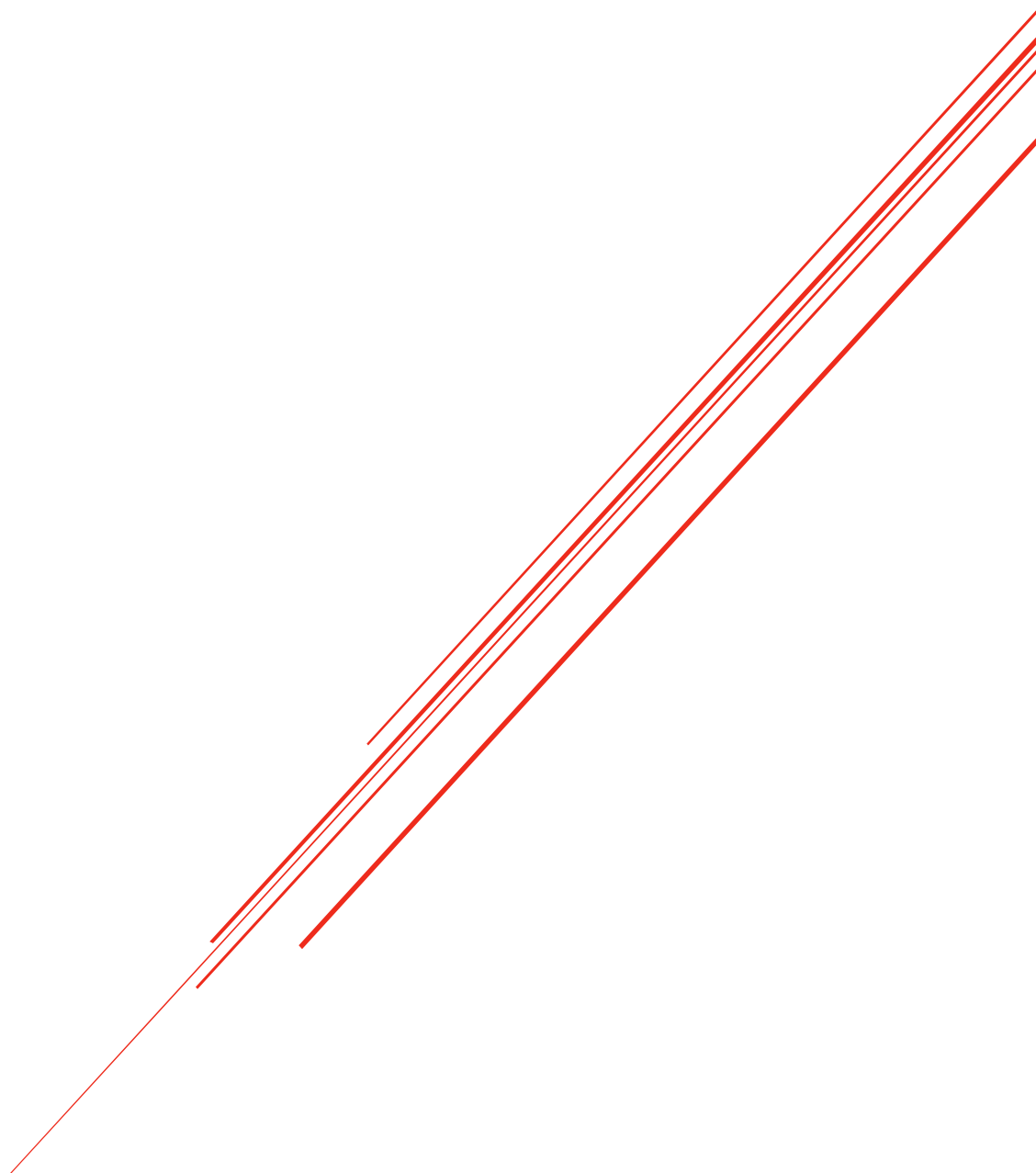


PROJETO FINAL: TABELAS E FORMULÁRIOS INTERATIVOS



Introdução ao Projeto de Tabelas e Formulários Interativos

Olá, estudante de programação!

Bem-vindo a um emocionante projeto gratuito de desenvolvimento web, onde exploraremos a criação de formulários interativos e tabelas dinâmicas.

Este projeto foi cuidadosamente elaborado para proporcionar a você uma experiência prática e envolvente no mundo da programação web.

Durante esta jornada, você terá a oportunidade de desenvolver habilidades valiosas enquanto constrói um projeto passo a passo.

