

**Matricula**  
2023 0407 7282

Web, baseado nas tecnologias de Servlets, JPA e JEE

The screenshot shows an IDE with the following code:

```

1  package com.example.springboot;
2
3  import org.springframework.web.bind.annotation.*;
4
5  @RestController
6  @RequestMapping("/api")
7  public class ProductController {
8
9      // GET /api/products
10     @GetMapping("/products")
11     public List<Product> getAllProducts() {
12         // Simulating database lookup
13         List<Product> products = new ArrayList<>();
14
15         // Adding sample data
16         products.add(new Product(1, "Laptop", 1200));
17         products.add(new Product(2, "Smartphone", 800));
18         products.add(new Product(3, "Tablet", 300));
19
20         return products;
21     }
22 }

```

The Product class is defined in the background as follows:

```

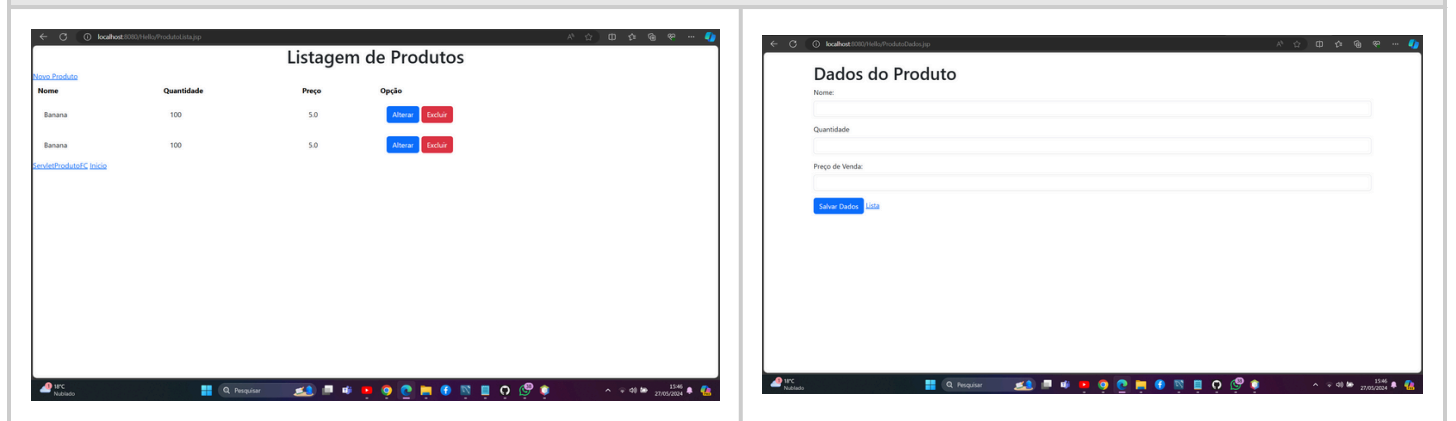
1  package com.example.springboot;
2
3  import java.util.Objects;
4
5  public class Product {
6
7      private int id;
8      private String name;
9      private double price;
10
11      // Constructor
12      public Product(int id, String name, double price) {
13          this.id = id;
14          this.name = name;
15          this.price = price;
16      }
17
18      // Getters and Setters
19      public int getId() { return id; }
20      public String getName() { return name; }
21      public double getPrice() { return price; }
22      public void setId(int id) { this.id = id; }
23      public void setName(String name) { this.name = name; }
24      public void setPrice(double price) { this.price = price; }
25
26      // Equals and HashCode
27      @Override
28      public boolean equals(Object o) {
29          if (this == o) return true;
30          if (o == null || getClass() != o.getClass()) return false;
31          Product product = (Product) o;
32          return id == product.id && name.equals(product.name) && price == product.price;
33      }
34
35      @Override
36      public int hashCode() {
37          return Objects.hash(id, name, price);
38      }
39
40      // ToString
41      @Override
42      public String toString() {
43          return "Product{" +
44              "id=" + id +
45              ", name='" + name + '\'' +
46              ", price=" + price +
47              '}';
48      }
49 }

