

<LAB365>

**SPRING + DATA**

<LAB365>



# AGENDA

- Continuação Projeto Lista de Tarefas

# MINI PROJETO - LISTA DE TAREFAS

## MÃO NA MASSA!

- Construção de uma API REST que contenha as 4 operações básicas para o gerenciamento de tarefas simples.
- As tarefas contam com um identificador, descrição, data de entrega e data de início, responsável, status e prioridade da tarefa.
- Criar, listar, atualizar e deletar.

## MINI PROJETO - LISTA DE TAREFAS

- Como deve ficar o objeto?

```
{  
  "id": 1,  
  "description": "Realizar atividade...",  
  "startDate": "2023-10-25",  
  "endDate": "2023-10-26",  
  "status": "DOING", // PENDING, DOING, COMPLETED  
  "priority": "LOW", // LOW, MEDIUM, HIGH  
  "assignee": {  
    "id": 1,  
    "name": "Lucas"  
  }  
}
```

## MINI PROJETO - LISTA DE TAREFAS

- **Criando um objeto para representar o banco de dados em memória.**
- Trabalhando com records.
- Criando um controlador.
- Criando um serviço.
- Listando as tarefas.
- Adicionando tarefas.
- Atualizando tarefas.
- Deletando tarefas.

## MINI PROJETO - LISTA DE TAREFAS

- Criando um objeto para representar o banco de dados em memória.
- **Trabalhando com records.**
- Criando um controlador.
- Criando um serviço.
- Listando as tarefas.
- Adicionando tarefas.
- Atualizando tarefas.
- Deletando tarefas.

# O QUE SÃO RECORDS?

- Classes imutáveis, ou seja, valores definidos sem possibilidade de alteração.
- Elimina a verbosidade de construtores, *getters*, *setters*, *toString*, *equals*, *hashCode*, etc.
- Fácil legibilidade, poucas linhas de código.
- <https://docs.oracle.com/en/java/javase/16/language/records.html>

```
public record Task(String description) {  
  
}
```

## MINI PROJETO - LISTA DE TAREFAS

- Criando um objeto para representar o banco de dados em memória.
- Trabalhando com records.
- **Criando um controlador.**
- Criando um serviço.
- Listando as tarefas.
- Adicionando tarefas.
- Atualizando tarefas.
- Deletando tarefas.



## PROPRIEDADES DE UM CONTROLADOR

- Similar a estrutura dos Servlets que construímos para nossa API com Servlets.
- Configurado através de anotações.

- `@RestController`

Como o nome sugere, informa ao Spring que nossa classe age como um controlador REST que receberá requisições HTTP.

- `@RequestMapping`

Define qual o nome que identifica o controlador, para que o Spring saiba para onde redirecionar a requisição, similar ao *path* que usamos nos `@WebServlets`.

# PROPRIEDADES DE UM CONTROLADOR

Aplicações Spring trabalham com requisições através de anotações (*annotations*)!

**@PostMapping:** Mapeia uma requisição em um controller para receber requisições POST.

**@RequestBody:** Sinaliza que dado objeto recebido como parâmetro é o corpo da requisição.

Como o objeto chega ao controlador quando realizamos uma requisição?

Como ele é transformado em um objeto Java?

# PROPRIEDADES DE UM CONTROLADOR

- **Serialização:** Converter um objeto em um formato que possa ser transportado pela rede, como JSON, por exemplo.
- **Desserialização:** Converter um JSON, por exemplo, em um objeto.

O Spring já possui uma ferramenta de serialização/desserialização embutida, então não precisamos nos preocupar em importar bibliotecas externas manualmente e lidar com isto!

# PROPRIEDADES DE UM CONTROLADOR

Quando executarmos uma operação, queremos retornar o status HTTP apropriado, além de retornar objetos caso necessário. Para isso, podemos usar o **ResponseEntity**.

- Representa toda a resposta HTTP, incluindo status, corpo e cabeçalhos.

## MINI PROJETO - LISTA DE TAREFAS

- Criando um objeto para representar o banco de dados em memória.
- Trabalhando com records.
- Criando um controlador.
- **Criando um serviço.**
- Listando as tarefas.
- Adicionando tarefas.
- Atualizando tarefas.
- Deletando tarefas.
- Validando objetos.

## MINI PROJETO - LISTA DE TAREFAS

- Criando um objeto para representar o banco de dados em memória.
- Trabalhando com records.
- Criando um controlador.
- Criando um serviço.
- **Listando as tarefas.**
- Adicionando tarefas.
- Atualizando tarefas.
- Deletando tarefas.
- Validando objetos.

## MINI PROJETO - LISTA DE TAREFAS

- Criando um objeto para representar o banco de dados em memória.
- Trabalhando com records.
- Criando um controlador.
- Criando um serviço.
- Listando as tarefas.
- **Adicionando tarefas.**
- Atualizando tarefas.
- Deletando tarefas.
- Validando objetos.

## MINI PROJETO - LISTA DE TAREFAS

- Criando um objeto para representar o banco de dados em memória.
- Trabalhando com records.
- Criando um controlador.
- Criando um serviço.
- Listando as tarefas.
- Adicionando tarefas.
- **Atualizando tarefas.**
- Deletando tarefas.
- Validando objetos.



## MINI PROJETO - LISTA DE TAREFAS

- Criando um objeto para representar o banco de dados em memória.
- Trabalhando com records.
- Criando um controlador.
- Criando um serviço.
- Listando as tarefas.
- Adicionando tarefas.
- Atualizando tarefas.
- **Deletando tarefas.**
- Validando objetos.

## MINI PROJETO - LISTA DE TAREFAS

- Criando um objeto para representar o banco de dados em memória.
- Trabalhando com records.
- Criando um controlador.
- Criando um serviço.
- Listando as tarefas.
- Adicionando tarefas.
- Atualizando tarefas.
- Deletando tarefas.
- **Validando objetos.**

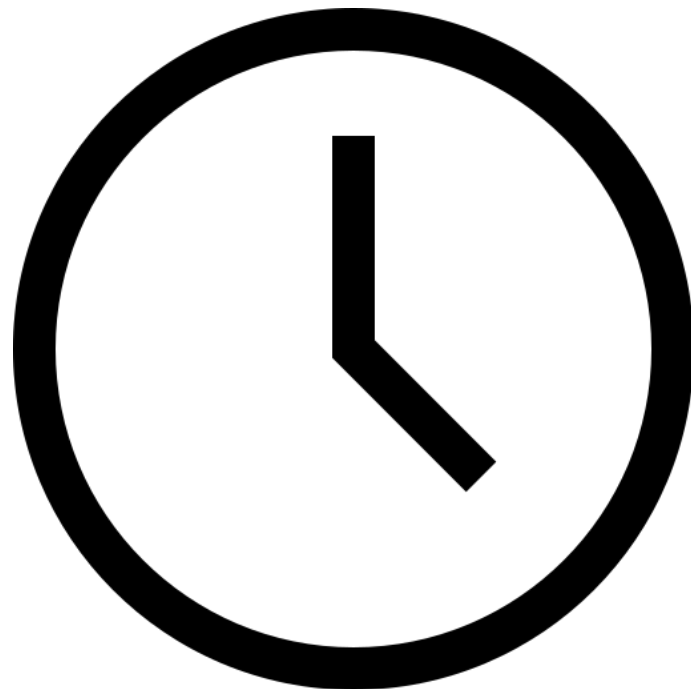
## INTERVALO!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

**Início:** 20:20

**Retorno:** 20:40



## AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

Clique [aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.



<LAB365>