

Atividade Prática: Sistema de Processamento de Pedidos com Comunicação Assíncrona

Objetivo da Atividade

Desenvolver uma aplicação distribuída simples que simule o processamento de pedidos em um sistema de e-commerce, utilizando **mensageria** e **comunicação assíncrona** para garantir **resiliência**, **tolerância a falhas** e **desacoplamento** entre componentes.

Descrição do Projeto

O projeto simulará o fluxo de pedidos em uma loja virtual com os seguintes componentes distribuídos:

1. Serviço de Pedidos (API REST):

- a. Responsável por receber pedidos de clientes.
- b. Cada pedido deve conter: nome do cliente, itens, quantidade, valor total.

2. Fila de Processamento (ex: RabbitMQ, Kafka, Redis Streams):

- a. O serviço de pedidos envia mensagens para a fila contendo os dados do pedido.
- b. As mensagens devem ser persistidas até que sejam consumidas com sucesso.

3. Serviço de Processamento de Pedidos (Consumidor):

- a. Consome as mensagens da fila.
- b. Simula ações como: validação do pedido, geração de nota fiscal e atualização de estoque.
- c. Pode gerar logs ou armazenar os resultados processados em um arquivo ou banco de dados.

4. Serviço de Notificação (opcional):

- a. Após o processamento, uma nova mensagem pode ser enviada para uma fila de notificações.
- b. Um serviço consumidor dessa fila envia uma notificação (simulada via log ou email fake).

Validação do Projeto

O projeto será validado por meio de:

- **Demonstração prática (obrigatória)** em sala ou vídeo de no máximo 10 minutos, com:
 - Simulação de falhas (ex: desligar consumidor e mostrar que as mensagens não se perdem).
 - Demonstração de mensagens fluindo da API até o processamento final.
- **Entrega de código-fonte** em repositório (GitHub ou GitLab), com instruções de execução.
- **Relatório técnico (máximo 3 páginas)** com:
 - Arquitetura adotada (com diagrama).
 - Descrição da troca de mensagens.
 - Estratégias de tolerância a falhas (ex: retries, persistência, logs).
 - Tecnologias utilizadas.

Critérios de Avaliação (Nota 0 a 10)

Critério	Pontuação Máxima	Detalhes
Funcionamento da comunicação assíncrona	3,0 pts	Uso correto de fila, consumo e produção de mensagens, entrega garantida
Desacoplamento entre serviços	1,5 pts	API e processamento funcionando de forma independente e resiliente
Tolerância a falhas e resiliência	2,0 pts	Demonstração de que falhas não causam perda de pedidos; uso de logs, retries ou dead-letter queues
Qualidade do relatório e explicação da arquitetura	2,0 pts	Clareza, completude do diagrama e justificativas técnicas
Organização e funcionamento geral da aplicação	1,5 pts	Código funcional, bem organizado, com instruções claras para rodar

Tecnologias recomendadas (mas não obrigatórias)

- **Mensageria:** RabbitMQ, Apache Kafka, Redis Streams
- **Linguagens:** Java, Python, Node.js, Clojure, etc.

- **Frameworks:** Spring Boot, Express, FastAPI, etc.
- **Ambiente:** Docker Compose para orquestração dos serviços

Forma de Realização

- **Individual ou em dupla**
- Entregas via **Google Classroom:**
 - Link do repositório
 - Relatório em PDF
 - Link do vídeo da demonstração