```

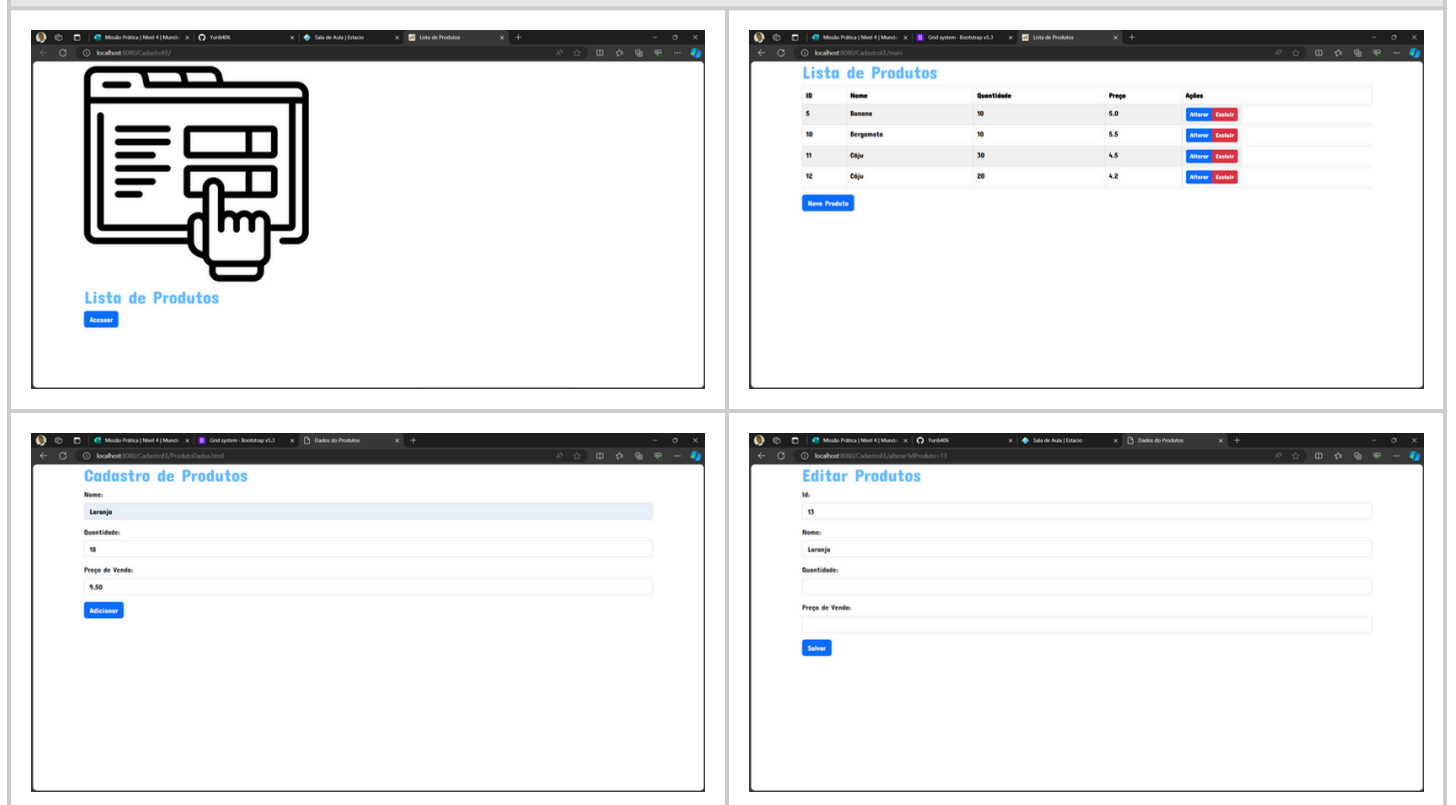
### Resultados da execução:



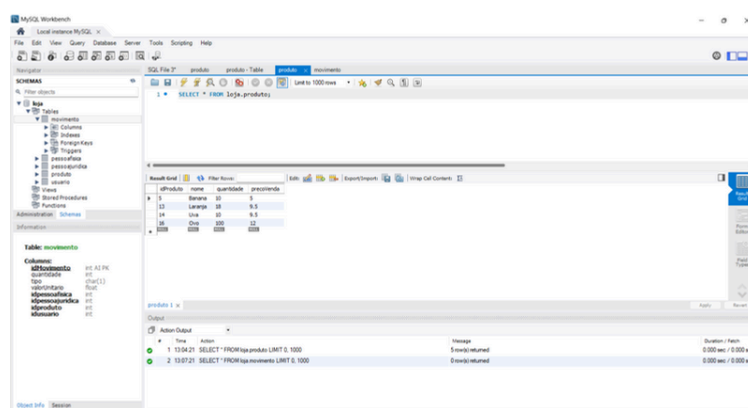
Resultados da execução :



