05

Faça como eu fiz: Modelando um produto

Utilizando como base a seguinte tabela, vamos criar uma classe TypeScript para representar Produtos na nossa aplicação.

Campo	Tipo	Descrição
Id	Número	Identificador único
Código	Texto	Código de referência
Nome	Texto	Nome do produto
Preço de venda	Número	Preço de venda

Dentro da pasta src vamos criar o arquivo produto.model.ts com o seguinte conteúdo:

```
export class Produto {

}
```

COPIAR CÓDIGO

Dentro das chaves, vamos propriedades da classe:

```
id: number;
codigo: string;
nome: string;
preco: number;
```

COPIAR CÓDIGO

Logo abaixo das propriedades, vamos criar um método construtor:

```
constructor(codigo, nome, preco) {
    this.codigo = codigo;
    this.nome = nome;
    this.preco = preco;
}
```

COPIAR CÓDIGO

A versão final da classe Produto ficou assim:

```
export class Produto {
    id: number;
    codigo: string;
    nome: string;
    preco: number;
```

```
        constructor(codigo: string, nome: s
            this.codigo = codigo;
            this.nome = nome;
            this.preco = preco;
        }
    }
```

COPIAR CÓDIGO

Agora precisamos refatorar o ProdutosController para que trabalhe com Produto ao invés de strings:

Primeiramente, no topo do arquivo vamos importar a classe Produto com a seguinte linha de código: `import { Produto } from './produto.model';`

Agora, dentro da classe ProdutosController vamos criar uma propriedade `produtos` que será um array de produtos com alguns dados iniciais:

```
  produtos: Produto[] = [
      new Produto("LIV01", "Livro TDD e B
      new Produto("LIV02", "Livro Inician
```

```
        new Produto("LIV03", "Inteligência
];
```

COPIAR CÓDIGO

Agora vamos alterar o método `obterTodos` para que retorne o array de produtos:

```
@Get()
obterTodos(): Produto[] {
    return this.produtos;
}
```

COPIAR CÓDIGO

Chegou a vez do método `obterUm`, vamos alterá-lo também:

```
@Get(':id')
obterUm(@Param() parametros): Produto {
    return this.produtos[0];
}
```

COPIAR CÓDIGO

O método `criar` também precisa ser refatorado, ele vai ficar assim:

```
@Post()
criar(@Body() produto: Produto) {
    produto.id = 100;
    this.produtos.push(produto);
}
```

COPIAR CÓDIGO

O método `alterar` vai ficar da seguinte forma:

```
@Put()
alterar(@Body() produto: Produto): Prod
    return produto;
}
```

COPIAR CÓDIGO

Por fim o método apagar, por enquanto, ele simplesmente removerá o último produto da lista.

```
@Delete(':id')
apagar(@Param() parametros) {
```

```
        this.produtos.pop();
    }
```

COPIAR CÓDIGO