

# **AULA 13 - ATIVIDADE 01**

Disciplina de Backend - Professor Ramon Venson - SATC 2024

## **Fase 01: Discussão**

Conceitos chave do modelo REST, boas práticas.

## Fase 02: Modelagem

Em grupos de 4 estudantes, vocês deveram modelar uma API REST contendo dois (2) recursos do domínio de um serviço popular.

O objetivo é identificar dois **casos de uso** e pelo menos dois recursos necessários que devem ser solicitados por um aplicativo mobile.

Para cada um dos recursos, vocês devem especificar um conjunto de métodos necessários para realizar **operações CRUD**. Cada método deverá incluir: **URI, Método HTTP, Requisição esperada, Erros esperados e Status Codes..**

## Serviços

- Rede social de fotos (ex.: Instagram)
- Sistema de pagamento (ex.: Paypal)
- Rede social de microblogging (ex.: Twitter)
- Aplicativo de música (ex.: Spotify)
- Serviço de armazenamento (ex.: Dropbox)
- Serviço de meteorologia (ex.: weather.com)
- Canal de notícias (ex.: bbc.com)
- Loja de aplicativos (ex.: Google Play)

## Exemplo: Diário de Notas

### **Casos de uso:**

- Manipular dados de estudantes
- Manipular dados de avaliações

### **Recursos**

- Estudantes
- Avaliações

## Endpoints

### Avaliação

- Inserir avaliação ( POST )
- Ler avaliação ( GET )
- Modificar avaliação ( PUT ou PATCH )
- Excluir avaliação ( DELETE )

## Requisição: Inserir avaliação

**POST** /estudante/{estudante-id}/

```
{  
  "nome": "Avaliação Final",  
  "nota": 9.8,  
}
```

## Erros esperados: Inserir avaliação

- Não existe estudante com este ID;
- Avaliação já foi cadastrada nessa data (apenas uma por dia);
- Já existe uma avaliação com o mesmo nome;
- A nota deve ser entre 0 e 10 .

## Requisição: Modificar avaliação

**PATCH** /estudante/{estudante-id}/{avaliacao-id}

```
{  
  "nota": 10.0,  
}
```

## Erros esperados: Inserir avaliação

- Não existe estudante com este ID;
- Não existe avaliação com este ID;
- A nota deve ser entre 0 e 10 .



## Fase 03: Implementação

Agora com os modelos em mão, a equipe divide-se em duas duplas, onde cada uma será responsável pela implementação de cada caso de uso / recurso.

- O desenvolvimento deverá ser feito utilizando Spring Boot.
- Apenas um controller é necessário
- Não é necessário implementar services ou modelos. Cada rota pode retornar null ou um dado mockup (de exemplo e estático).

Utilize as anotações para cada um dos métodos: `@GetMapping`, `@PostMapping`, `@PutMapping`, `@DeleteMapping`.