Resultados da execução :



## Banco de dados :



## Análise e Conclusão:

Durante a elaboração do projeto, deparei-me com desafios significativos ao lidar com o servidor Glassfish, que atualmente é menos comum devido à existência de tecnologias mais contemporâneas que oferecem os mesmos resultados e são mais prevalentes. Optei por empregar a IDE Eclipse, uma vez que o tutor permitiu a escolha da tecnologia, desde que o resultado final fosse o mesmo.

Em certas etapas, decidi utilizar o NetBeans devido à sua praticidade. Apesar de tentativas repetidas de desenvolver o projeto com as tecnologias sugeridas, enfrentei diversos problemas com o Glassfish, incluindo questões de versão e comandos obsoletos. Por conseguinte, optei por adotar o servidor Tomcat, que se revelou mais intuitivo e com suporte mais acessível, uma característica que não encontrei no Glassfish. Além disso, utilizei o MySQL como banco de dados.

## Organização de um Projeto Corporativo no NetBeans:

No NetBeans, a organização de um projeto corporativo geralmente segue a estrutura de um projeto Enterprise “EAR”. Esse tipo de projeto inclui módulos EJB “Enterprise JavaBeans” e módulos Web “geralmente usando Servlets e JSP”.

## A estrutura típica de um projeto corporativo no NetBeans inclui:

**Projeto EAR:** Contém os módulos EJB e Web.

**Módulos EJB:** Implementam a lógica de negócios e interagem com o banco de dados usando

JPA Java Persistence API”.

**Módulos Web:** Contêm Servlets, JSP “JavaServer Pages” e outros componentes da interface do usuário.

O NetBeans oferece assistentes e ferramentas para criar, configurar e gerenciar esses módulos, facilitando o desenvolvimento de aplicativos corporativos.

### **Papel das Tecnologias JPA e EJB:**

**JPA “Java Persistence API”:** É uma especificação que permite persistir objetos Java em bancos de dados relacionais. Ela simplifica o mapeamento objeto-relacional e oferece funcionalidades essenciais para aplicações corporativas, como transações e consultas.

**EJB “Enterprise JavaBeans”:** São componentes do lado do servidor que encapsulam a lógica da aplicação comercial. Eles podem ser sem estado ou com estado. Os EJBs interagem com o banco de dados, implementam regras de negócio e oferecem serviços no nível de sistema para aplicações corporativas.

### **Melhoria de Produtividade com NetBeans e Tecnologias JPA/EJB:**

O NetBeans oferece assistentes para criar e configurar EJBs e módulos Web.

Com o suporte a JPA, você pode gerar automaticamente entidades a partir do banco de dados, facilitando o desenvolvimento e a manutenção.

O NetBeans também simplifica a criação de Servlets e JSP, permitindo que você se concentre na lógica de negócios em vez de detalhes de implementação.

A integração com servidores de aplicação como o Apache Tomcat permite testar e implantar seus aplicativos diretamente na IDE.

Em resumo, o NetBeans agiliza o desenvolvimento, oferecendo ferramentas e assistentes para lidar com as tecnologias JPA e EJB.

### **Servlets e Suporte no NetBeans:**

Servlets são componentes Java que processam requisições HTTP e geram respostas dinâmicas para o navegador.

### **O NetBeans oferece suporte à construção de Servlets:**

Você pode criar Servlets diretamente na IDE.

O mapeamento dos Servlets é feito no arquivo web.xml, que é gerado automaticamente.

O NetBeans facilita a criação de páginas JSP “JavaServer Pages” que interagem com os Servlets.

A estrutura de diretórios do projeto Web é gerenciada pelo NetBeans, incluindo a pasta WEB-INF com o arquivo web.xml.

### **Comunicação entre Servlets e Session Beans:**

Os Session Beans são componentes EJB que podem ser sem estado ou com estado. A comunicação entre Servlets e Session Beans ocorre por meio de chamadas remotas ou locais.

