivos `server.crt` e `server.key`. Esses arquivos poderão, então, ser importados para o seu projeto para habilitar a Certificação Digital, que será lida pelo seu navegador como segura e permitirá o uso do protocolo HTTPS na sua navegação local. Para isso, você deve consultar a documentação da linguagem que está desenvolvendo o seu projeto e verificar como deve ser feita a importação das chaves e do certificado.

Referências

ROSA, D. Como fazer o HTTPS funcionar em seu ambiente local de desenvolvimento em 5 minutos. Disponível em: <https://www.freecodecamp.org/portuguese/news/como-fazer-o-https-funcionar-em-seu-ambiente-local-de-desenvolvimento-em-5-minutos/>. Acesso em: 26 maio. 2024.

STRAIGHT TO CODING. How to install OpenSSL on Windows. Disponível em: <https://www.youtube.com/watch?v=coaGBdUcKiw>. Acesso em: 26 maio. 2024.