



MBA DEVOPS ENGINEERING - INTEGRATION ARCHITECTURE

MICROSERVICES ARCHITECTURE / API / CONTAINERS

CAMILLA RAMALDES – RM 336003

HENRIQUE MARGARIDO – RM 335553

ROGÉRIO BULHÕES – RM 336011

VINICIUS VIEIRA – RM 335855

JUNHO DE 2020

## SUMÁRIO

1. Cenário .....	3
2. Arquitetura da Solução .....	3
3. Documentação do Projeto .....	4
4. Documentação dos Serviços.....	4
5. Infraestrutura.....	4
6. Anexos .....	<b>Erro! Indicador não definido.</b>

## 1. Cenário

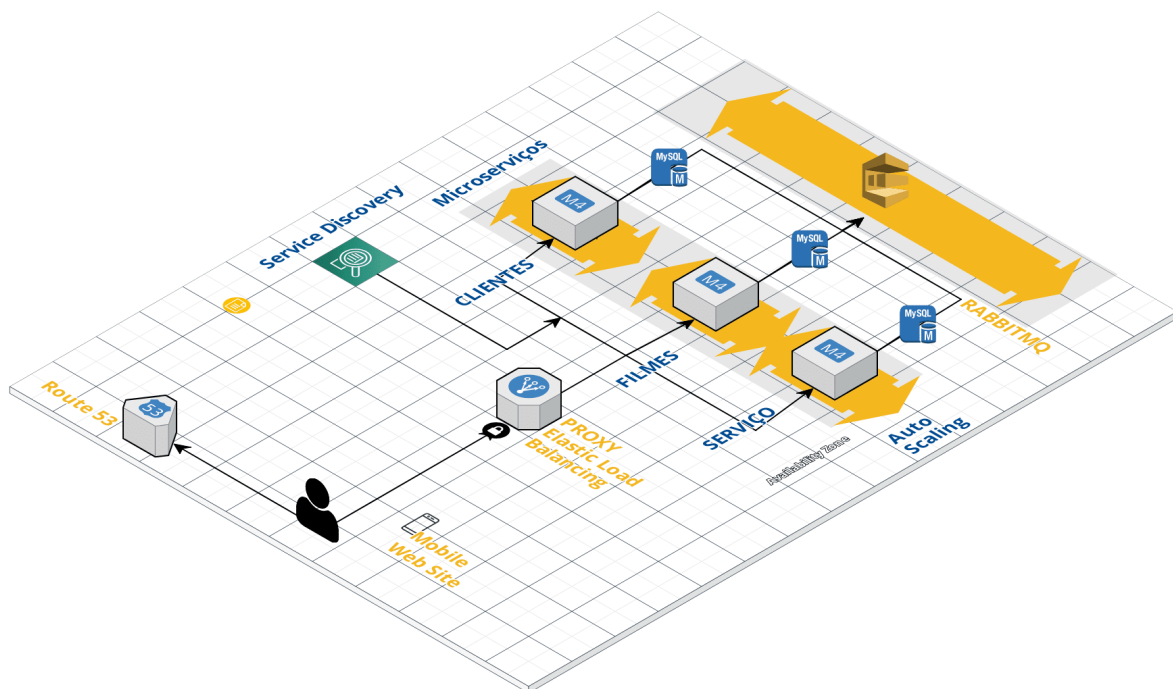
O cenário escolhido foi o da Netflix, vamos apresentar os pontos de entrega:

- Possibilidade de visualizar os filmes de um determinado gênero;
- Possibilidade de visualizar os detalhes de cada filme;
- Possibilidade de votar nos filmes que mais gostei;
- Possibilidade de marcar um filme ou serie para ser visto no futuro;
- Possibilidade de buscar um filme por palavra-chave;
- Possibilidade de exibir os filmes mais vistos por categorias;
- Possibilidade de abrir um chamado técnico de algum problema que está acontecendo;
- Possibilidade de visualizar os filmes e series que já foram assistidos;

## 2. Arquitetura da Solução

Desenvolvemos a solução na AWS, com EC2, instalação do java 8, maven, docker e docker compose.

- Instancia: t2.medium



### 3. Documentação do Projeto

Utilizamos java 8, maven, docker, docker Compose, service discovery, mensageria, resiliência e serviços específicos:

- Clientes: escrita e leitura na base de Clientes
- Filmes: escrita e leitura na base de Filmes
- Serviços: outras operações (likes, favoritos, marcações para o futuro, etc)

O sistema utiliza um gateway Zuul como ponto único de entrada das requisições.

### 4. Documentação dos Serviços

- Clientes
  - `/swagger-ui.html?urls.primaryName=clientes-api-docs#/clientes_service`
- Filmes
  - `/swagger-ui.html?urls.primaryName=filmes-api-docs`
- Serviços
  - `/swagger-ui.html?urls.primaryName=servicos-api-docs`

### 5. Infraestrutura

- Banco de dados: Mysql
- PHP: phpadmin
- MQ: rabbitmq
- Registry: registry
- Proxy: Zuul
- Imagens
  - `docker_zuul`
  - `docker_filmes`
  - `docker_clientes`
  - `docker_servicos`
  - `docker_discovery`
  - `mysql`
  - `phpmyadmin/phpMyAdmin`
  - `rabbitmq`
  - `openjdk`

## 6. Instalação

- Decompacte o zip

```
$ cd fiapmicro
$ mvn clean install
$ cd docker
$ docker-compose build
$ docker-compose up -d
```

## 7. Anexos

- Código Fonte