O Servlet pode fazer uma busca para obter uma referência ao Session Bean e, em seguida, invocar seus métodos.

O Session Bean processa a lógica de negócios e retorna os resultados ao Servlet.

O NetBeans facilita a criação e configuração desses componentes, permitindo que você se concentre na funcionalidade do aplicativo.

### **Padrão Front Controller na Arquitetura MVC:**

O Front Controller é um padrão de projeto arquitetural que centraliza o processamento de requisições em um único ponto de entrada.

Na arquitetura MVC “Model-View-Controller”, o Front Controller atua como o controlador central que recebe todas as requisições do cliente geralmente via URL e decide como roteá-las para os componentes apropriados modelos e visualizações.

### **O Front Controller é responsável por:**

Autenticar e autorizar usuários, roteamento de URLs para os controladores apropriados, execução de lógica de negócios comuns a todas as páginas e gerenciamento de sessões e estado da aplicação.

### **Implementação em um aplicativo Web Java:**

No contexto Java EE, você pode usar um Servlet como o Front Controller. O Servlet recebe todas as requisições e decide como encaminhá-las para os controladores específicos ou outros componentes, como JSPs ou EJBs.

O Servlet pode ser mapeado para uma URL específica no arquivo de configuração, como web.xml ou anotações.

Quando uma requisição chega, o Servlet decide qual controlador ou ação deve processá-la com base na URL.

O Front Controller ajuda a manter a consistência e a modularidade em todo o aplicativo, facilitando a manutenção e a escalabilidade.

### **Diferenças e Semelhanças entre Servlets e JSPs:**

Servlets e JSPs são componentes da plataforma Java EE usados para criar aplicativos Web.

Ambos são executados no servidor e geram respostas HTML para o cliente e podem ser usados para processar solicitações HTTP.

A diferença é que Servlets são classes Java que estendem a classe `HttpServlet`, são mais flexíveis e podem ser usados para qualquer tipo de processamento de dados, requerem mais código e são mais difíceis de escrever. Enquanto JSPs são arquivos HTML com trechos de código Java incorporados, são mais fáceis de usar para desenvolvedores familiarizados com HTML, são mais adequados para a apresentação e interação com o usuário, permitem a separação de lógica de negócios e lógica de apresentação usando JavaBeans.

### **Redirecionamento Simples vs. Método Forward:**

Redirecionamento Simples “`sendRedirect`” o cliente “navegador” recebe uma resposta HTTP com um cabeçalho informando que ele deve requisitar outra página.

O navegador faz uma nova requisição para a URL especificada.

O redirecionamento ocorre no lado do cliente.

Método Forward “`RequestDispatcher.forward`” a requisição do usuário é encaminhada para ser atendida por outro recurso outro servlet, JSP etc. O processamento ocorre no lado do servidor, o usuário não percebe o redirecionamento, pois a URL permanece a mesma.

O forward é mais adequado para manter a experiência de navegação do usuário e a consistência da URL.

### **O que é o Bootstrap?**

O Bootstrap é um framework de código aberto, ele oferece uma vasta biblioteca de componentes pré-construídos, como botões, formulários, barras de navegação e muito mais. Sua principal característica é a responsividade, permitindo que os elementos da página se adaptem automaticamente a diferentes tamanhos de tela, desde notebooks até smartphones e monitores maiores.

### **Independência Estrutural do HTML:**

O Bootstrap não apenas estiliza elementos, mas também fornece funcionalidades além do visual, ele permite implementar menus de navegação, controles de paginação, janelas modais e

outros recursos. Sua estrutura simples de arquivos requer conhecimentos básicos de HTML, CSS e JavaScript para utilizá-lo.

O Bootstrap pode ser usado em diferentes linguagens de programação e frameworks, adaptando-se bem a diversos tipos de aplicações.

### **Relação entre Bootstrap e Responsividade:**

O Bootstrap é construído com a responsividade em mente.

Seus sistemas de grid flexíveis e classes de mídia permitem que os sites se ajustem automaticamente a diferentes dispositivos, ao usar o Bootstrap, você garante que sua aplicação seja amigável para dispositivos móveis e ofereça uma experiência consistente em telas variadas.

**Link GitHub:** [Yuri6406/3-Semestre-ProjetoNivel4 \(github.com\)](https://github.com/Yuri6406/3-Semestre-ProjetoNivel4).