

# Domain Driven Design

## *Eliane Marion*

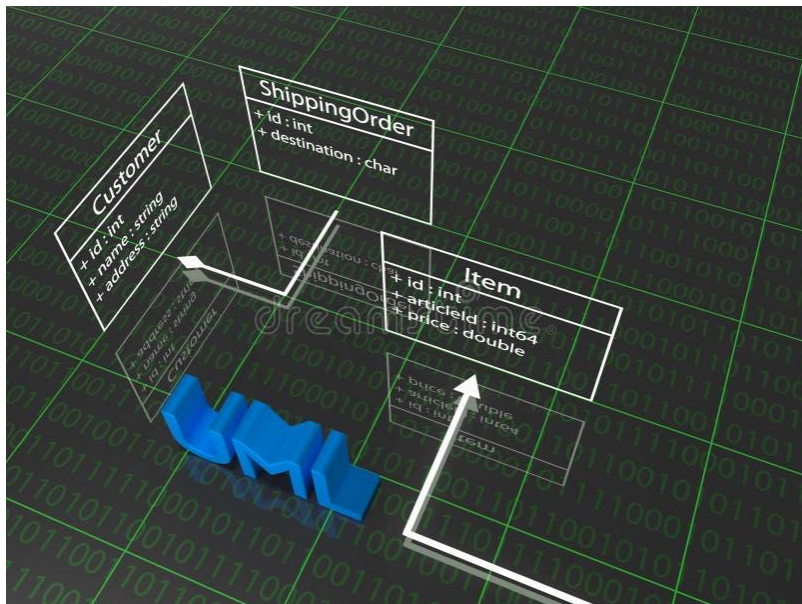
FIAP

2025

# AGENDA DE TRABALHO



INTEGRAÇÃO  
WEB



# 01

## WEB SERVICES

# MÉTODO GET – BUSCAR POR CÓDIGO

- Adicione um serviço para recuperar um produto pelo seu código.

Parte da URL para  
acessar a busca com um  
parâmetro (id)

Tipo do retorno  
(JSON)

```
@GET
@Path("/{id}")
@Produces(MediaType.APPLICATION_JSON)
public ProdutoRequestDto
  buscar(@PathParam("id") int codigo){
    return service.buscarPorId(codigo);
}
```

Injeta o parâmetro da URL  
no parâmetro do método

# Cadastrar PRODUTO – ProdutoResource

- Crie um método para *cadastrar* um Produto. O método recebe um objeto *ProdutoDTO* e retorna um *Response*.

Método POST

Recebe um JSON

Recebe o produto e as informações da URL

```
@POST
@Consumes(MediaType.APPLICATION_JSON)
public Response cadastrar(ProdutoRequestDto produto, @Context UriInfo
urilInfo){
    service.cadastrar(produto);
    UriBuilder builder = urilInfo.getAbsolutePathBuilder();
    builder.path(Integer.toString(produto.getCodigo()));
    return Response.created(builder.build().build());
}
```

Retorna um response com status 201 e URL para acessar o recurso cadastrado

# LISTAR PRODUTOS - ProdutoService

{ }

```
public List<ProdutoRequestDto> listar(){
    List<Produto> produtos = produtoDao.listar();
    return produtos.stream()
        .map(produto -> {
            ProdutoRequestDto produtoDto = new
            ProdutoRequestDto();
            produtoDto.setCodigo(produto.getCodigo());
            produtoDto.setNome(produto.getNome());
            produtoDto.setPreco(produto.getPreco());
            produtoDto.setQuantidade(produto.getQuantidade());
            return produtoDto;
        })
        .collect(Collectors.toList());
}
```

# TESTE – MÉTODO GET

- O método HTTP padrão para acessar um endereço através do browser é o GET;
- Execute o Servidor e faça uma chamada ao serviço através de sua URL no navegador:

[http://localhost:8080/webapp\\_war/api/produtos](http://localhost:8080/webapp_war/api/produtos)

