# Mutual TLS

Este tutorial tem como foco demonstrar o passo a passo para geração dos certificados necessários para a utilização do mutual TLS na comunicação entre cliente / servidor.

## Cenário Ideal

* O servidor gera sua rootCA e com isso sua CA intermediária que será utilizada para assinar o Certificado de Solicitação de Assinatura (CSR) do cliente.
* O Cliente gera a KEY e o CSR e envia apenas o CSR para o servidor.
* Servidor assina o CSR com a CA intermediária gerando o CRT do cliente.
* Cliente utiliza o CRT nas requisições para o servidor.
* Servidor verifica nas requisições recebidas se o CRT do cliente é válido através da ca-chain, que nada mais é que a cadeia de confiança gerada entre a rootCA e a CA intermediária. (Arquivo .cert.pem com as hash do rootCa e intermediateCA).

**OBS**: o ideal é que o processo descrito acima seja realizado também para o lado do cliente. Sendo assim, o cliente irá assinar o CSR do servidor para também verificar o caminho inverso.

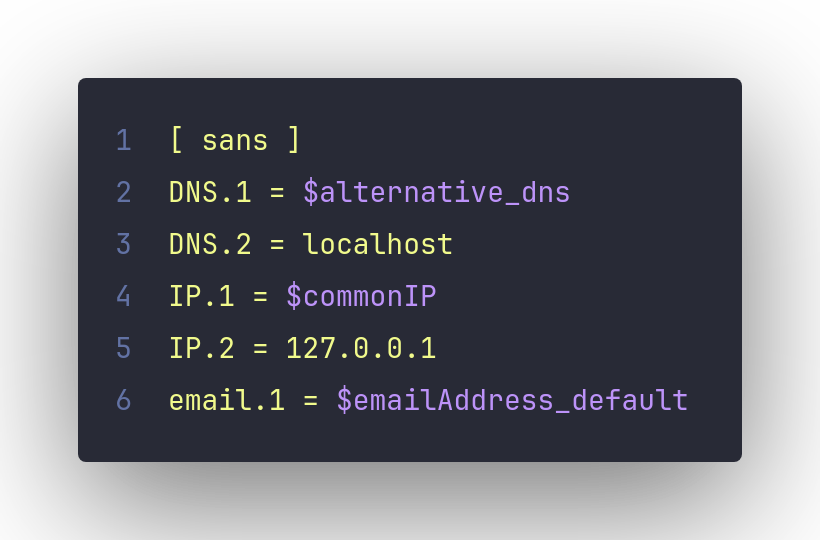
***#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#***

## Configurando os Subject Alternative Names (SANs)

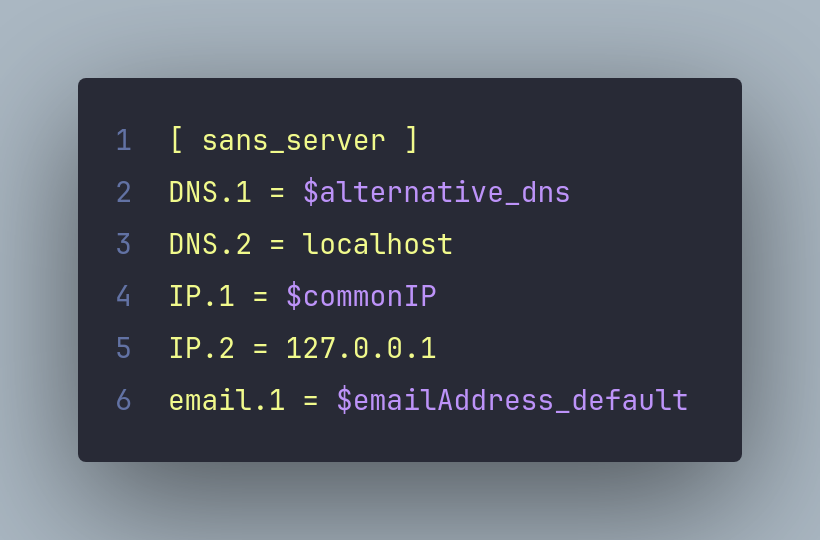
Os Subject Alternative Names (SANs) em certificados SSL/TLS são usados para especificar quais nomes de domínio (além do nome de domínio principal) são considerados válidos para o certificado. Por exemplo, se um certificado SSL/TLS é emitido para o domínio "example.com", mas também precisa cobrir "www.example.com" e "secure.example.com", os SANs podem ser usados para incluir esses nomes alternativos no certificado. Isso permite que o certificado seja aceito por navegadores e outros clientes quando eles acessam o site usando esses nomes alternativos.

