



Curso:



Desenvolvimento Full Stack

Campus:

UNIVERSIDADE ESTÁCIO DE SÁ

Disciplina:

Vamos Integrar Sistemas

Turma:

2024.3

Aluno:

DARCI RODRIGUES FALCÃO NETO

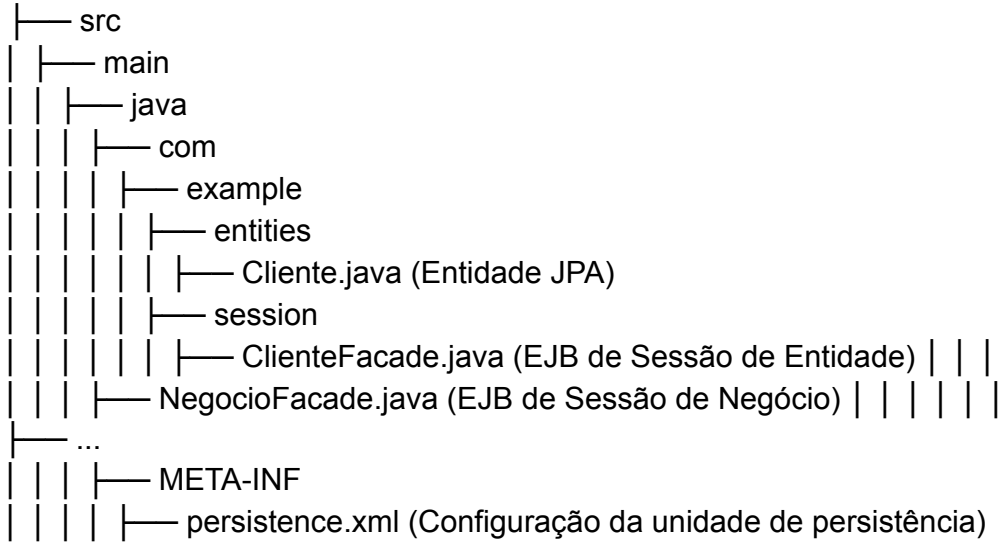


OBJETIVO NA PRÁTICA

- 1. Implementar persistência com base em JPA.**
- 2. Implementar regras de negócio na plataforma JEE, através de EJBs.**
- 3. Implementar sistema cadastral Web com base em Servlets e JSPs.**
- 4. Utilizar a biblioteca Bootstrap para melhoria do design.**
- 5. No final do exercício, o aluno terá criado todos os elementos necessários para exibição e entrada de dados na plataforma**

Java Web, tornando-se capacitado para lidar com contextos reais de aplicação.

CadastroEE-ejb



```
package com.example.model;
```

```
import java.io.Serializable;
```



Estácio

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Produto implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nome;
    private int quantidade;
    private float precoVenda;

    // Getters e setters
}
```

```
package com.example.ejb;
```

```
import com.example.model.Produto;  
import java.util.List;  
import javax.ejb.Local;
```

```
@Local
```

```
public interface ProdutoFacadeLocal {  
    void create(Produto produto);  
    void edit(Produto produto);  
    void remove(Produto produto);  
    Produto find(Object id);  
    List<Produto> findAll();  
}
```

```
package com.example.ejb;
```

```
import com.example.model.Produto;  
import java.util.List;  
import javax.ejb.Stateless;  
import javax.persistence.EntityManager;  
import javax.persistence.PersistenceContext;
```

```
@Stateless
```

```
public class ProdutoFacade extends AbstractFacade<Produto> implements  
ProdutoFacadeLocal {
```

```
    @PersistenceContext(unitName =  
        "NomeDaSuaUnidadeDePersistencia") private EntityManager em;
```

```
@Override
```



Estácio

```
protected EntityManager getEntityManager() {  
    return em;  
}  
  
public ProdutoFacade() {  
    super(Produto.class);  
}  
}
```

```
package com.example.servlets;
```

```
import com.example.ejb.ProdutoFacadeLocal;  
import com.example.model.Produto;  
import java.io.IOException;  
import java.util.List;
```

```

import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name = "ProdutoServlet", urlPatterns =
{"/produtos"}) public class ProdutoServlet extends HttpServlet {
    @EJB
    private ProdutoFacadeLocal produtoFacade;

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
        List<Produto> produtos = produtoFacade.findAll();
        request.setAttribute("produtos", produtos);
        request.getRequestDispatcher("/produto-lista.jsp").forward(request,
            response); }
}

```

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Lista de Produtos</title>
    <!-- Inclua os links para o Bootstrap -->
</head>
<body>
    <div class="container">

