

**CENTRO UNIVERSITÁRIO CATÓLICA DO LESTE DE
MINAS GERAIS**

CAMPUS CORONEL FABRICIANO

Documentação de Desenvolvimento de Projeto

Seminário Materialize

Alunos:

Raphael Vilete Sete Coelho Silva – RA: A06169860

Samuel dos Passos Andrade – RA: A06169739

Antônio Erick Silva Oliveira – RA: A06169780

Coronel Fabriciano - MG

1º semestre 2024

Seminário Materialize

1. Introdução ao Materialize

- **História e propósito do desenvolvimento do Materialize:**

O Materialize é um framework front-end que surgiu como uma resposta às diretrizes de design inovadoras do Google, conhecidas como Material Design. Lançado para simplificar e enriquecer a criação de interfaces de usuário modernas e responsivas, o Materialize oferece uma variedade de componentes e estilos prontos para uso. Seu propósito é facilitar o desenvolvimento web, seguindo padrões de design contemporâneos e alinhando-se às demandas da web moderna. Desde seu lançamento, o Materialize tem desempenhado um papel significativo no panorama do desenvolvimento web, influenciando inúmeras plataformas e projetos.

- **Principais características e benefícios:**

- Componentes Prontos para Uso:

O Materialize oferece uma ampla gama de componentes prontos para uso, como botões, cards, barras de navegação, formulários e muito mais. Isso permite aos desenvolvedores construírem interfaces de usuário de forma rápida e eficiente, sem a necessidade de criar cada elemento do zero.

- Responsividade:

Todos os componentes do Materialize são projetados para serem responsivos, o que significa que se adaptam automaticamente a diferentes tamanhos de tela e dispositivos. Isso garante uma experiência consistente e amigável para o usuário, independentemente do dispositivo que estão usando.

- Estilo baseado no Material Design:

O Materialize segue as diretrizes de design do Material Design do Google, o que resulta em interfaces de usuário modernas, limpas e visualmente atraentes. Isso não apenas melhora a estética do site ou aplicativo, mas também proporciona uma experiência de usuário mais intuitiva e coesa.

- Facilidade de Customização:

Embora o Materialize forneça estilos e componentes prontos para uso, também oferece flexibilidade para personalização. Os desenvolvedores

Seminário Materialize

podem facilmente ajustar cores, fontes e outros estilos para atender às necessidades específicas de seus projetos.

- Suporte a Funcionalidades Avançadas:

Além dos componentes básicos, o Materialize oferece suporte a funcionalidades avançadas, como animações e transições suaves. Isso permite criar interfaces de usuário mais dinâmicas e interativas, melhorando a experiência do usuário final.

- Documentação Abundante e Comunidade Ativa:

O Materialize é bem documentado, com guias e exemplos detalhados disponíveis para os desenvolvedores. Além disso, possui uma comunidade ativa de usuários e contribuidores, o que facilita o suporte e a troca de conhecimentos.

- Leveza e Desempenho:

Apesar de oferecer uma ampla gama de recursos, o Materialize é leve e otimizado para garantir um desempenho rápido e eficiente. Isso é crucial para garantir uma experiência de usuário fluida e sem interrupções.

Essas características e benefícios fazem do Materialize uma escolha popular entre os desenvolvedores que buscam uma maneira rápida e eficaz de criar interfaces de usuário modernas e responsivas.

Seminário Materialize

2. Princípios Básicos do Materialize:

- Demonstração prática do sistema de grade responsiva:

Exemplo prático

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Exemplo de Grade Responsiva com Materialize</title>
8      <!-- Incluindo os arquivos CSS do Materialize -->
9      <link rel="stylesheet"
10         href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
11  </head>
12
13  <body>
14      <!-- Container principal -->
15      <div class="container">
16          <!-- Linha para conter as colunas -->
17          <div class="row">
18              <!-- Coluna 1 em telas pequenas, metade da
19                  largura em telas médias e um terço em telas grandes -->
20              <div class="col s12 m6 l4">Coluna 1</div>
21              <!-- Coluna 2 em telas pequenas, metade da
22                  largura em telas médias e um terço em telas grandes -->
23              <div class="col s12 m6 l4">Coluna 2</div>
24              <!-- Coluna 3 em telas pequenas, metade da
25                  largura em telas médias e um terço em telas grandes -->
26              <div class="col s12 m6 l4">Coluna 3</div>
27          </div>
28      </div>
29      <!-- Incluindo o arquivo JavaScript do Materialize -->
30      <script
31         src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js">
32      </script>
33  </body>
34
35  </html>
```

Seminário Materialize

- Exemplos de componentes pré-estilizados, como botões e formulários:

Exemplo prático botões