Protocolo, Host, Porta e Contexto

Definido no web.xml

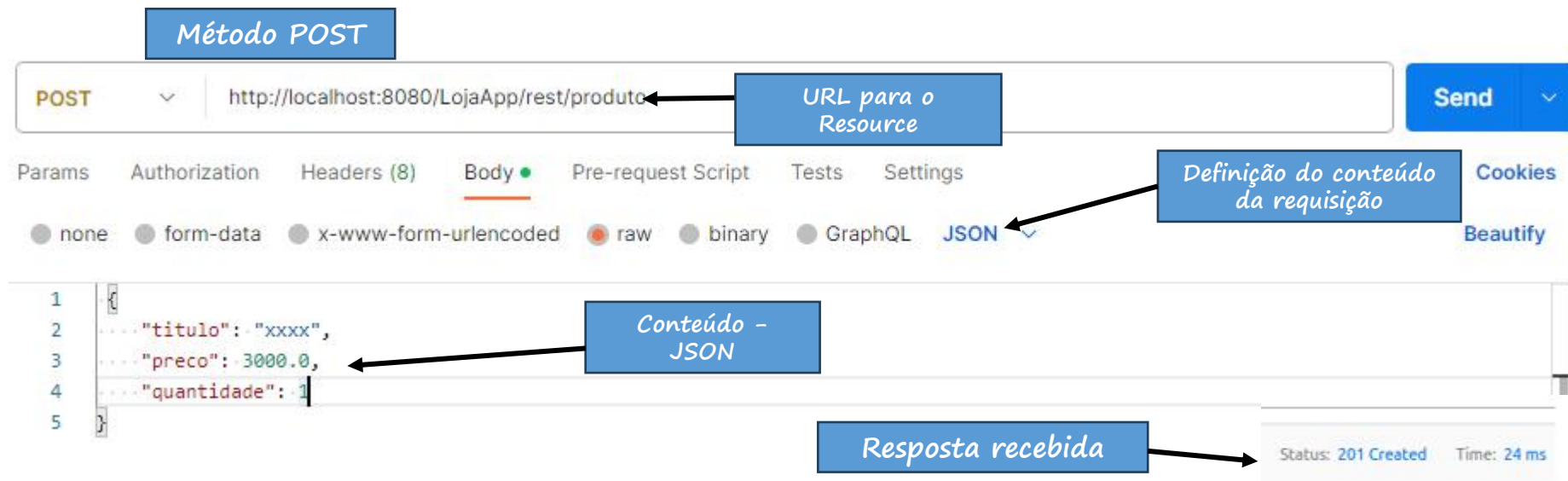
Definido no @Path



```
1 [
2   {
3     "codigo": 1,
4     "nome": "mouse",
5     "preco": 50,
6     "quantidade": 1
7   },
8   {
9     "codigo": 2,
10    "nome": "teclado",
11    "preco": 80,
12    "quantidade": 2
13  },
14 ]
```

# TESTE – MÉTODO POST

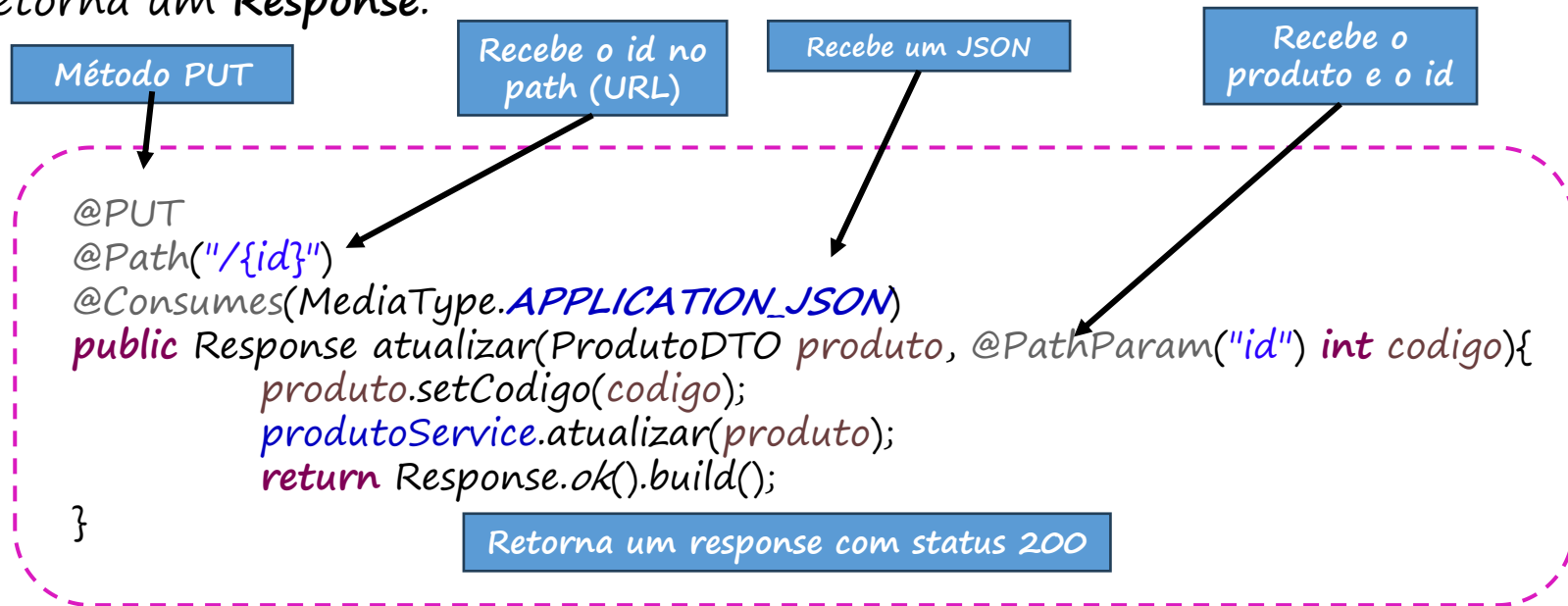
- Para enviar uma requisição POST, é preciso de uma ferramenta, como Postman.