```



Estácio

```
<h1>Lista de Produtos</h1>
<table class="table table-striped">
  <tr>
    <th>ID</th>
    <th>Nome</th>
    <th>Quantidade</th>
    <th>Preço de Venda</th>
  </tr>
  <c:forEach var="produto" items="{produtos}">
    <tr>
      <td>${produto.id}</td>
      <td>${produto.nome}</td>
      <td>${produto.quantidade}</td>
      <td>${produto.precoVenda}</td>
    </tr>
  </c:forEach>
```



```

        </table>
    </div>
</body>
</html>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Cadastro de Produto</title>
    <!-- Inclua os links para o Bootstrap -->
</head>
<body>
    <div class="container">
        <h1>Cadastro de Produto</h1>
        <form action="produtos" method="POST">
            <input type="hidden" name="acao" value="{acao}">
            <input type="hidden" name="id" value="{produto.id}">
            <div class="mb-3">
                <label class="form-label">Nome</label>
                <input type="text" class="form-control" name="nome" value="{produto.nome}">
            </div>
            <div class="mb-3">
                <label class="form-label">Quantidade</label>
                <input type="number" class="form-control" name="quantidade"
value="{produto.quantidade}">
            </div>
            <div class="mb-3">
                <label class="form-label">Preço de Venda</label>
                <input type="number" class="form-control" name="precoVenda"
value="{produto.precoVenda}">
            </div>
            <button

```



Estácio

```
        type="submit" class="btn btn-primary">Salvar</button>
    </form>
</div>
</body>
</html>
```

```
create-jdbc-resource --connectionpoolid NomeDoSeuPool jdbc/NomeDoRecurso
```

Análise e Conclusão:

Como o framework Bootstrap é utilizado?

O Bootstrap é um popular framework de código aberto para desenvolvimento web que facilita a criação de sites e aplicativos responsivos e visualmente atraentes. Ele é baseado em HTML, CSS e JavaScript e fornece um conjunto de estilos e componentes predefinidos que podem ser facilmente personalizados e adaptados para atender às necessidades do seu projeto. Aqui estão os passos básicos para utilizar o Bootstrap:

1. ****Inclusão dos arquivos do Bootstrap:****

- Baixe os arquivos do Bootstrap do site oficial (<https://getbootstrap.com/>). - Inclua os arquivos CSS (bootstrap.css ou bootstrap.min.css) e JavaScript (bootstrap.js ou bootstrap.min.js) no seu projeto. Você também pode usar o CDN (Content Delivery Network) do Bootstrap para incluir esses arquivos diretamente no seu HTML.

Exemplo de inclusão via CDN:

```
``html
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.5.0/dist/css/bootstrap.min.css"
rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.5.0/dist/js/bootstrap.min.js"></script>
``
```

2. ****Estrutura HTML básica:****



- Crie a estrutura HTML básica para o seu projeto, incluindo os elementos necessários, como divs, botões, formulários, etc.

3. ****Uso de classes Bootstrap:****

- Utilize as classes Bootstrap em seus elementos HTML para aplicar estilos e funcionalidades predefinidos. Por exemplo, para criar um botão estilizado, você pode usar a classe `btn`:

```
```html
<button class="btn btn-primary">Botão</button>
```
```

4. ****Personalização:****

- Personalize os estilos e componentes do Bootstrap de acordo com as necessidades do seu projeto. Você pode sobrescrever as classes CSS do Bootstrap ou criar suas

próprias classes para estilizar elementos de forma única.

5. ****Layout responsivo:****

- O Bootstrap facilita a criação de layouts responsivos. Use classes como `container`, `row`, `col` para criar grids flexíveis que se ajustam automaticamente ao tamanho da tela.

```
```html
<div class="container">
 <div class="row">
 <div class="col-md-6">Conteúdo da coluna 1</div>
 <div class="col-md-6">Conteúdo da coluna 2</div>
 </div>
</div>
```
```

6. ****Componentes e plugins:****

- O Bootstrap oferece uma variedade de componentes e plugins prontos para uso, como modais, carrosséis, barras de navegação, etc. Você pode simplesmente incluir esses componentes em seu projeto e configurá-los conforme necessário.

7. ****Teste e otimização:****

- Certifique-se de testar o seu site ou aplicativo em diferentes dispositivos e tamanhos de tela para garantir que a experiência do usuário seja consistente e responsiva.

O Bootstrap é uma ferramenta poderosa para o desenvolvimento web rápido e eficiente, pois economiza tempo na criação de estilos e componentes comuns, permitindo que os desenvolvedores se concentrem mais na lógica de negócios e nas funcionalidades exclusivas do projeto. Além disso, a documentação do Bootstrap é extensa e oferece exemplos detalhados para ajudar os desenvolvedores a tirar o máximo proveito do framework.



