



## Faça como eu fiz: Criando o ProdutosController

Nossa API deve atender requisições feitas para a URL /produtos, para isso devemos criar um controlador (Controller), que será responsável por atender a rota e devolver uma resposta.

Dentro da pasta src vamos criar um novo arquivo chamado produtos.controller.ts. Este arquivo irá exportar uma classe chamada ProdutosController.

Para que o Nest entenda que a classe ProdutosController é um controlador, devemos anotá-la com o decorator @Controller, esse decorator deve ser importado do pacote @nestjs/common e também deve receber um parâmetro que é a rota pela qual deve responder.

```
import { Controller } from "@nestjs/common";

@Controller('produtos')
export class ProdutosController {
}
```

[COPIAR CÓDIGO](#)

Agora vamos criar o métodos que será responsáveis por listar, criar, alterar e apagar produtos:

Listando produtos:

```
@Get()
obterTodos(): string {
```

```
    return "Lista todos os produtos";  
  }
```

[COPIAR CÓDIGO](#)

Listando um produto específico:

```
@Get('/:id')  
obterUm(@Param() params): string {  
  console.log(params.id);  
  return `Retorna os dados do produto ${params.id}`;  
}
```

[COPIAR CÓDIGO](#)

Criando produto:

```
@Post()  
criar(@Body() produto): string {  
  console.log(produto);  
  return "Produto criado";  
}
```

[COPIAR CÓDIGO](#)

Alterando um produto:

```
@Put()  
alterar(@Body() produto): string {  
  console.log(produto);  
  return "Produto alterado";  
}
```

[COPIAR CÓDIGO](#)

## Apagando um produto:

```
@Delete('/:id')
apagar(@Param() params): string {
  console.log(params.id);
  return `Apaga o produto ${params.id}`;
}
```

[COPIAR CÓDIGO](#)