Sobre o Projeto

Neste curso, iremos mergulhar fundo na criação de uma página web que incorpora conceitos essenciais de desenvolvimento front-end.

O projeto consiste em construir uma página responsiva com um formulário interativo que permite que os usuários insiram seus dados pessoais, como nome completo, e-mail e idade. O

s dados inseridos pelos usuários serão exibidos de maneira organizada em uma tabela abaixo do formulário.

Além disso, implementaremos a funcionalidade de limpar os dados da tabela, permitindo que os usuários recomencem o processo conforme desejarem.

O que Você Aprenderá

Ao seguir este projeto passo a passo, você terá a oportunidade de aprimorar suas habilidades em HTML, CSS e JavaScript.

Você aprenderá a estruturar uma página web usando HTML semântico, estilizar a página de maneira atraente e responsiva com CSS e adicionar interatividade usando JavaScript.

Além disso, você compreenderá como manipular eventos, coletar dados de formulários, criar elementos dinamicamente e interagir com o DOM (Modelo de Objeto de Documento) de maneira eficiente.,

Como Usar Este Material

Cada etapa deste projeto foi cuidadosamente elaborada para orientar você em cada passo do processo.

Desde a estruturação do HTML até a adição de comportamentos interativos com JavaScript, você encontrará instruções claras e exemplos práticos.

Sugerimos que siga as etapas na ordem em que foram apresentadas para obter o máximo benefício do aprendizado.

Lembramos que este material é completamente gratuito e foi projetado para promover uma aprendizagem prática e envolvente.

À medida que avança no projeto, não hesite em explorar, experimentar e personalizar elementos de acordo com suas preferências e criatividade.

Agora, é hora de começar! Prepare-se para mergulhar na construção de uma página web interativa, onde os formulários e as tabelas ganharão vida.

Divirta-se, aprenda e compartilhe sua experiência ao longo deste curso. Seja bem-vindo a uma jornada emocionante de desenvolvimento web!

Vamos começar! 

Criando o arquivo HTML

Vamos começar criando um arquivo HTML chamado "Tabela-e-Formulario-Interativos.html".

Em seguida, vou explicar cada parte, linha por linha, para que você possa entender bem. Por último, vou mostrar o código completo.

Ah, e para escrever o código, vou usar o VScode, mas se você preferir, pode usar outro editor que goste mais.

1. **<!DOCTYPE html>**: Isso define o tipo de documento como HTML5, que é a versão mais recente da linguagem de marcação para a web.
2. **<html lang="pt-br">**: Aqui, estamos definindo o elemento **<html>** com o atributo **lang** para indicar que o idioma usado na página é o português do Brasil.
3. **<head>**: A seção **<head>** contém metadados e informações sobre o documento, como o título da página, metatags e referências a arquivos externos.
 - **<meta charset="UTF-8">**: Define a codificação de caracteres para UTF-8, que é uma codificação universal que suporta vários caracteres e símbolos de diferentes idiomas.
 - **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Especifica como o navegador deve ajustar a largura da página para se adequar ao dispositivo do usuário, garantindo uma experiência responsiva.
 - **<link rel="stylesheet" href="styles.css">**: Este link importa o arquivo de estilo CSS chamado "styles.css", que contém as regras de estilo para a página.
 - **<title>Formulário e Tabela Interativos</title>**: Define o título da página exibido na guia do navegador.
4. **<body>**: O elemento **<body>** contém o conteúdo visível da página.
 - **<div class="container">**: Uma **<div>** com a classe "container" é usada para envolver todo o conteúdo da página. Isso é comum em designs responsivos para controlar o layout e a formatação.
 - **<h1>Formulário e Tabela Interativos</h1>**: Um título de nível 1 (h1) que descreve o propósito da página.
 - **<form id="userForm">**: Um formulário que permite que os usuários insiram seus dados.

