



FACULDADE ESTÁCIO DE SÁ
POLO PRQ DAS NAÇÕES CURSO:
DESENVOLVIMENTO FULL STACK 3º SEMESTRE
MATRÍCULA 202302798721
JÚLIO CÉSAR SANTOS RAMOS

VAMOS INTEGRAR SISTEMAS

RELATÓRIO DA MISSÃO PRÁTICA

Santo André 2024

Objetivo:

Desenvolver um sistema cadastral web robusto e amigável, utilizando tecnologias Java como JPA, JEE e Bootstrap.

Detalhes:

- **Persistência:**
 - Armazenamento seguro e confiável de dados com JPA.
 - Regras de negócio organizadas e eficientes com EJBs (JEE).
- **Interface:**
 - Criação de um sistema web utilizando Servlets e JSPs.
 - Interface amigável e intuitiva com Bootstrap.
 - Experiência acessível e agradável para o usuário final.

Benefícios:

- Sistema completo para cadastro de informações.
- Segurança e confiabilidade dos dados.
- Interface amigável e intuitiva.
- Experiência acessível para todos os usuários.

Tecnologias:

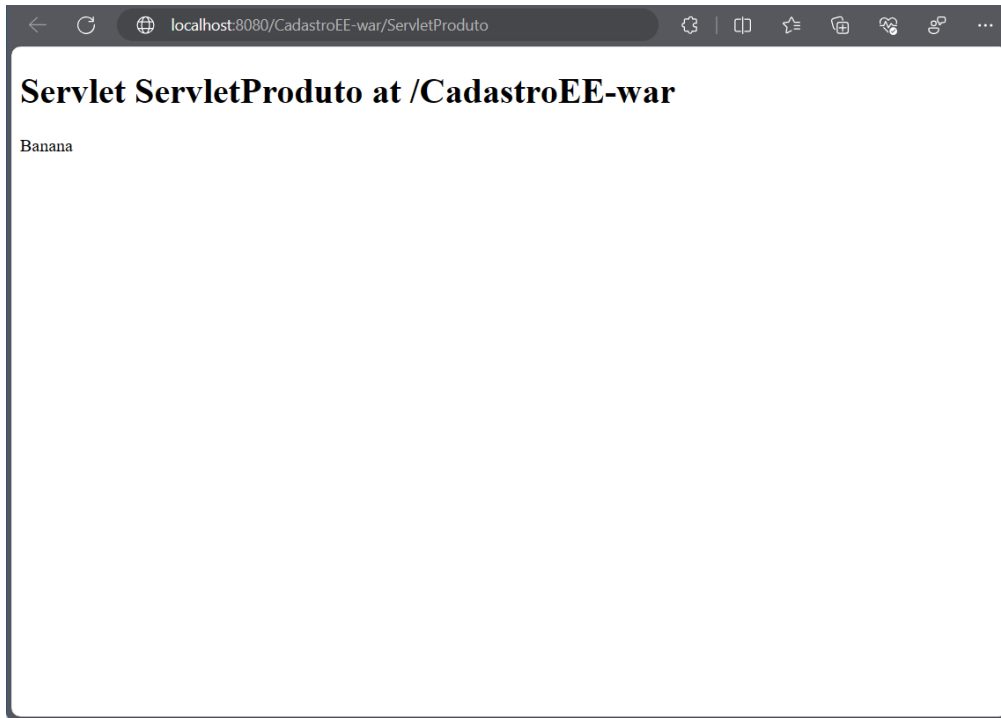
- **Camada de Persistência:** JPA
- **Regras de Negócio:** EJBs (JEE)
- **Interface Web:** Servlets, JSPs
- **Design e Usabilidade:** Bootstrap

Resultado:

