

Processo Criativo – Desafio Técnico API Customers

Nome: Matheus Teles Marques

Data: 02/09/2025

Desenvolvimento da API de Customers

- **Contextualização do projeto**

O projeto é uma API REST para gerenciamento de clientes, permitindo operações de criação, leitura, atualização e exclusão de dados. O objetivo era construir um sistema escalável, seguro e fácil de manter, utilizando práticas modernas de desenvolvimento em Java.

- **Planejamento inicial**

Antes de iniciar a implementação, planejei o fluxo de dados e os endpoints principais. Defini a entidade “Customers” e os seus DTOs para separar dados de entrada e saída. Estructurei o mapeamento JPA da entidade Customer e defini os endpoints CRUD.

- **Decisões técnicas**

O uso do Spring Boot, Spring Security e JWT foi definido pelo desafio, então minha principal tomada de decisão técnica ficou na escolha da persistência de dados e ferramentas de teste. Optei por utilizar PostgreSQL como banco de dados principal por sua robustez e compatibilidade com produção, e H2 como banco em memória para testes, garantindo rapidez e isolamento nos testes.

Implementei testes unitários e de integração para validar o comportamento da API, cobrindo operações de CRUD, busca por atributos e tratamento de exceções. Para os testes unitários, utilizei JUnit e Mockito, garantindo que cada componente isolado se comportasse conforme esperado. Para os testes de integração, utilizei MockMvc, simulando requisições HTTP para os endpoints e verificando o fluxo completo da API, incluindo serialização/deserialização de JSON e autenticação via JWT. Os testes alcançaram 82% de cobertura de código e 62% de cobertura de ramificações (branches), garantindo que diferentes cenários fossem validados e aumentando a confiabilidade da aplicação antes do deploy. Além disso, utilizei Swagger para documentação da API, tornando mais fácil testar e compreender os endpoints, tanto para desenvolvedores quanto para avaliadores do desafio.

- **Fluxo de desenvolvimento e resolução de problemas**

Comecei definindo a entidade principal, mapeando-a com JPA, criando os DTOs e implementando o CRUD completo. Somente depois comecei a pesquisar sobre

Spring Security, JWT e testes, áreas nas quais eu tinha pouca experiência. Durante o desenvolvimento, enfrentei algumas dificuldades, principalmente na geração e validação do JWT. Por se tratar de uma parte complexa, optei por enviar os dados via JSON no endpoint de login, garantindo a autenticação de forma consistente. O backend então autentica o usuário e gera o JWT, que pode ser utilizado via Bearer Authentication para acessar outras funcionalidades da API.

O deploy também foi uma etapa desafiadora. Durante a publicação da API, surgiram erros relacionados à conversão da chave privada e pública de Base64 para String, o que exigiu pesquisa e ajustes nos arquivos de configuração e propriedades. Apesar dos obstáculos, essa experiência foi extremamente enriquecedora, pois me permitiu compreender melhor questões de segurança, configuração de variáveis de ambiente e particularidades do Spring Boot em ambiente de produção.

- **Reflexão final**

O processo de desenvolvimento da Customer API foi uma experiência de aprendizado incrível. Reforcei boas práticas de design de APIs REST, aprofundei meus conhecimentos em Spring Boot, Spring Security, JWT e testes de integração e unitários. Além disso, desenvolvi habilidades para resolver problemas reais de deployment, o que ampliou minha capacidade de análise e solução de erros em ambientes de produção.