- **<label for="fullName">Nome completo:</label>**: Uma etiqueta associada ao campo de entrada "fullName".
 - **<input type="text" id="fullName" required>**: Um campo de entrada de texto onde os usuários inserem seu nome completo. O atributo **required** torna esse campo obrigatório.
 - **<label for="email">E-mail:</label>**: Uma etiqueta associada ao campo de entrada "email".
 - **<input type="email" id="email" required>**: Um campo de entrada de email com validação de formato e o atributo **required**.
 - **<label for="age">Idade:</label>**: Uma etiqueta associada ao campo de entrada "age".
 - **<input type="number" id="age" required>**: Um campo de entrada numérico para a idade, também com o atributo **required**.
 - **<button type="submit">Enviar</button>**: Um botão que, quando clicado, envia os dados do formulário para processamento.
 - **<h2>Armazenamento dos dados</h2>**: Um título de nível 2 (h2) que descreve a seção onde os dados inseridos pelos usuários serão exibidos.
 - **<table id="userTable">**: Uma tabela onde os dados dos usuários serão exibidos.
 - **<thead>**: A seção de cabeçalho da tabela.
 - **<tr>**: Uma linha de cabeçalho.
 - **<th>Nome</th>**: Uma célula de cabeçalho que indica "Nome".
 - **<th>E-mail</th>**: Uma célula de cabeçalho que indica "E-mail".
 - **<th>Idade</th>**: Uma célula de cabeçalho que indica "Idade".
 - **<tbody>**: A seção do corpo da tabela, onde as linhas dos dados serão inseridas dinamicamente.
 - **<button id="clearButton">Limpar Dados</button>**: Um botão que, quando clicado, limpará os dados inseridos na tabela.
5. **<script src="script.js"></script>**: Inclui um arquivo JavaScript chamado "script.js", que contém a lógica para tornar a página interativa.

Essa estrutura HTML cria uma página interativa com um formulário para coletar os dados dos usuários e uma tabela para exibir esses dados.

O JavaScript associado ao botão "Enviar" adiciona os dados do formulário à tabela, enquanto o botão "Limpar Dados" reinicia a tabela, permitindo que os usuários recomecem.

Essa abordagem é eficaz para interagir com os visitantes do seu site e fornecer uma experiência envolvente.

Apresentação do código HTML

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Formulário e Tabela Interativos</title>
</head>
<body>
  <div class="container">
    <h1>Formulário e Tabela Interativos</h1>
    <form id="userForm">
      <label for="fullName">Nome completo:</label>
      <input type="text" id="fullName" required>

      <label for="email">E-mail:</label>
      <input type="email" id="email" required>

      <label for="age">Idade:</label>
      <input type="number" id="age" required>

      <button type="submit">Enviar</button>
    </form>
    <h2>Armazenamento dos dados</h2>
    <table id="userTable">
      <thead>
        <tr>
          <th>Nome</th>
          <th>E-mail</th>
          <th>Idade</th>
        </tr>
      </thead>
      <tbody>
        <!-- Linhas da tabela serão adicionadas
dinamicamente aqui -->
      </tbody>
    </table>

    <button id="clearButton">Limpar Dados</button>
  </div>

  <script src="script.js"></script>
</body>
</html>

```

LEconnect

Criando o arquivo CSS

Assim como fizemos ao lidar com o HTML, agora iremos dar continuidade ao processo, desta vez envolvendo o CSS. Vamos criar um arquivo CSS e chamá-lo de `styles.css`.

Embora eu esteja dando um nome genérico, sintá-se à vontade para personalizá-lo de acordo com a sua preferência.

No campo do CSS, pretendo aplicar um estilo bastante simples, uma vez que desejo estimular a sua criatividade para que você possa conferir o seu toque pessoal ao design.

Certamente! O código CSS que você forneceu trata da estilização dos elementos da página HTML. Vou explicar cada parte do código para que você compreenda como os estilos são aplicados aos diferentes elementos:

```
body {  
  font-family: Arial, sans-serif;  
  margin: 0;  
  padding: 0;  
  background-color: #f0f0f0;  
}
```

- Define as configurações básicas para o corpo da página.
- Aplica a fonte "**Arial**" ou uma fonte genérica "**sans-serif**" aos elementos de texto.
- Remove as margens e o preenchimento padrão do corpo.
- Define a cor de fundo como um tom de cinza claro (`#f0f0f0`).

```
.container {  
  max-width: 800px;  
  margin: 0 auto;  
  padding: 20px;  
  background-color: white;  
  border-radius: 8px;  
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
}
```

- Estiliza a div com a classe "**container**", que envolve todo o conteúdo da página.
- Define uma largura máxima de **800px** para o conteúdo e centraliza horizontalmente usando **margin: 0 auto**.

- Aplica preenchimento interno de **20px** para criar espaçamento ao redor do conteúdo.
- Define a cor de fundo da **div** como branca.
- Adiciona um raio de borda de **8px** para suavizar as bordas.
- Aplica uma sombra sutil à **div** para dar uma sensação de elevação.