```
1 <div class="container">
2   <!-- Botão pré-estilizado do Materialize -->
3   <a class="waves-effect waves-light btn">Botão</a>
4   <!-- 'waves-effect waves-light' são classes do
5       Materialize para aplicar efeitos visuais ao botão -->
6 </div>
```

Exemplo prático botões

```
1 <div class="container">
2   <!-- Formulário pré-estilizado do Materialize -->
3   <form>
4     <!-- 'input-field' é uma classe do Materialize para
5         estilizar o campo de entrada e seu rótulo associado -->
6     <div class="input-field">
7       <input id="nome" type="text" class="validate">
8       <label for="nome">Nome</label>
9     </div>
10    <div class="input-field">
11      <!-- 'validate' é uma classe do Materialize para aplicar validação
12          ao campo de entrada (como verificar se o e-mail é válido) -->
13      <input id="email" type="email" class="validate">
14      <label for="email">E-mail</label>
15    </div>
16    <!-- 'btn waves-effect waves-light' são classes do
17        Materialize para estilizar o botão e aplicar efeitos visuais -->
18    <button class="btn waves-effect waves-light" type="submit" name="action">
19      Enviar
20    </button>
21  </form>
22 </div>
```

3. Layouts e Responsividade

- **Apresentação das classes e utilitários para criar layouts responsivos:**

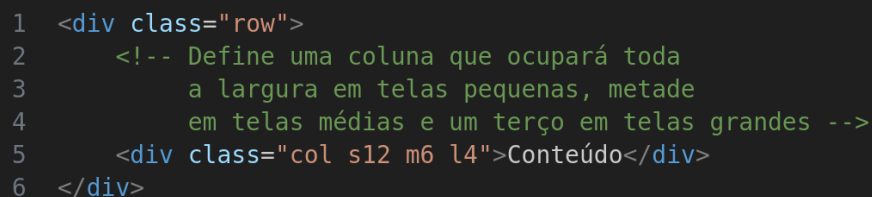
Ao desenvolver um website ou aplicativo, é fundamental garantir uma experiência de usuário consistente e agradável em diferentes dispositivos e tamanhos de tela. Para alcançar esse objetivo, o Materialize oferece um conjunto abrangente de classes e utilitários para criar layouts responsivos de forma eficiente e elegante.

- **Grade Responsiva:**

O Materialize utiliza um sistema de grade responsiva baseado em classes CSS para criar layouts flexíveis que se adaptam a diferentes tamanhos de tela. As principais classes são:

- **row:** Define uma linha para conter as colunas.
- **col:** Define uma coluna dentro de uma linha.
- **s:** Pequeno (small) - para telas pequenas.
- **m:** Médio (medium) - para telas médias.
- **l:** Grande (large) - para telas grandes.

Exemplo prático:



```
1 <div class="row">
2   <!-- Define uma coluna que ocupará toda
3       a largura em telas pequenas, metade
4       em telas médias e um terço em telas grandes -->
5   <div class="col s12 m6 l4">Conteúdo</div>
6 </div>
```

Seminário Materialize

- Ocultar e Exibir em Diferentes Tamanhos de Tela:

Você pode controlar a visibilidade de elementos em diferentes tamanhos de tela usando as classes `hide` e `show` seguidas do tamanho de tela desejado:

- `hide`: Oculta o elemento em determinado tamanho de tela.
- `show`: Exibe o elemento apenas em determinado tamanho de tela.

Exemplo prático

```
1 <!-- Este conteúdo é oculto em telas pequenas -->
2 <div class="hide-on-small-only">
3     Este conteúdo é oculto em telas pequenas
4 </div>
5 <!-- Este conteúdo é exibido apenas em telas médias -->
6 <div class="show-on-medium">
7     Este conteúdo é exibido apenas em telas médias
8 </div>
```

- Alinhamento de Conteúdo:

Para alinhar conteúdo horizontalmente, você pode usar as classes `center-align`, `left-align` e `right-align`. Para alinhar verticalmente, pode-se usar as classes `valign-wrapper` e `valign`.

Exemplo prático

```
1 <!-- Alinha o conteúdo centralmente horizontalmente -->
2 <div class="center-align">Conteúdo centralizado horizontalmente</div>
3 <!-- Alinha o conteúdo centralmente verticalmente -->
4 <div class="valign-wrapper" style="height: 100px;">
5     <div class="valign">Conteúdo centralizado verticalmente</div>
6 </div>
```

Seminário Materialize

- Exemplos de designs adaptáveis desenvolvidos com Materialize:

Exemplo 1: Página de Perfil de Usuário Responsiva