Além disso, os SANs também podem incluir endereços IP, o que é útil em cenários onde um servidor pode ser acessado por meio de um nome de domínio e um endereço IP.

Dito isso, entre no arquivo scripts/create-root-openssl-cnf.sh e adicione outros domínios além do CN e IPs que serão validados pelo seu certificado. Por padrão o arquivo está preenchido com localhost, o IP e domínio alternativo passado opcionalmente nos parâmetros ao rodar o script.



Faça o mesmo no arquivo scripts/create-intermediate-openssl-cnf.sh.



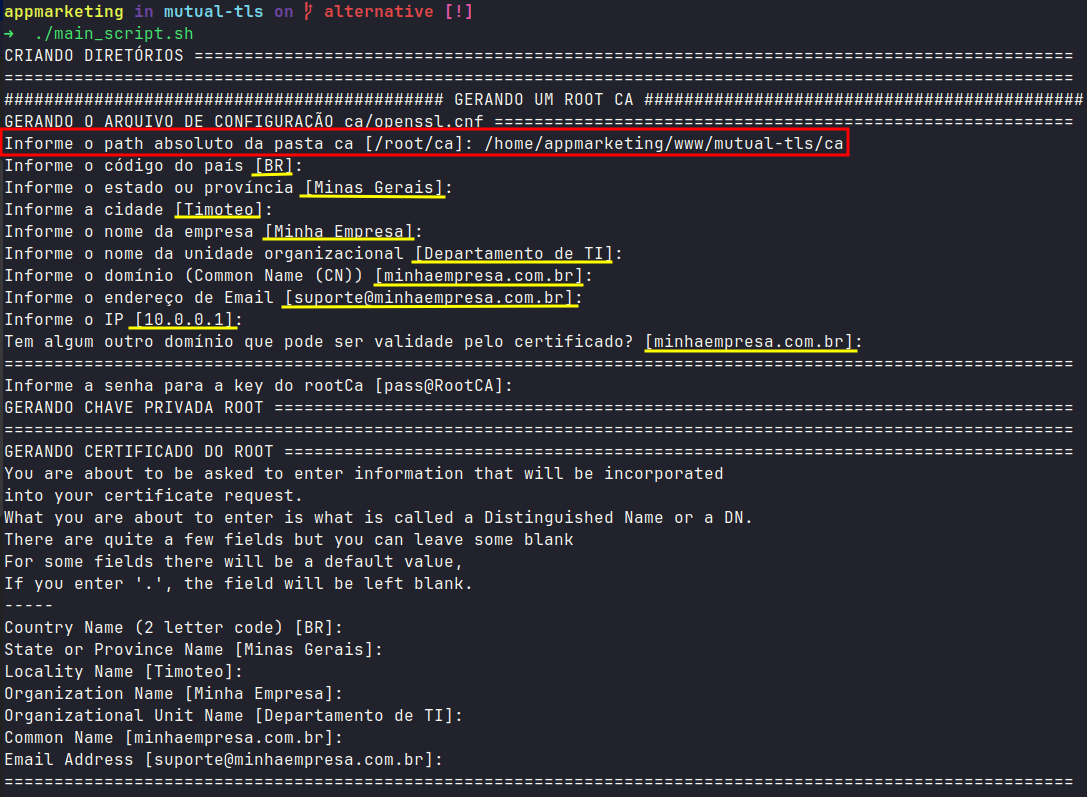
**Gerando a RootCA, IntermediateCA e CA-CHAIN**

Após realizar a configuração dos SANs, vamos executar o script main\_script.sh paragerar nossa RootCA, a IntermediateCA e a cadeia de confiança (ca-chain). O main\_script.sh criará os diretórios adequados para armazenar os arquivos gerados, além disso irá chamar os scripts de dentro da pasta scripts na ordem correta para gerar os certificados.

