Entrega do trabalho de conclusão da disciplina  
Bruno Giannella de Melo (RM 334451)  
Walter Rodrigues (RM 334479)  
Lucas Di Biasi (RM 334481)

* O que fizemos?

Implementamos o case da Amazon, onde decidimos utilizar Spring Boot e Spring Cloud para implementar os Micoservices.

* Como fizemos?

Microservices que implementamos:

**Cliente:** Microsserviço para operações com clientes da plataforma **Pedido:** Microsserviço para operações com carrinho de compras e pedidos da plataforma **Frete:** Microsserviço para operação de calculo de frete **Produto:** Microsserviço para operações relacionadas aos produtos, por exemplo consulta de detalhes, consulta por gênero e palavra chave.  
**Suporte:** Microsserviço para abertura de chamados

Para apoio ao desenvolvimento dos Microservices usamos a stack da Neflix com Spring Cloud:

**Config Server:** Arquivos de configuração localizados no GitHub (também coloquei o mesmo projeto do Git na pasta de artefatos)  
**Eureka:** Service Discovery para os Microsserviços **Zipkin:** Monitoração e rastreabilidade **Sleuth:** Monitoração e rastreabilidade **Hystrix:** Circuit breaker para operações com possibilidade de falhas **Zuul:** API Gateway utilizando como fonte o Service Discovery Eureka

Para mensageria optamos por usar:

**RabbitMQ:** Criamos uma fila chamada PEDIDOS para atualizar as informações

de estoque dos produtos sempre que um pedido for realizado.

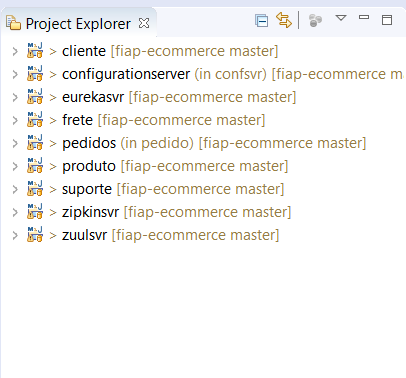
Para base de dados optamos por utilizar:

**MySQL:** Banco de dados relacionais para armazenar as informações

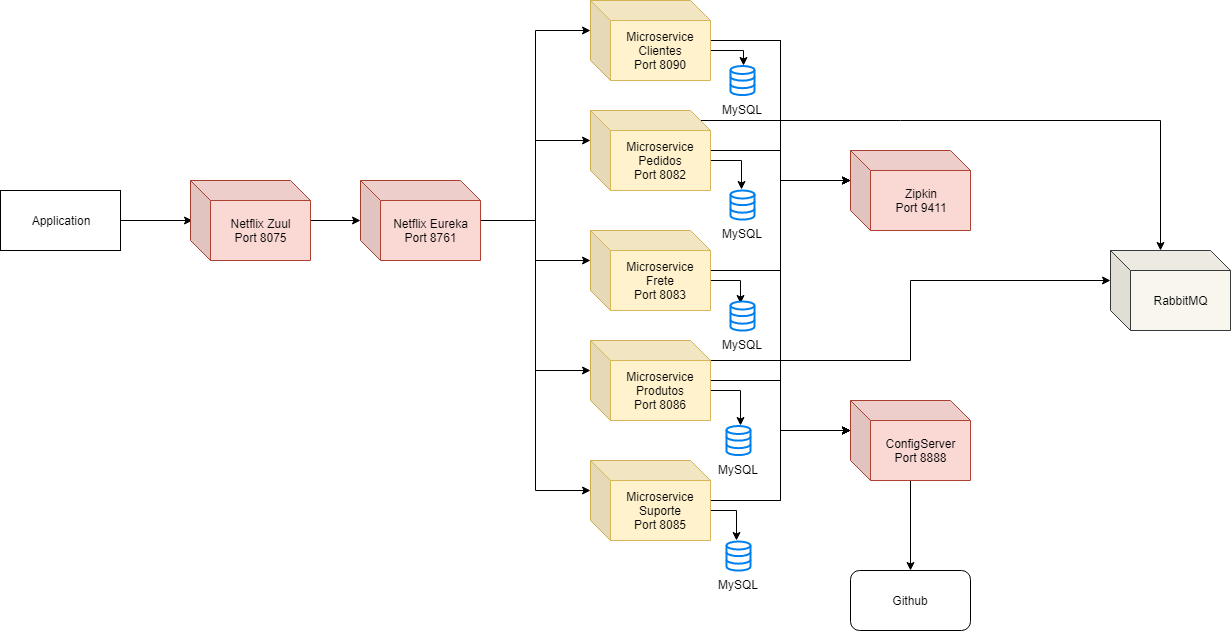
Linguagem e Build:

**Java  
 Maven**

* Estrutura dos projetos



* Desenho da solução



* Artefatos
* Para o front end implementamos todos os swaggers da aplicação utilizando a anotação OpenAPI 3.0, que está anexo na pasta /artefatos/swaggers
* Para o desenho da solução utilizamos Draw.io, que está localizado na pasta /artefatos/Desenho solução
* Para os testes utilizamos o Postman, onde tem uma Collection em anexo na pasta /artefatos/Ecommerce Fiap.postman\_collection.json
* Documentação no código utilizando a anotação Javadoc.
* Inicialização

1. Inicializar um MySQL local com usuário root e password root, rodando na porta padrão 3306 (nós optamos por inicializar via Docker)
2. Inicializar um RabbitMQ local rodando na porta 5672 com usuário guest e senha guest conforme o padrão (nós optamos por inicializar via Docker)
3. Inicializar o projeto configurationserver
4. Inicializar o projeto eurekasvr
5. Inicializar o projeto zipkinsvr
6. Inicializar o projeto zuulsvr
7. Inicializar todos os Microsserviços

* Testes e entendimento

Para realizar os testes e entendimento do projeto, pode ser utilizada a Collection do Postman, os Swaggers e a documentação do próprio código fonte.

Todo projeto além do zip enviado, também pode ser visualizado nos repositórios abaixo:

<https://github.com/brunogiannella/fiap-ecommerce-config>

<https://github.com/brunogiannella/fiap-ecommerce>

Qualquer dúvida também estamos a disposição.