```
h1, h2 {
  text-align: center;
}

h1:hover{
  text-decoration: underline;
}

h2:hover {
  text-decoration: underline;
}
```

- Centraliza o texto dos títulos **<h1>** e **<h2>**.
- Quando o mouse passa sobre um título **<h1>** ou **<h2>**, ele sublinha o texto para indicar que é clicável.

```
form {
  margin-bottom: 20px;
}

label {
  display: block;
  margin-bottom: 8px;
  font-weight: bold;
}

input {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 4px;
}
```

- Adiciona um espaçamento de **20px** na parte inferior do formulário.

- Define o estilo das etiquetas associadas aos campos de entrada, fazendo com que elas sejam exibidas como blocos.
- Define a largura dos campos de entrada como **100%** para ocupar todo o espaço disponível.
- Adiciona espaçamento interno de **8px** nos campos de entrada.
- Adiciona margem na parte inferior dos campos de entrada para separá-los.

```
button {
  background-color: #007bff;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}
```

- Define o estilo dos botões, incluindo a cor de fundo **azul (#007bff)** e a cor do texto branca.
- Define o preenchimento interno do botão.
- Remove a borda padrão e define um raio de borda de **4px**.
- Altera o cursor para "**pointer**" ao passar o mouse sobre o botão.
- Quando o mouse passa sobre o botão, a cor de fundo muda para um tom mais escuro de **azul (#0056b3)**.

```
table {
  border-collapse: collapse;
  width: 100%;
  margin-bottom: 2rem;
}

th, td {
  border: 1px solid #ddd;
  padding: 0.5rem;
  text-align: left;
}
```

- Define o estilo das tabelas, aplicando a propriedade **border-collapse: collapse** para unir as bordas das células adjacentes.
- Define a largura da tabela como **100%** para ocupar todo o espaço disponível.
- Adiciona margem na parte inferior da tabela para criar um espaçamento.
- Define o estilo das células de cabeçalho **<th>** e células de dados **<td>**.
- Aplica uma borda de **1px** sólida de cor cinza claro (**#ddd**) às células.
- Define um preenchimento interno de **0.5rem** para as células.
- Alinha o conteúdo do texto à esquerda nas células.

Apresentação do código CSS

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f0f0f0;
}

.container {
  max-width: 800px;
  margin: 0 auto;
  padding: 20px;
  background-color: white;
  border-radius: 8px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

h1, h2 {
  text-align: center;
}

h1:hover{
  text-decoration: underline;
}

h2:hover {
  text-decoration: underline;
}

form {
  margin-bottom: 20px;
}

label {
  display: block;
  margin-bottom: 8px;
  font-weight: bold;
}
```

```
input {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

button {
  background-color: #007bff;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

button:hover {
  background-color: #0056b3;
}

/* Estilos da tabela */
table {
  border-collapse: collapse;
  width: 100%;
  margin-bottom: 2rem;
}

th, td {
  border: 1px solid #ddd;
  padding: 0.5rem;
  text-align: left;
}
```

Criando o arquivo JavaScript

Agora, nesta etapa do projeto, criaremos o arquivo JavaScript, que será responsável por realizar a manipulação dos elementos presentes no nosso arquivo HTML.

Vamos prosseguir seguindo a mesma abordagem, na qual detalharei cada linha do código para proporcionar uma compreensão completa.

Quanto ao nome do arquivo, adotarei uma denominação genérica script.js, porém você tem total liberdade para efetuar modificações conforme a sua preferência.

Sem mais delongas, vamos dar início a essa etapa empolgante.

Claro, vou explicar o código JavaScript passo a passo para que você compreenda como ele interage com os elementos HTML e adiciona funcionalidade à sua página.

```
const userForm = document.getElementById('userForm');
const userTable = document.getElementById('userTable');
const clearButton = document.getElementById('clearButton');
```

- Essas linhas obtêm referências para os elementos HTML usando seus IDs.
- **userForm** representa o formulário onde os usuários inserem seus dados.
- **userTable** representa a tabela onde os dados dos usuários serão exibidos.
- **clearButton** representa o botão "Limpar Dados".

```
userForm.addEventListener('submit', function (event) {
    event.preventDefault();

    const fullName = document.getElementById('fullName').value;
    const email = document.getElementById('email').value;
    const age = document.getElementById('age').value;

    const newRow = userTable.insertRow();
    newRow.innerHTML = `
        <td>${fullName}</td>
        <td>${email}</td>
        <td>${age}</td>
    `;

    userForm.reset();
});
```