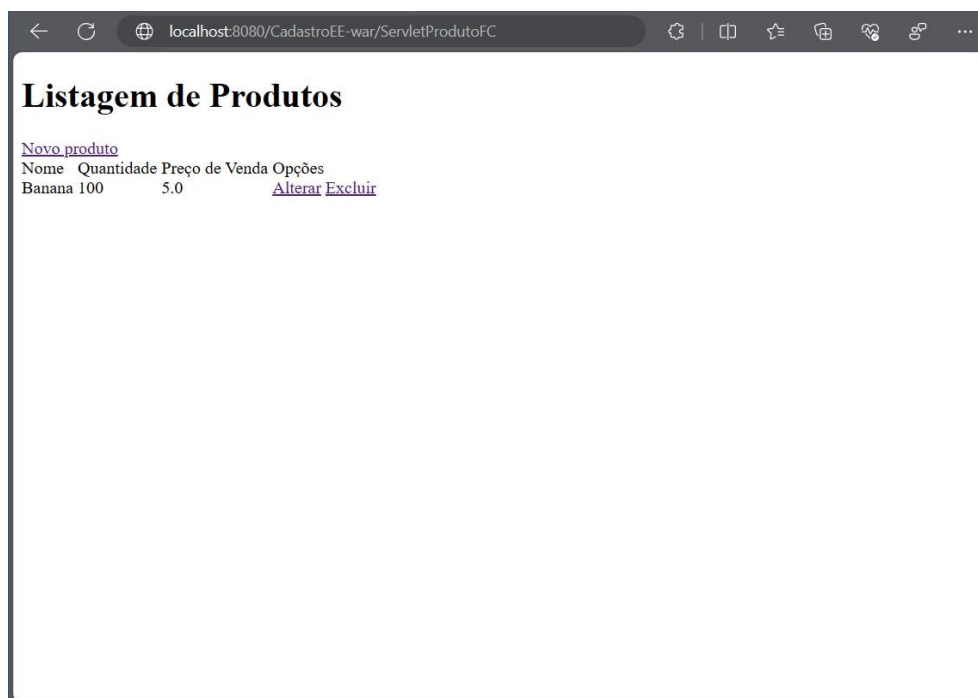
Um sistema cadastral web completo, seguro, amigável e acessível, construído com tecnologias Java robustas e modernas.

Resultados

1º Procedimento | Camadas de persistência e controle



2º Procedimento | Interface cadastral com Servlet e JSPs



← ↻ 🌐 http://localhost:8080/CadastroEE-war/ServletProdutoFC?acao=incl... 🔍 📄 ⭐ 📁 🔄 🗑️ ...

Dados do Produto

Nome: Quantidade: Preço de venda Adicionar produto

3º Procedimento | Melhorando o design da interface

← ↻ 🌐 http://localhost:8080/CadastroEE-war/ServletProdutoFC 🔍 📄 ⭐ 📁 🔄 🗑️ ...

Listagem de Produtos

Novo Produto

Nome	Quantidade	Preço de Venda	Opções
Banana	100	5.0	<button>Alterar</button> <button>Excluir</button>
Laranja	500	2.0	<button>Alterar</button> <button>Excluir</button>
Manga	800	4.0	<button>Alterar</button> <button>Excluir</button>

The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/CadastroEE-war/ServletProdutoFC?acao=incl...`. The main content area has a title **Dados do Produto**. Below the title are three input fields: 'Nome:', 'Quantidade:', and 'Preço de Venda:'. Each field is represented by a white rectangular box with a thin border. At the bottom of the form is a blue button with the text 'Adicionar produto' in white.

Análise e Conclusão

1. Como é organizado um projeto corporativo no NetBeans?

Um projeto corporativo no NetBeans é estruturado de forma hierárquica e modular, separando as diferentes camadas e componentes da aplicação. Essa organização facilita o desenvolvimento, a manutenção e a escalabilidade do projeto, além de promover a colaboração entre os desenvolvedores.

Estrutura Hierárquica:

- **Projeto Principal:** Encapsula os demais subprojetos e define as configurações gerais da aplicação.
- **Subprojetos:** Módulos independentes que representam as diferentes camadas e funcionalidades do projeto.