Observe que durante a execução, aparecerá algumas perguntas no terminal, certifique-se de respondê-las corretamente. Vamos percorrer o passo a passo da execução do script:

### Gerando RootCA:

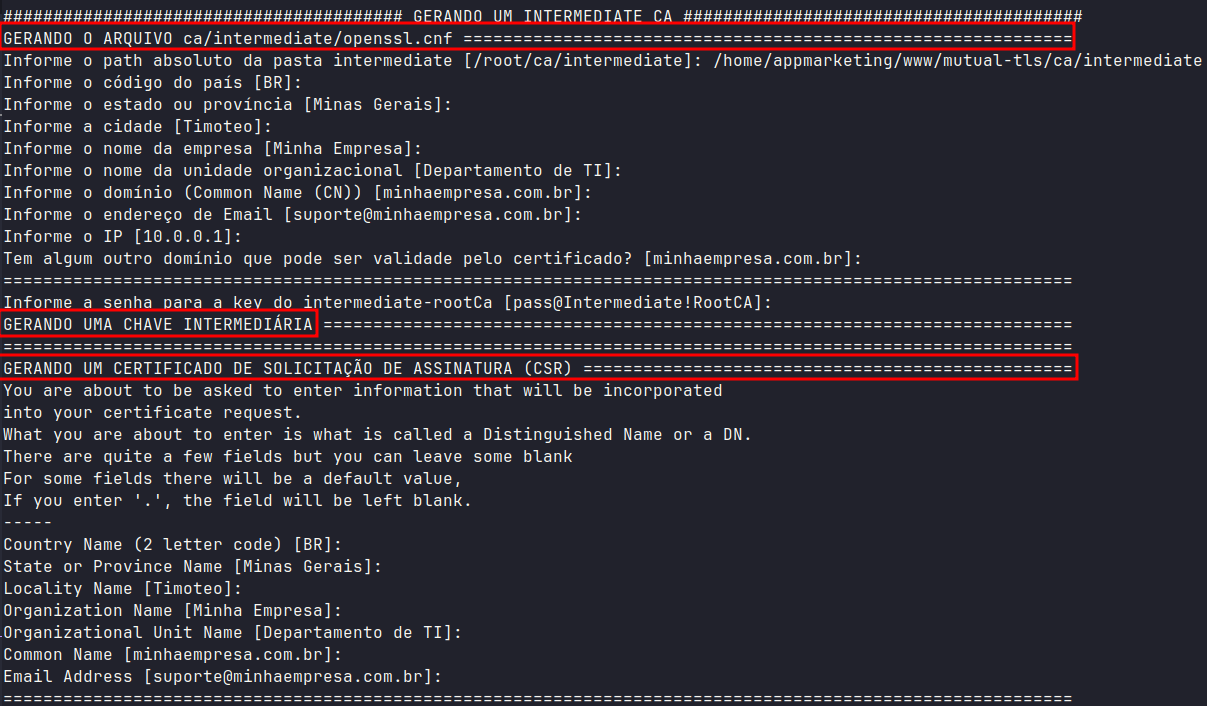
Inicialmente o script pedirá informações para gerar o arquivo de configuração ca/openssl.cnf a chave e o certificado RootCA:



Note que informei apenas o path da pasta CA, o restante mantive os valores setado como padrão que podem ser identificados dentro dos colchetes em cada pergunta.