- Esse bloco de código adiciona um ouvinte de evento ao formulário com o evento "**submit**". Isso significa que quando o usuário clicar no botão "**Enviar**" no formulário, esse código será executado.
- **event.preventDefault()**; previne o comportamento padrão de enviar o formulário, que faria a página ser recarregada.
- **fullName**, **email** e **age** obtêm os valores inseridos nos campos de entrada do formulário usando seus IDs.
- **const newRow = userTable.insertRow()**; cria uma nova linha na tabela e atribui a referência à variável **newRow**.
- O trecho de código dentro de **newRow.innerHTML** cria as células da tabela com os valores do nome, e-mail e idade que foram inseridos.
- **userForm.reset()**; limpa os campos do formulário após os dados terem sido enviados.

```
clearButton.addEventListener('click', function () {
  userTable.innerHTML = `
    <thead>
      <tr>
        <th>Nome</th>
        <th>E-mail</th>
        <th>Idade</th>
      </tr>
    </thead>
    <tbody>
      <!-- Linhas da tabela serão adicionadas dinamicamente
aqui -->
    </tbody>
  `;
});
```

- Esse bloco de código adiciona um ouvinte de evento ao botão "**Limpar Dados**" com o evento "**click**". Isso significa que quando o usuário clicar no botão, esse código será executado.
- Dentro da função, o código redefine o conteúdo da tabela para o seu estado inicial, com um cabeçalho vazio e uma área de corpo vazia para as linhas dos dados.
- Isso efetivamente limpa os dados exibidos na tabela, reiniciando-a para um estado vazio.

Apresentação do código JavaScript

```

const userForm = document.getElementById('userForm');
const userTable = document.getElementById('userTable');
const clearButton = document.getElementById('clearButton');

userForm.addEventListener('submit', function (event) {
    event.preventDefault();

    const fullName = document.getElementById('fullName').value;
    const email = document.getElementById('email').value;
    const age = document.getElementById('age').value;

    const newRow = userTable.insertRow();
    newRow.innerHTML = `
        <td>${fullName}</td>
        <td>${email}</td>
        <td>${age}</td>
    `;

    userForm.reset();
});

clearButton.addEventListener('click', function () {
    userTable.innerHTML = `
        <thead>
            <tr>
                <th>Nome</th>
                <th>E-mail</th>
                <th>Idade</th>
            </tr>
        </thead>
        <tbody>
            <!-- Linhas da tabela serão adicionadas dinamicamente
            aqui -->
        </tbody>
    `;
});

```

Considerações finais

Parabéns, caro aprendiz!

Você concluiu com sucesso o nosso emocionante projeto de desenvolvimento web, onde exploramos a criação de formulários interativos e tabelas dinâmicas.

Esperamos que essa jornada tenha sido enriquecedora e gratificante, proporcionando a você um mergulho profundo em conceitos cruciais de programação front-end.

Através deste projeto, você teve a oportunidade de adquirir habilidades valiosas em HTML, CSS e JavaScript.

Desde a estruturação do layout até a implementação de funcionalidades interativas, você percorreu um caminho repleto de aprendizado prático.

Essas habilidades podem ser aplicadas em uma variedade de projetos futuros, impulsionando ainda mais sua jornada no mundo do desenvolvimento web.

Compartilhe sua Experiência

Acreditamos que o compartilhamento é uma das melhores maneiras de aprender e crescer juntos.

Se você gostou deste projeto e achou valiosa a experiência de construir formulários interativos e tabelas dinâmicas, convidamos você a compartilhar seu sucesso com seus amigos, colegas e nas redes sociais.

Seu compartilhamento pode inspirar outros entusiastas a se aventurarem no mundo da programação web!

Avalie o Curso

Sua opinião é extremamente importante para nós! Se você puder dedicar alguns minutos para avaliar este curso, isso nos ajudará a entender como podemos continuar aprimorando nosso conteúdo para atender melhor às suas necessidades.

Suas sugestões e feedback são fundamentais para construirmos um ambiente de aprendizado cada vez mais eficaz.

Agradecemos a você por escolher participar deste projeto de desenvolvimento web. Esperamos que você continue explorando, aprendendo e expandindo suas habilidades no mundo da programação.

Lembre-se de que a jornada do desenvolvimento é contínua e cheia de oportunidades emocionantes.

Com gratidão, Equipe **LEconnect**

Faça outro curso