Exemplo:

- **EAR (Enterprise Archive):** Contém a aplicação completa e os recursos necessários para a sua execução.
- **EJB (Enterprise JavaBeans):** Implementa a lógica de negócio da aplicação.
- **Camada Web:** Interface web da aplicação, composta por Servlets, JSPs, recursos estáticos, etc.

Organização Interna dos Subprojetos:

- **Agrupamento por Funcionalidade:**
 - **EJB:** Entidades JPA, Session Beans e outras classes de lógica de negócio são organizadas em pacotes específicos.
 - **Camada Web:** Servlets, JSPs, CSS, JavaScript e outros recursos são organizados em diretórios e pacotes adequados.
- **Gerenciamento de Dependências:**
 - O NetBeans facilita a inclusão de bibliotecas externas e o gerenciamento de dependências entre os subprojetos.
- **Integração com Controle de Versão:**
 - Integração com sistemas de controle de versão como o Git para um controle eficiente do código-fonte e colaboração entre desenvolvedores.

Benefícios da Organização Modular:

- **Clareza e Separação de Responsabilidades:** Facilita a compreensão, o desenvolvimento e a manutenção do código.
- **Reutilização de Componentes:** Permite a reutilização de componentes em diferentes projetos.
- **Escalabilidade:** Facilita a adição de novas funcionalidades e o crescimento do projeto.
- **Colaboração:** Promove a colaboração entre desenvolvedores, permitindo que trabalhem em diferentes partes do projeto simultaneamente.
- **Qualidade do Software:** Melhora a qualidade do software, pois facilita a identificação e correção de erros.

2. Qual o papel das tecnologias JPA e EJB na construção de um aplicativo para a plataforma Web no ambiente Java?

JPA e EJB: Duas tecnologias essenciais para o desenvolvimento de aplicações web Java robustas e escaláveis.

JPA (Java Persistence API):

- Camada de persistência de dados orientada a objetos.
- Interação com bancos de dados sem SQL.
- Entidades mapeiam objetos para tabelas do banco de dados.
- Operações CRUD (Criar, Ler, Atualizar, Excluir) eficientes.
- Consultas JPQL (Java Persistence Query Language) e relacionamentos entre entidades.

EJB (Enterprise JavaBeans):

- Ambiente de execução seguro e transacional para lógica de negócio.
- Componentes distribuídos implantados em servidores Java EE.
- Controle de transações, segurança, escalabilidade e gerenciamento de concorrência.
- Ideal para implementar lógica de negócio complexa em aplicações web.

Integração JPA + EJB:

- Arquitetura MVC (Model-View-Controller).
- JPA como modelo (dados da aplicação).
- EJB como controlador (lógica de negócio).
- Separação de responsabilidades.
- Manutenção e evolução facilitadas.

Benefícios da Integração:

- Aplicativos web Java robustos e escaláveis.
- Código modular, reutilizável e fácil de manter.
- Desempenho otimizado e segurança aprimorada.
- Base sólida para aplicações web enterprise complexas.

3. Como o NetBeans viabiliza a melhoria de produtividade ao lidar com as tecnologias JPA e EJB?

O NetBeans, como IDE (Ambiente de Desenvolvimento Integrado) para Java, oferece diversos recursos que otimizam o desenvolvimento de aplicações JPA (Java Persistence API) e EJB (Enterprise JavaBeans), impulsionando a produtividade dos desenvolvedores.

Simplificação de Testes e Implantação:

- Suporte direto para servidores de aplicativos Java EE.
- Eliminação de configurações complexas externas.
- Agilidade no processo de teste e implantação de aplicações EJB.

Assistência durante a Codificação:

- Sugestões de código e preenchimento automático.
- Realce de sintaxe para escrita de código mais rápido e preciso.
- Redução de erros e aumento da produtividade.

Geração Automatizada de Código:

- Criação rápida de entidades JPA a partir de tabelas de banco de dados.
- Geração de EJBs a partir de entidades JPA.
- Economia de tempo e esforço para focar em aspectos mais críticos do desenvolvimento.