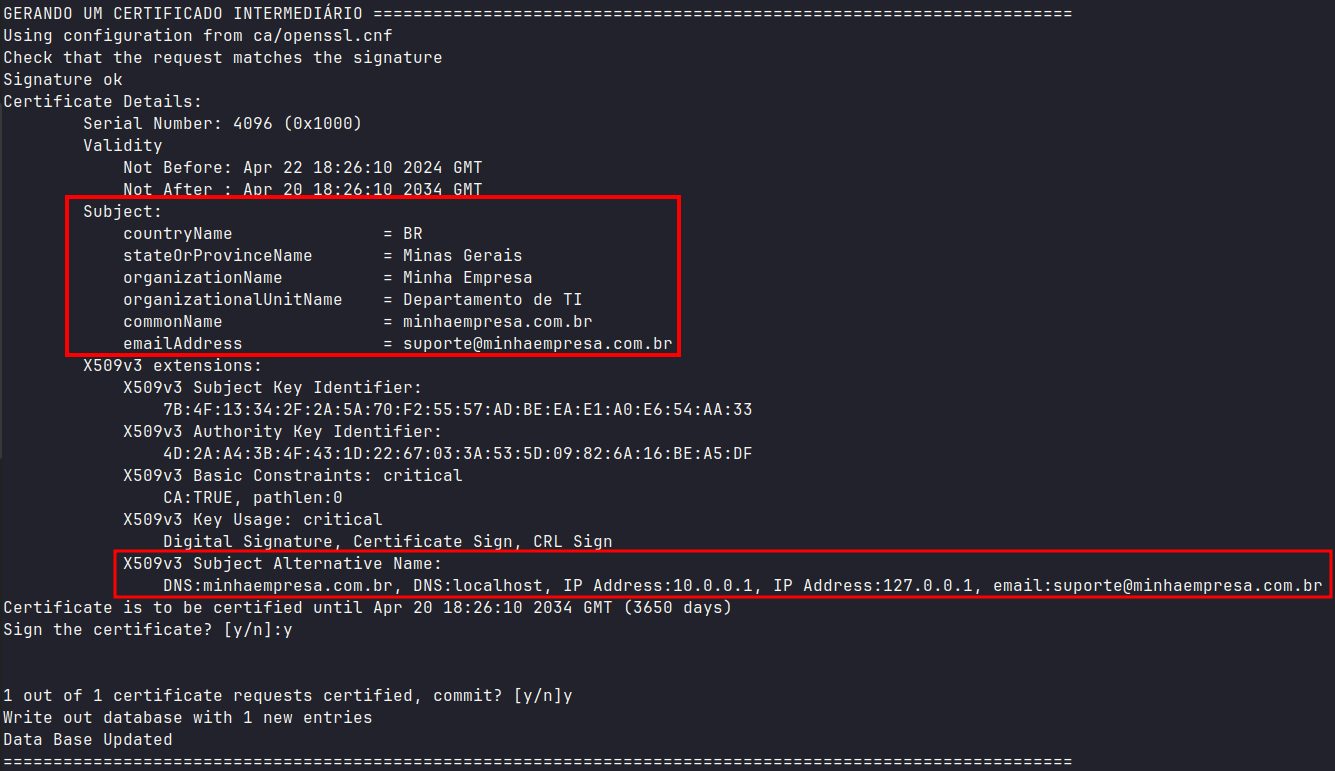
### Gerando IntermediateCA:

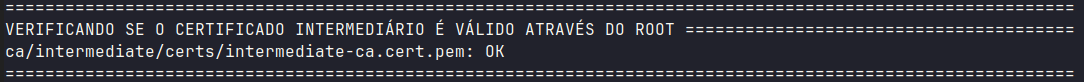
Após a geração do certificado RootCA, o script pede as informações para gerar o arquivo ca/intermediate/openssl.cnf a chave e o Certificado de Solicitação de Assinatura (CSR) da CA intermediária:



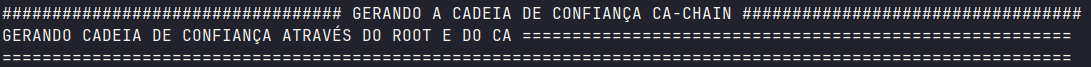
Nesse caso também informei apenas o path da pasta “intermediate”, mantendo as outras informações com o padrão de exemplo.

Após gerar o CSR, o script irá assiná-lo com a rootCA e verificar sua autenticidade. Sendo assim será gerado o certificado da CA intermediária com as informações passadas e os Subject Alternative Name (SANs) configurados nos arquivos inicialmente.



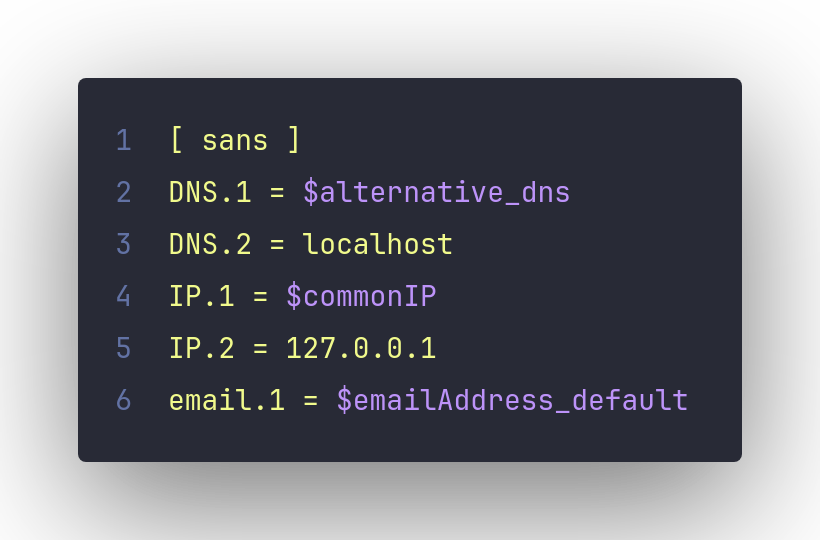


Após gerar e verificar a CA intermediária, é gerada a cadeia de segurança que será utilizada para validar os certificados dos clientes.