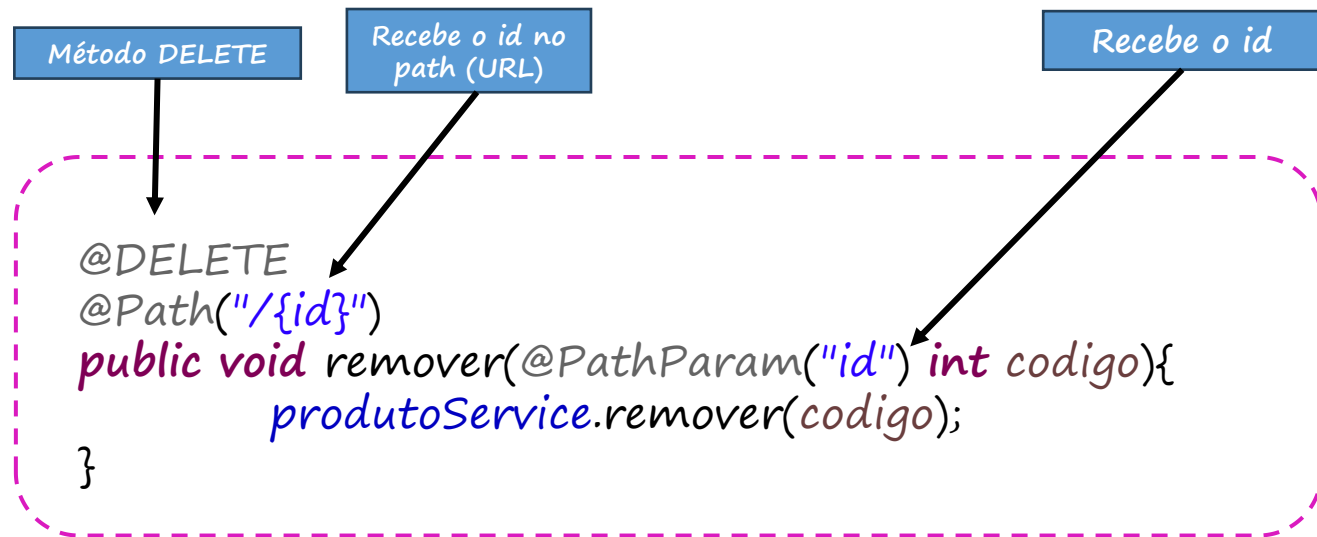
# ATUALIZAR UM PRODUTO – MÉTODO PUT

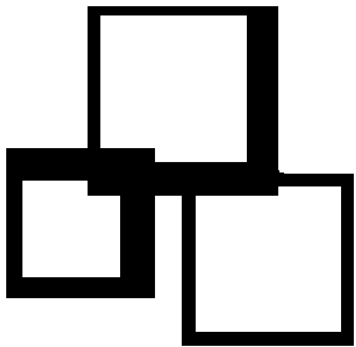
- O método para atualização recebe um objeto *ProdutoDTO* e o *id* do produto que será atualizado.
- Retorna um *Response*.



# DELETAR UM PRODUTO – MÉTODO DELETE

- O método de remoção recebe um código e não retorna nada (código HTTP 204 (no content));





*OBRIGADO*

To be continued...