Depuração Eficaz:

- Suporte completo para depuração de aplicações EJB e JPA.
- Identificação e correção de problemas de lógica de negócios e acesso a dados.
- Garantia da qualidade do software e redução do tempo de resolução de bugs.

Integração com Ferramentas de Gerenciamento de Banco de Dados:

- Criação e manutenção eficientes de bancos de dados utilizados por aplicações JPA.
- Garantia da integridade e do desempenho dos dados da aplicação.

Gerenciamento Simplificado de Dependências e Bibliotecas:

- Adição fácil de bibliotecas JPA e EJB aos projetos.
- Compatibilidade e atualização adequadas das dependências.
- Experiência de desenvolvimento mais fluida e eficaz.

4. O que são Servlets, e como o NetBeans oferece suporte à construção desse tipo de componentes em um projeto Web?

Servlets são componentes Java que expandem as funcionalidades de servidores web, possibilitando o processamento de requisições HTTP. Com eles, é possível criar aplicações web dinâmicas e interativas, capazes de interagir com os clientes e gerar respostas personalizadas com base nas solicitações recebidas. Os Servlets são executados no servidor e podem lidar com diversas tarefas, como receber dados de formulários, acessar bancos de dados, gerar conteúdo HTML dinâmico e muito mais.

O NetBeans, por sua vez, oferece um conjunto robusto de ferramentas para a construção de Servlets em projetos web. Ele simplifica drasticamente o processo de criação, desenvolvimento e implantação de Servlets, tornando o desenvolvimento web mais eficiente e produtivo.

Funcionalidades do NetBeans para Servlets:

- **Criação facilitada:** Assistentes e modelos predefinidos agilizam o processo de criação de Servlets.
- **Integração com o ciclo de vida do projeto:** Facilita o gerenciamento e a organização do projeto.
- **Realce de sintaxe e assistência à codificação:** Reduzem erros e aumentam a produtividade.
- **Depuração e teste facilitados:** Permitem a rápida identificação e correção de problemas.

Benefícios do NetBeans para Servlets:

- **Ambiente de desenvolvimento integrado completo:** Tudo em um só lugar para criar Servlets com eficiência.
- **Aumento da produtividade dos desenvolvedores:** Processo de desenvolvimento mais rápido e menos complexo.
- **Aplicações web robustas e dinâmicas:** Criação de softwares de alta qualidade com mais facilidade.

5. Como é feita a comunicação entre os Servlets e os Session Beans do pool de EJBs?

A comunicação entre Servlets e Session Beans do pool de EJBs é essencial para o desenvolvimento de aplicações Java EE robustas e escaláveis. Duas abordagens principais são utilizadas:

Injeção de Dependência (Preferida):

- Mais simples, menos propensa a erros e fácil de manter.
- O Servlet declara a dependência no Session Bean usando a anotação @EJB.
- O contêiner EJB gerencia a criação e injeção do EJB no Servlet.
- O Servlet acessa métodos e atributos do Session Bean de forma transparente.

Lookup:

- Mais manual e complexo.
- O Servlet obtém uma referência para o Session Bean do pool de EJBs usando JNDI.
- O Servlet precisa conhecer o nome JNDI do EJB.
- Chamadas de método no Session Bean são feitas normalmente.

6. Como funciona o padrão Front Controller, e como ele é implementado em um aplicativo Web Java, na arquitetura MVC?

O padrão Front Controller revoluciona o desenvolvimento de aplicações web Java ao centralizar o processamento de requisições em um único ponto: o controlador frontal. Esse maestro da aplicação atua como intermediário entre o cliente e o restante do sistema, assumindo diversas responsabilidades cruciais:

- **Roteador de requisições:** Recebe todas as solicitações HTTP do cliente e identifica o controlador apropriado para cada uma.
- **Coordenador do fluxo de trabalho:** Orquestra a execução das etapas necessárias para atender à requisição, garantindo uma experiência fluida para o usuário.
- **Gerenciador de tarefas transversais:** Manipula aspectos comuns da aplicação, como autenticação, autorização e tratamento de erros, evitando redundância e otimizando o código.

