

# Relatório desafio De-Backend

Marcos Tomaszewski

## 1: Certificados x Java

Como nunca havia trabalhado com certificados digitais, comecei o desenvolvimento buscando a biblioteca sugerida pelo desafio como ponto inicial, a Bouncy Castle, contudo ao encontrar a documentação desta, que é bem sucinta e notei que seria extremamente necessário buscar mais informações sobre a tarefa. Após algumas leituras na web, na documentação do padrão e análise de exemplos encontrados na documentação da biblioteca e em outros sites, comecei a prática com um tutorial do site *baeldung.com* aonde aprendi o básico do manuseio dos arquivos de certificado, bem como de chaves e seus geradores.

Como achei uma grande quantidade de exemplos da biblioteca OpenSSL no meio das leituras e esta pareceu ser simples, gerei e assinei alguns certificados usando ela.

Com uma maior confiança no manuseio de chaves e certificados, passei um bom tempo analisando um exemplo de uso da biblioteca Bouncy Castle do usuário *vivekkr12* do *github*, desmanchei cada linha, vasculhei em busca de documentação para cada método. Com uma familiaridade bem maior com a biblioteca, criei um código que gerava e assinava um certificado para o um nome e validades pré-definidas, apesar da extensiva leitura e de ser uma das partes com maior tempo dispendido no desafio, eu conseguia gerar certificados x509 e assiná-los.

## 2: Back-end

Como não tenho muita experiência com boa parte das tecnologias de desenvolvimento WEB, comecei um projeto via Spring Boot, novamente seguindo tutoriais do site *baeldung.com*, desenvolvi um simples "Hello World" e integrei o código desenvolvido na primeira etapa, que agora gerava e assinava os certificados via solicitação web.

## 3: Banco de Dados

Apesar de já ter certa experiência com linguagens SQL, eu nunca havia usado o SQLite, após uma rápida lida na documentação entendi ideia básica e gostei da proposta, um banco de dados inteiro em um arquivo, comecei um *DB* e aproveitei o terminal para tirar a poeira do meu SQL. Aproveitei o *back-end* que eu tinha feito e criei alguns *Mappings* que manipulam o banco do SQLite.

Como não faria sentido continuar a desenvolver o *back-end* sem ter ao menos um *front-end* mínimo, suspendi o desenvolvimento do *back-end* para começar um *front-end* básico.

## 4: Front-end

Novamente de frente para o (pouco) desconhecido, segui o tutorial em *reactjs.org* para criar uma aplicação de jogo da velha, que não durou muito tempo, porque logo fui mudando a estrutura para atender melhor a ideia básica: campos de texto e data, em seguida uma lista com os certificados. Com o esqueleto pronto, busquei algo que me permitisse fazer requisições HTTP e encontrei o *axios*, logo depois de uma pequena porém épica batalha com um erro causado por *CORS*, meu front-end simplório estava fazendo requisições para o meu back-end, que respondia criando um certificado para cada solicitação.

## 5: Melhorias no Cliente-Servidor

Com uma base já desenvolvida, fui alterando o *front-end* para deixá-lo minimamente apresentável, alterei o texto, incluí a biblioteca *bootstrap* para facilitar a estilização, adicionei campos de calendário. Genuinamente confuso com os hooks do React demorei um tempo até que consegui enviar a data para o servidor, permitindo finalmente escolher a validade dos certificados.

Altere o código desenvolvido na parte 3 para que após o Servidor gerar o certificado, ele já o incluísse no banco de dados, para isso criei uma tabela, desenvolvi a *query* e por fim, o certificado estava sendo armazenado no banco.

Fiquei um tempo testando a geração e armazenamento dos certificados, descobri dois problemas: Estava salvando o expoente e o módulo da chave pública no banco e levei um tempo até por fim, armazenar a *PKI* no banco, permitindo uma melhor visualização da chave; também notei que as datas dos certificados estavam sendo geradas corretamente, porém estavam sendo tratadas incorretamente para serem armazenadas no banco, após um *parsing* o problema foi solucionado.

Voltei ao front-end para desenvolver uma opção de busca de certificados do banco, terminei de montar a tabela que eu havia deixado pendente na etapa anterior, após desenvolver uma busca primitiva, sem parâmetros, adicionei uma opção para deleção unitária dos registros, e em seguida, adicionei a opção de filtrar a busca no banco.

Para finalizar adicionei uma opção de seleção múltipla e um botão para confirmar a deleção dos registros selecionados.

## 5: Finalizando o código

Revisei todos os arquivos do projeto, melhorando onde possível. Novamente voltei a fazer melhorias estéticas no front-end que estava carecendo de mais suavização.

## 6: Resumo

Existem algumas situações que eu gostaria de solucionar porém o tempo não permite, particularmente, a que ocorre ao remover um ou vários certificados, onde a lista é limpa, gostaria também de melhorar a apresentação dos campos da consulta ao banco, e por fim de fazer o deploy via docker, contudo o tempo, só não menos limitado que meu conhecimento em docker, me impede.