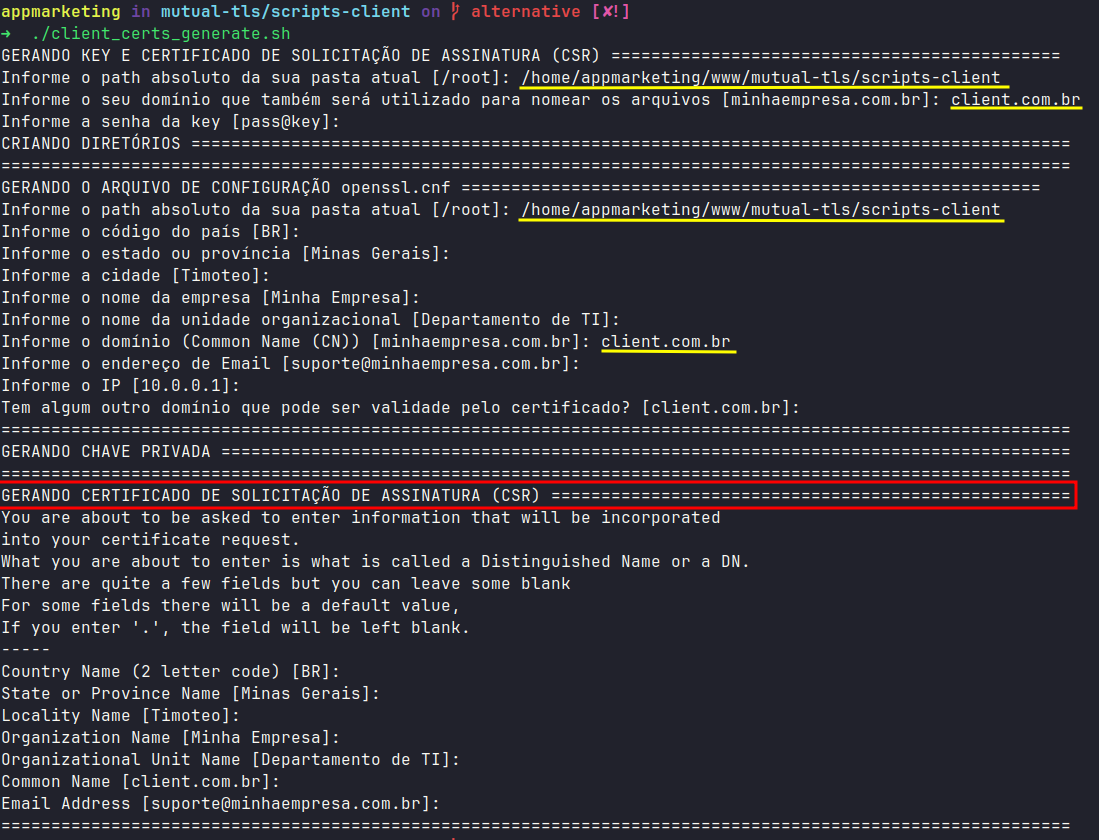


## Gerando CSR para o cliente

Para gerar o CSR do cliente, primeiro realize a configuração do SANs no script scripts-client/scripts/create-openssl-cnf.sh:



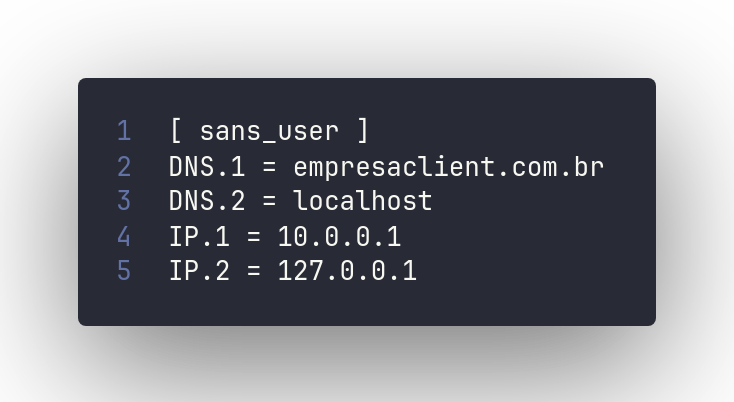
Após configurar o SANs, execute o script scripts-client/client\_certs\_generate.sh para gerar o Certificado de Solicitação de Assinatura que será assinado pela CA intermediária do servidor.