Na arquitetura MVC (Model-View-Controller), o Front Controller assume o papel fundamental de controlador frontal. Ele intercepta todas as requisições HTTP, determina o controlador responsável por cada uma, invoca o controlador selecionado e facilita a interação entre o controlador, o modelo de dados e a camada de visualização para gerar a resposta adequada ao cliente.

Para associar uma requisição a um controlador específico, o Front Controller utiliza um mecanismo de mapeamento de URLs. Esse mapeamento pode ser feito através de configurações declarativas em arquivos como o web.xml ou por meio de anotações diretamente no código-fonte, proporcionando flexibilidade e facilidade de implementação.

Ao centralizar o controle de fluxo da aplicação e encapsular tarefas transversais, o padrão Front Controller oferece diversos benefícios:

- **Manutenabilidade aprimorada:** Facilita o gerenciamento e a modificação do código, reduzindo o tempo e o esforço necessários para manter a aplicação atualizada.
- **Extensibilidade facilitada:** Permite a adição de novos controladores e funcionalidades com mais simplicidade, tornando a aplicação mais adaptável às necessidades em constante mudança.
- **Teste simplificado:** Facilita a criação de testes unitários e de integração para os controladores, garantindo a qualidade e confiabilidade do código.
- **Separação de preocupações:** Promove a coesão do código ao agrupar a lógica de processamento de requisições em um único local, tornando-o mais organizado e fácil de entender.
- **Organização e legibilidade do código:** Contribui para um código mais estruturado e intuitivo, facilitando a colaboração entre desenvolvedores e a compreensão do sistema.

7. Quais as diferenças e semelhanças entre Servlets e JSPs?

No universo vibrante do desenvolvimento web Java, Servlets e JSPs se apresentam como tecnologias essenciais para a criação de aplicações dinâmicas e interativas. Apesar de compartilharem o mesmo objetivo final, elas divergem em suas abordagens e vocações, exigindo uma análise cuidadosa para a escolha ideal em cada projeto.

Pontos em Comum:

- **Base Java EE:** Tanto Servlets quanto JSPs florescem na plataforma Java EE, proporcionando robustez e escalabilidade para aplicações web.
- **Execução no Servidor:** Ambas residem no servidor, processando requisições HTTP dos clientes e respondendo com conteúdo dinâmico.
- **Geração de Conteúdo Dinâmico:** Conteúdos HTML vibrantes e personalizados surgem da interação com dados do servidor, tornando a experiência do usuário mais rica.

Mergulhando nas Diferenças:

- **Servlets: Arquitetos de Negócios Poderosos:**
 - Classes Java que expandem as capacidades do servidor para lidar diretamente com requisições HTTP.

- Lógica de negócios e processamento de requisições residem em seus domínios, controlando o fluxo da aplicação.
- Separação clara entre lógica e apresentação, promovendo organização e manutenabilidade em projetos complexos.
- Ideal para:
 - Processamento de formulários.
 - Acesso a bancos de dados.
 - Tarefas complexas de negócio.
- **JSPs: Maestros da Apresentação Dinâmica:**
 - Arquivos de texto que combinam código Java com marcação HTML, criando páginas web interativas.
 - Foco na apresentação de conteúdo, permitindo a fácil integração de código Java no HTML.
 - Simplificam a criação de interfaces de usuário dinâmicas.
 - Ideal para:
 - Interfaces de usuário ricas e interativas.
 - Apresentação de dados dinâmicos.
 - Conteúdo web personalizado.

Desempenho:

- Servlets desfrutam de leve vantagem em termos de performance, pois são pré-compilados em bytecode durante a compilação.
- JSPs, por outro lado, precisam ser compilados em Servlets antes da execução, o que pode gerar um pequeno impacto no tempo de resposta.

8. Qual a diferença entre um redirecionamento simples e o uso do método forward, a partir do RequestDispatcher? Para que servem parâmetros e atributos nos objetos HttpRequest?