Por que o Bootstrap garante a independência estrutural do HTML? ○

Bootstrap não garante exatamente a independência estrutural do HTML, mas ele ajuda a separar a camada de apresentação (estilo) da camada de estrutura (HTML) em projetos web. Esse é um conceito fundamental conhecido como separação de preocupações (separation of concerns) e é uma prática recomendada no desenvolvimento web.

Aqui estão algumas maneiras pelas quais o Bootstrap ajuda a promover essa separação de preocupações:

1. **Classes CSS:**

O Bootstrap fornece classes CSS predefinidas que você pode aplicar aos elementos HTML. Isso significa que você pode adicionar estilos aos elementos HTML sem precisar incluir estilos diretamente no HTML usando atributos `style`. Em vez disso, você usa classes Bootstrap para aplicar estilos, o que mantém a estrutura HTML limpa e livre de estilos embutidos.

Exemplo:

```
```html
<button class="btn btn-primary">Clique em mim</button>
```
```

2. **Classes de componentes:**

O Bootstrap oferece uma variedade de classes para criar componentes de interface do usuário, como botões, formulários, barras de navegação e muito mais. Isso permite que você adicione esses componentes facilmente ao seu HTML sem precisar escrever código CSS personalizado.

Exemplo:

```
```html
<nav class="navbar navbar-expand-lg navbar-light bg-light">
 <!-- Conteúdo da barra de navegação -->
</nav>
```
```

3. **Personalização:**

Embora o Bootstrap forneça estilos predefinidos, ele também permite que você personalize esses estilos. Você pode criar um arquivo CSS personalizado para substituir ou estender os estilos do Bootstrap sem modificar o HTML subjacente. Isso mantém a independência estrutural do HTML, pois os estilos são definidos separadamente.

4. **Responsividade:**



O Bootstrap ajuda a criar layouts responsivos sem modificar a estrutura HTML. Usando classes como ``container``, ``row`` e ``col``, você pode criar grades flexíveis para diferentes tamanhos de tela, tornando seu site ou aplicativo mais adaptável.

Portanto, o Bootstrap promove boas práticas de desenvolvimento web, permitindo que você mantenha a separação entre o HTML e os estilos, facilitando a manutenção, personalização e colaboração em projetos web. Isso não apenas torna o código mais organizado, mas também contribui para a reutilização de componentes e a consistência visual em todo o projeto.

Qual a relação entre o Bootstrap e a responsividade da página? O

Bootstrap tem uma relação fundamental com a responsividade de uma página web. Ele é conhecido por ser um dos frameworks mais populares para desenvolvimento de sites e aplicativos responsivos. Aqui estão algumas das maneiras pelas quais o

Bootstrap facilita a criação de layouts responsivos:

1. **Grid System:** O Bootstrap utiliza um sistema de grade (grid system) flexível que permite criar layouts responsivos com facilidade. O grid é baseado em classes como ``container``, ``row``, e ``col``, que permitem organizar o conteúdo em colunas e linhas. Você pode especificar o número de colunas que cada elemento ocupará em diferentes tamanhos de tela (por exemplo, dispositivos móveis, tablets e desktops). Isso garante que seu conteúdo se ajuste automaticamente ao tamanho da tela.
2. **Classe de Colunas Responsivas:** As classes de colunas responsivas, como ``col-xs-``, ``col-sm-``, ``col-md-`` e ``col-lg-``, permitem definir o comportamento das colunas com base no tamanho da tela. Por exemplo, você pode ter uma coluna que ocupe 12 colunas em telas pequenas (`col-xs-12`) e 6 colunas em telas médias (`col-md-6`), o que é útil para criar layouts flexíveis.
3. **Controle de Visibilidade:** O Bootstrap fornece classes para controlar a visibilidade de elementos com base no tamanho da tela. Você pode usar classes como ``d-none`` e ``d-md-block`` para mostrar ou ocultar elementos em diferentes dispositivos, ajudando a criar uma experiência de usuário mais adaptativa.
4. **Componentes Responsivos:** Muitos dos componentes do Bootstrap são projetados para serem responsivos por padrão. Isso inclui elementos como barras de navegação, carrosséis, formulários e modais, que são estilizados e ajustados automaticamente para funcionar bem em dispositivos de diferentes tamanhos.
5. **Classes de Utilitários Responsivos:** O Bootstrap também oferece classes de utilitários responsivos, como ``d-flex``, ``d-inline``, ``text-center``, entre outras, que permitem ajustar facilmente o comportamento dos elementos em diferentes tamanhos de tela.



Portanto, o Bootstrap desempenha um papel significativo na criação de sites e aplicativos responsivos, fornecendo uma estrutura e um conjunto de ferramentas que simplificam a adaptação do conteúdo para atender às diversas resoluções de tela e dispositivos utilizados pelos usuários finais. Isso é essencial para proporcionar uma experiência de usuário consistente e agradável, independentemente do dispositivo que está sendo usado para acessar o site ou aplicativo.