

## Computação Distribuída – Aula 9 – Tarefa prática

Em arquitetura de microsserviços, **orquestração** e **coreografia** são abordagens diferentes para gerenciar a comunicação e coordenação entre serviços independentes. A escolha entre essas duas abordagens depende dos requisitos e da complexidade do sistema. Vamos entender cada uma por meio de uma situação prática.

### Situação: Processamento de Pedidos em um E-commerce

Imagine um sistema de e-commerce que possui vários microsserviços, incluindo **Catálogo de Produtos**, **Carrinho de Compras**, **Processamento de Pagamento**, **Envio** e **Notificação de Cliente**. Quando um cliente realiza um pedido, é necessário que esses serviços trabalhem em conjunto para garantir que o pedido seja concluído com sucesso.

#### Orquestração

Na abordagem de orquestração, um serviço central (o **Orquestrador**) coordena e controla o fluxo do processo. Vamos imaginar que existe um **Serviço de Orquestração de Pedidos** que gerencia todo o processo:

1. **Pedido Recebido:** O Serviço de Orquestração de Pedidos recebe o pedido e decide o que precisa ser feito em seguida.
2. **Execução Sequencial:** O orquestrador chama o serviço de **Processamento de Pagamento**. Quando o pagamento é aprovado, ele chama o serviço de **Envio**. Após o envio, ele chama o serviço de **Notificação de Cliente** para informar o cliente sobre o pedido.
3. **Controle Centralizado:** Todo o processo é controlado pelo orquestrador, que decide a sequência das operações e gerencia o status de cada etapa.

#### Vantagens:

- Controle centralizado e fácil de monitorar.
- Simples de implementar em processos lineares e bem definidos.

#### Desvantagens:

- Alta dependência de um serviço central, o que pode causar um ponto único de falha.
- Pode dificultar a escalabilidade, pois o orquestrador pode se tornar um gargalo.

#### Coreografia

Na abordagem de coreografia, cada microsserviço trabalha de maneira mais autônoma e reage a eventos que outros serviços publicam. Não existe um controlador central; ao invés disso, cada serviço sabe como reagir a determinados eventos.

1. **Evento de Pedido Criado:** Quando um pedido é criado, o serviço de **Pedido** emite um evento "Pedido Criado".
2. **Reação Autônoma:** O serviço de **Processamento de Pagamento** escuta o evento "Pedido Criado" e inicia o processo de pagamento. Após a aprovação, ele emite um evento "Pagamento Aprovado".
3. **Encadeamento por Eventos:** O serviço de **Envio** escuta o evento "Pagamento Aprovado" e processa o envio. Depois, emite um evento "Pedido Enviado".
4. **Notificação de Cliente:** O serviço de **Notificação de Cliente** escuta o evento "Pedido Enviado" e notifica o cliente.