No universo vibrante do desenvolvimento web Java, Servlets, JSPs, Redirecionamento e Encaminhamento (Forward), Parâmetros e Atributos se configuram como ferramentas essenciais para a construção de aplicações dinâmicas, interativas e escaláveis. Compreender as nuances de cada tecnologia e sua interação é crucial para tomar decisões assertivas e criar softwares de alta qualidade.

Servlets: Arquitetos de Negócios Poderosos

- **Classes Java:** Expandem as capacidades do servidor para processar requisições HTTP.
- **Lógica de Negócios:** Controlam o fluxo da aplicação, manipulando dados e definindo regras de negócio.
- **Separação de Preocupações:** Promovem organização e manutenabilidade em projetos complexos.
- **Indicados para:**
 - Processamento de formulários.
 - Acesso a bancos de dados.
 - Tarefas de negócio complexas.

JSPs: Maestros da Apresentação Dinâmica

- **Arquivos de Texto:** Combinam código Java com HTML para criar páginas web interativas.
- **Foco na Apresentação:** Facilitam a criação de interfaces de usuário dinâmicas e personalizadas.
- **Integração de Código Java:** Permitem a manipulação de dados e lógica de negócio diretamente no HTML.
- **Indicados para:**
 - Interfaces de usuário ricas e interativas.
 - Apresentação de dados dinâmicos.
 - Conteúdo web personalizado.

Redirecionamento vs. Encaminhamento (Forward): Navegando pelas Opções de Fluxo de Controle

- **Redirecionamento:** Envia uma resposta ao cliente com um código de status de redirecionamento e a URL de destino.
 - Transparente para o cliente, que percebe apenas a mudança na URL.
 - Ideal para direcionar o cliente para outra página, especialmente em diferentes servidores ou quando uma nova solicitação é necessária.
- **Encaminhamento (Forward):** Encaminha a solicitação e a resposta internamente para outro recurso no servidor antes de enviar a resposta final.
 - A URL permanece a mesma no navegador.
 - Ideal para processamento interno adicional da resposta no servidor, como validação de dados ou encadeamento de Servlets e JSPs.

Parâmetros e Atributos: Passando Informações em um Ciclo de Vida

- **Parâmetros:**
 - Passam informações de entrada do cliente para o servidor, como dados de formulário ou parâmetros de URL.
 - Acessados usando os métodos `getParameter()` ou `getParameterMap()`.
- **Atributos:**
 - Armazenam dados que podem ser acessados e compartilhados entre diferentes componentes da aplicação durante o ciclo de vida da solicitação.
 - Definidos usando o método `setAttribute()` e acessados usando `getAttribute()`.
 - Úteis para passar dados entre Servlets ou entre Servlets e JSPs durante o encaminhamento (forward).

9. Como o framework Bootstrap é utilizado?

No dinâmico cenário do desenvolvimento web, o Bootstrap se destaca como um framework indispensável para a criação de interfaces de usuário responsivas e visualmente atraentes. Sua relevância reside na oferta de um conjunto robusto de ferramentas e componentes pré-estilizados, como botões, formulários, barras de navegação, cards e grids, prontos para serem facilmente incorporados em páginas HTML, otimizando o processo de desenvolvimento e garantindo resultados consistentes.

Integração Simplificada para Agilidade no Desenvolvimento

A integração do Bootstrap em projetos web é um processo simples e direto, viabilizando a agilidade no desenvolvimento. Os desenvolvedores podem optar por duas formas de inclusão dos arquivos CSS e JavaScript do framework em suas páginas HTML:

- **Download:** Realizar o download direto dos arquivos a partir do site oficial do Bootstrap.
- **CDN (Content Delivery Network):** Utilizar um link CDN para acessar os arquivos hospedados em servidores externos, garantindo carregamento rápido e confiável.