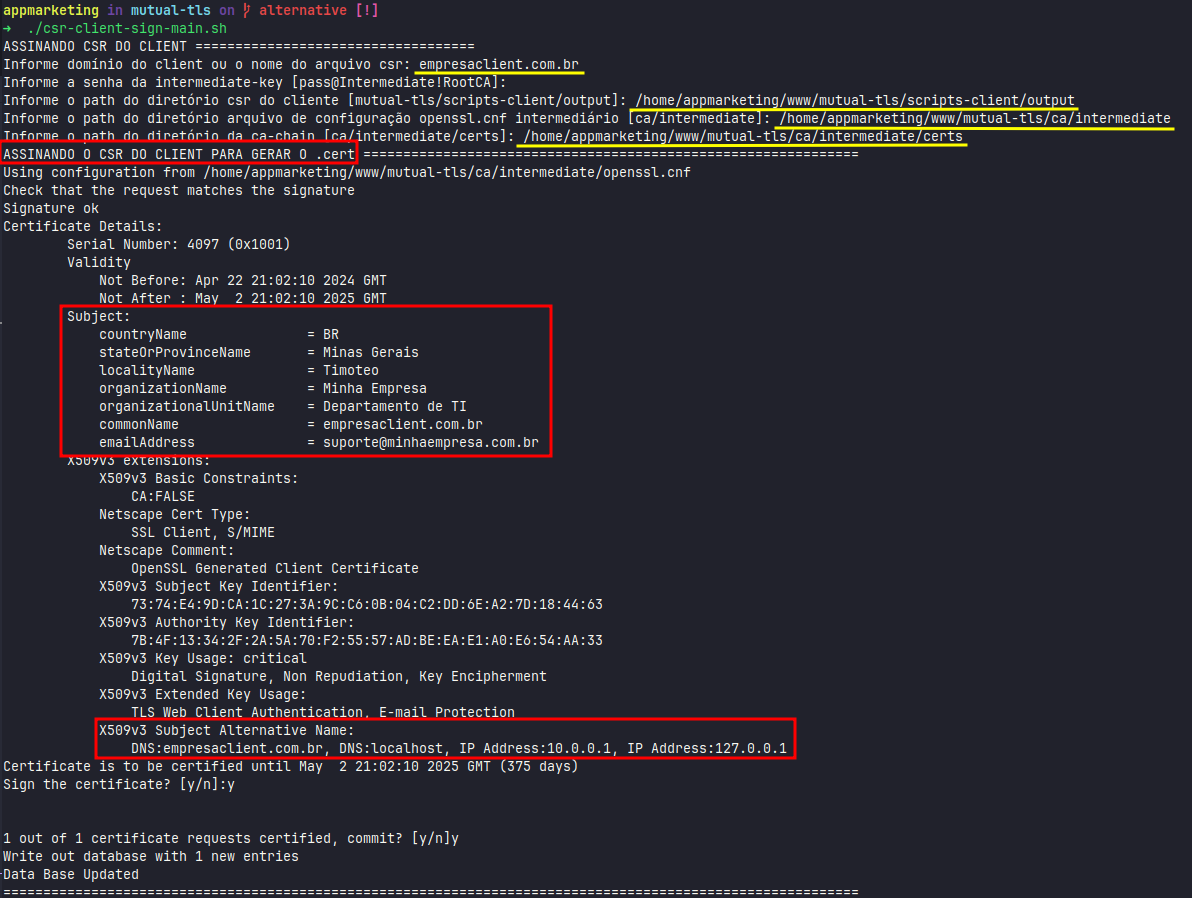
Nesse processo, configurei somente o path da pasta onde estão os scripts para geração do CSR do cliente e também um domínio de exemplo para o cliente.

**Assinando CSR do cliente**

Antes de rodar o script para assinar o CSR do cliente, configure o SANs da extensão [sans\_user] no arquivo ca/intermediate/openssl.cnf informando o IP do servidor do cliente e os domínios alternativos caso possua:



Apos configurar o [sans\_user], basta executar o script csr-client-sign-main.sh e informar os dados solicitados:



Para realizar a criação do CA do cliente e o CSR do servidor para o caminho inverso, basta realizar o mesmo processo com os dados do lado do cliente!