



# Certified Tech Developer

The Ultimate Degree

## Especialização Back end Java

# Exercício Mesa de trabalho

A empresa “Supermercado Bonanza” tem reclamado da indisponibilidade constante de seu serviço de estoque e nos pediu que criássemos um sistema mais confiável com a possibilidade de monitorar o tempo de resposta entre usuário e serviço. Vamos criar um microserviço para persistir o estoque de produtos, um microserviço para recuperar e adicionar novos produtos, um simples circuit breaker e um service discovery (servidor Eureka, já criado).

## Dependências

- Estoque e Repositório:
  - Spring Boot Starter Web;
  - Spring Boot Starter Data JPA;
  - Spring Boot Starter JDBC;
  - Spring Boot Autoconfigure;
  - Spring Cloud Starter Netflix Eureka Client;
  - Spring Cloud Starter OpenFeign;
  - H2 Database;
  - Lombok.
- Circuit Breaker:
  - Spring Cloud Starter Circuitbreaker Reactor Resilience4j;
  - Resilience4j Spring Boot2;
  - Spring Cloud Starter Gateway;
  - Spring Cloud Starter Netflix Eureka Client.



## Instruções

1. Crie o arquivo docker-compose.yml e inicie o servidor zipkin;
2. Crie o microserviço "estoque" com uma REST API contendo os endpoints necessários para recuperar um produto e salvar um novo produto;
3. Crie o microserviço "repositório" com uma REST API para consumir o microserviço de estoque;
  - 3.1. O serviço "repositório" deve consumir o serviço "estoque" através do OpenFeign;
4. Crie o circuit breaker garantindo o roteamento e conexão dos dois microserviços anteriores;
5. Ao final, identifique (com base no exercício guiado anterior e no conteúdo da aula 16) quais são as duas dependências que, quando adicionadas no pom.xml dos 4 serviços, permite a observabilidade através do nosso servidor zipkin.