Fiap - Services Architecture - Amazon Application Example

André Luiz Soares - 339880
Juan Carielo - 339589
Rafael Marinho - 339763
Vinicius Komninakis - 339762

Linguagem - Javascript (NodeJS)

A decisão de usar a plataforma Node JS, além da proficiência da maioria dos integrantes do grupo, se deu por algum motivadores:

1. Leve e rápida para I/O.
2. Muito adotada na comunidade open source.
3. Curva de aprendizado baixo.

Separação por Domínio

Para ter uma boa distribuição de expertises, distribuímos nossos serviços e sistemas por diferentes contextos e seus limitadores, formando 5 no total, onde são eles:

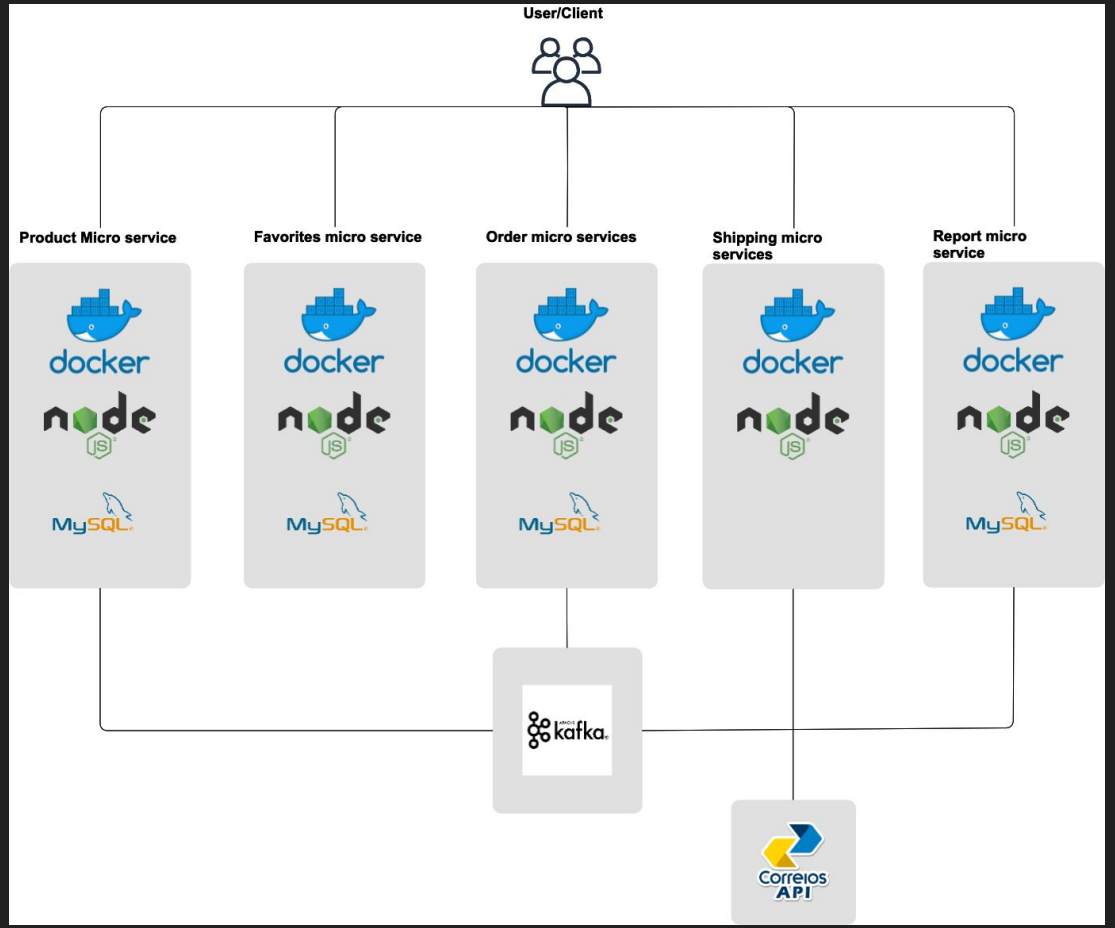
1. Serviço de Favoritos
2. Serviço de Produto
3. Serviço de Entrega/Frete
4. Serviço de Ordem de compra
5. Serviço de Reporte/Ticket

Onde usamos Kafka

Para conter uma comunicação assíncrona entre nossos serviços e ainda sim contemplar as demandas esperadas, utilizamos o Kafka como centro de todas as mensagens. Motivações:

1. Não ter a necessidade de chamadas síncronas (demoradas).
2. Distribuição de dados de forma independente.
3. Desacoplamento entre os serviços.

Arquitetura



Especificidades

Serviço de Frete/Entrega:

- Não utilizamos um banco de dados pois entendemos que poderíamos resolver apenas consultando o serviço dos Correios, sendo gerenciado por um Circuit Break e entregando um fallback padrão caso seja aberto ou meio aberto.

Documentações

Criamos em cada Micro Serviço um README contendo as especificações do swagger/OpenAPI e suas instruções.

E um README geral contendo instruções para rodar toda aplicação junta!