```
1 <div class="container">
2   <!-- Foto de perfil responsiva -->
3   
4   <!-- Detalhes do usuário -->
5   <div class="row">
6     <div class="col s12">
7       <h5>Nome do Usuário</h5>
8       <p>Email: usuario@email.com</p>
9       <p>Biografia: Lorem ipsum dolor sit amet...</p>
10    </div>
11  </div>
12  <!-- Botão para editar perfil -->
13  <div class="row">
14    <div class="col s12">
15      <!-- Botão pré-estilizado do Materialize -->
16      <a class="waves-effect waves-light btn">Editar Perfil</a>
17    </div>
18  </div>
19 </div>
```

Este exemplo mostra uma página de perfil de usuário responsiva, exibindo informações do usuário e um botão para editar o perfil.

Recursos Utilizados:

- Imagem de perfil responsiva (.responsive-img).
- Layout responsivo usando a classe de grade do Materialize (col s12).
- Botão pré-estilizado do Materialize (.btn).

Seminário Materialize

Exemplo 2: Lista de Tarefas Responsiva

```
1 <div class="container">
2   <!-- Campo de entrada para adicionar tarefas -->
3   <div class="row">
4     <div class="input-field col s12">
5       <input id="tarefa" type="text" class="validate">
6       <label for="tarefa">Adicionar Tarefa</label>
7     </div>
8   </div>
9   <!-- Lista de tarefas responsiva -->
10  <ul class="collection">
11    <!-- Item da lista de tarefas -->
12    <li class="collection-item">Tarefa 1 <a href="#" class="secondary-content"><i
13      class="material-icons">done</i></a></li>
14    <li class="collection-item">Tarefa 2 <a href="#" class="secondary-content"><i
15      class="material-icons">done</i></a></li>
16    <li class="collection-item">Tarefa 3 <a href="#" class="secondary-content"><i
17      class="material-icons">done</i></a></li>
18  </ul>
19 </div>
```

Este exemplo mostra uma lista de tarefas responsiva, onde os usuários podem adicionar novas tarefas e marcar as concluídas.

Recursos Utilizados:

- Campo de entrada de texto para adicionar tarefas (.input-field).
- Lista de tarefas responsiva utilizando a classe de coleção do Materialize (collection).
- Ícone de tarefa concluída utilizando ícones do Materialize (material-icons).

Seminário Materialize

4. Estilização e Temas Personalizados

- Demonstração de como customizar estilos usando opções do Materialize

Exemplo prático

```
1 <div class="container">
2   <!-- Botão customizado -->
3   <a class="waves-effect waves-light btn btn-custom">
4     Botão Personalizado
5   </a>
6   <!-- Texto customizado -->
7   <p class="text-custom">
8     Este é um texto personalizado com fonte Arial e cor preta.
9   </p>
10 </div>
```

```
1 <style>
2   /* Customizando a cor dos botões */
3   .btn-custom {
4     background-color: #ff6f00;
5     /* Cor laranja */
6   }
7
8   /* Customizando a fonte do texto */
9   .text-custom {
10    font-family: 'Arial', sans-serif;
11    /* Fonte Arial */
12    color: #333;
13    /* Cor preta */
14  }
15 </style>
```

Neste exemplo, customizamos os estilos dos botões e do texto usando CSS embutido. Definimos uma nova classe `btn-custom` para os botões, alterando a cor de fundo para laranja. Para o texto, criamos uma classe `text-custom`, onde definimos a fonte como Arial e a cor do texto como preta.

Seminário Materialize

- Criação de um tema personalizado para um exemplo prático

Exemplo prático

```
1 <div class="container">
2   <h3>Meu Tema Personalizado</h3>
3   <p>
4     Este é um exemplo prático de como criar
5     um tema personalizado usando o Materialize.
6   </p>
7   <!-- Botão com cor personalizada -->
8   <a class="waves-effect waves-light btn btn-custom">
9     Botão Personalizado
10  </a>
11 </div>
12 <!-- In
```