Com os arquivos do Bootstrap integrados, os desenvolvedores estão prontos para construir interfaces com estilo e funcionalidade aprimoradas. As classes CSS e os componentes JavaScript do framework facilitam a criação de elementos web com aparência profissional:

- **Botões:** A classe "btn" permite a estilização de botões com diversos estilos predefinidos, atendendo às necessidades de cada projeto.
- **Formulários:** A classe "form-control" facilita a estilização de inputs de formulários, proporcionando um visual profissional e intuitivo.
- **Layouts Responsivos:** A classe "container" garante a criação de layouts que se adaptam perfeitamente a diferentes tamanhos de tela, desde desktops até smartphones, assegurando uma experiência de usuário otimizada em todos os dispositivos.

Personalização Versátil para Interfaces Únicas

Um dos principais diferenciais do Bootstrap reside em sua alta capacidade de personalização. Os desenvolvedores podem ajustar o visual e o comportamento dos componentes de acordo com as necessidades específicas de seus projetos, garantindo interfaces únicas e alinhadas à identidade da marca:

- **Modificação do CSS:** A edição do código CSS do Bootstrap permite a personalização de cores, fontes, espaçamentos e outros aspectos visuais, criando interfaces que refletem a identidade visual do projeto.
- **Variáveis SASS:** Para uma personalização mais intuitiva, as variáveis SASS fornecidas pelo Bootstrap facilitam a customização de cores, fontes e outros elementos, otimizando o processo de criação de interfaces personalizadas.

Benefícios Essenciais para Projetos de Sucesso

A adoção do Bootstrap em projetos web proporciona diversos benefícios que contribuem para o sucesso do desenvolvimento:

- **Desenvolvimento Acelerado:** A utilização de componentes pré-estilizados e responsivos agiliza o processo de desenvolvimento, otimizando o tempo e o esforço dos desenvolvedores.
- **Coerência Visual:** O framework garante a padronização do estilo em todo o projeto, assegurando uma experiência de usuário consistente e agradável.
- **Responsividade Garantida:** A adaptabilidade dos layouts a diferentes tamanhos de tela garante o acesso fluido e intuitivo à aplicação em desktops, tablets e smartphones.
- **Base Robusta e Confiável:** O Bootstrap se destaca como um framework robusto e confiável, adotado por milhões de desenvolvedores em todo o mundo, garantindo a qualidade e a segurança da aplicação.
- **Comunidade Vibrante:** Uma comunidade ampla e ativa de desenvolvedores oferece suporte, compartilhamento de conhecimentos e acesso às últimas novidades do framework, facilitando a resolução de problemas e o aprimoramento contínuo das interfaces.

10. Por que o Bootstrap garante a independência estrutural do HTML?

O Bootstrap promove a separação de preocupações entre o conteúdo e a apresentação visual do HTML, otimizando o processo de desenvolvimento e a manutenibilidade do código. Essa separação é alcançada através da utilização de classes CSS predefinidas que podem ser facilmente aplicadas aos elementos HTML, sem a necessidade de modificar a estrutura semântica da página.

Com o Bootstrap, os desenvolvedores concentram seus esforços na criação de conteúdo significativo e bem estruturado, enquanto o framework se encarrega de estilizar os elementos HTML de acordo com as diretrizes de design pré-definidas. Isso resulta em um código HTML mais limpo, legível e modular, facilitando a compreensão e a manutenção do projeto ao longo do tempo.

Flexibilidade e Responsividade: Adaptabilidade Garantida

11. Qual a relação entre o Bootstrap e a responsividade da página?

O Bootstrap oferece um sistema de grid flexível que permite aos desenvolvedores criarem layouts responsivos sem a necessidade de definir manualmente estilos CSS para diferentes tamanhos de tela. Essa flexibilidade garante que as interfaces de usuário se adaptem automaticamente a diversos dispositivos, desde desktops até smartphones, proporcionando uma experiência consistente e agradável para todos os usuários.

Com o Bootstrap, o mesmo código HTML pode ser utilizado para criar interfaces que se ajustam perfeitamente a diferentes resoluções de tela, sem a necessidade de adaptações manuais. Essa característica otimiza o processo de desenvolvimento e garante que as interfaces funcionem corretamente em qualquer dispositivo, independentemente do tamanho da tela.