Teste para vaga de Java SR

**Serviço 1 (Produto):**

- Criar uma API com Java, Spring Boot Web Flux ou Quakus + Mongo Db + Kafka ou Rabbit para atender aos requisitos abaixo

- Modelo de arquitetura a escolha do candidato

**Entidades:**

- Categoria (id, nome e descrição)

- Produto (id, nome, descrição, quantidade e valor)

- Carrinho (produtos, quantidade, data de criação e data de alteração)

- Pedido (carrinho, data de criação, status=[WAIT\_PAYMENT, DONE, ERROR\_PAYMENT]

**Endpoints:**

- Buscar produtos por Categoria

- Buscar pedido por Id do Carrinho

- Criar Carrinho

- Excluir Carrinho

- Incluir Produto ao Carrinho

- Excluir Produto ao Carrinho

- Criar Pedido

**Produtor:**

- Ao criar um Pedido enviar uma mensagem para uma fila ou tópico

**Consumidor:**

- Criar um serviço que consuma o retorno do serviço de Pagamento

**Regras:**

- Criar uma massa de produtos e categorias para facilitar o teste e desenvolvimento

- Somente listar produtos com quantidade disponível

- Ao fechar concluir um pedido decrementar a quantidade do produto

- Ao consumir a mensagem de pagamento caso tenha erro incrementar a quantidade do produto

- Ao realizar o pedido verificar se tem disponibilidade do produto

**Serviço 2 (Pagamento):**

- Criar uma API com Java, Spring Boot Web Flux ou Quakus + Mongo Db + Kafka ou Rabbit para atender aos requisitos abaixo

**Consumidor:**

- Criar um serviço que consuma a mensagem de pedido criado

- Criar um método para simular um meio de pagamento com retorno de status randômico de sucesso ou erro

- Emitir um evento com o status do pagamento pro determinado pedido