```
1 <!-- Definindo as cores do tema -->
2 <style>
3   /* Cor primária personalizada */
4   :root {
5     --cor-primaria: #4caf50;
6     /* Verde */
7   }
8
9   /* Cor secundária personalizada */
10  .btn-custom {
11    background-color: var(--cor-primaria);
12  }
13
14  /* Cor de fundo personalizada */
15  body {
16    background-color: #f5f5f5;
17    /* Cinza claro */
18  }
19 </style>
```

Neste exemplo, criamos um tema personalizado definindo as cores primária, secundária e de fundo. Utilizamos a variável `--cor-primaria` para a cor primária e a aplicamos aos botões com a classe `btn-custom`. Além disso, definimos a cor de fundo do corpo da página como cinza claro.

5. Componentes Avançados do Materialize

- Exploração de recursos avançados, como carrosséis, modais e tabs:

O Materialize é um framework front-end moderno que oferece uma variedade de componentes e recursos para criar interfaces web elegantes e funcionais. Além dos elementos básicos de design, o Materialize também fornece uma série de recursos avançados que podem elevar a experiência do usuário e tornar suas aplicações mais dinâmicas e interativas.

Neste contexto, vamos explorar três recursos avançados oferecidos pelo Materialize: carrosséis, modais e tabs. Esses recursos permitem criar elementos como apresentações de slides, janelas modais e sistemas de navegação por abas, adicionando camadas de interatividade e usabilidade aos seus projetos.

Exemplo Carousel (Carrossel)

```
1 <div class="carousel">
2   <!-- Slide 1 -->
3   <a class="carousel-item" href="#one!">
4     
5   </a>
6   <!-- Slide 2 -->
7   <a class="carousel-item" href="#two!">
8     
9   </a>
10  <!-- Slide 3 -->
11  <a class="carousel-item" href="#three!">
12    
13  </a>
14 </div>
15
16 <!-- Incluindo o arquivo JavaScript do Materialize -->
17 <script
18   src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js">
19 </script>
20 <!-- Inicializando o carrossel -->
21 <script>
22   document.addEventListener('DOMContentLoaded', function () {
23     var elems = document.querySelectorAll('.carousel');
24     var instances = M.Carousel.init(elems, {});
25   });
26 </script>
```

Seminário Materialize

Exemplo Modal

```
1 <!-- Botão para acionar o modal -->
2 <a class="waves-effect waves-light btn modal-trigger" href="#modal1">Abrir Modal</a>
3
4 <!-- Estrutura do modal -->
5 <div id="modal1" class="modal">
6   <div class="modal-content">
7     <h4>Meu Modal</h4>
8     <p>Este é um modal simples com o Materialize.</p>
9   </div>
10  <div class="modal-footer">
11    <a href="#" class="modal-close waves-effect waves-green btn-flat">Fechar</a>
12  </div>
13 </div>
14
15 <!-- Incluindo o arquivo JavaScript do Materialize -->
16 <script
17   src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js">
18 </script>
19 <!-- Inicializando o modal -->
20 <script>
21   document.addEventListener('DOMContentLoaded', function () {
22     var elems = document.querySelectorAll('.modal');
23     var instances = M.Modal.init(elems, {});
24   });
25 </script>
```

Exemplo Tabs

```
1 <div class="row">
2   <div class="col s12">
3     <ul class="tabs">
4       <li class="tab col s3"><a href="#test1">Tab 1</a></li>
5       <li class="tab col s3"><a class="active" href="#test2">Tab 2</a></li>
6       <li class="tab col s3 disabled"><a href="#test3">Tab 3</a></li>
7     </ul>
8   </div>
9   <div id="test1" class="col s12">Conteúdo da Tab 1</div>
10  <div id="test2" class="col s12">Conteúdo da Tab 2</div>
11  <div id="test3" class="col s12">Conteúdo da Tab 3</div>
12 </div>
13
14 <!-- Incluindo o arquivo JavaScript do Materialize -->
15 <script
16   src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js">
17 </script>
18 <!-- Inicializando as tabs -->
19 <script>
20   document.addEventListener('DOMContentLoaded', function () {
21     var elems = document.querySelectorAll('.tabs');
22     var instances = M.Tabs.init(elems, {});
23   });
24 </script>
```

Estes exemplos demonstram como implementar carrosséis, modais e tabs utilizando o Materialize em suas aplicações. Cada um desses recursos oferece uma maneira elegante e funcional de apresentar informações e interagir com os usuários.

Seminário Materialize

- Demonstração prática de integração desses componentes em projetos:

Exemplo prático

```
1 <div class="container">
2   <h4>Integração de Componentes com Materialize</h4>
3
4   <!-- Carrossel -->
5   <div class="carousel">
6     <a class="carousel-item" href="#one!">
7       
8     </a>
9     <a class="carousel-item" href="#two!">
10      
11    </a>
12    <a class="carousel-item" href="#three!">
13      
14    </a>
15  </div>
16
17  <!-- Modal -->
18  <a class="waves-effect waves-light btn modal-trigger" href="#modal1">
19    Abrir Modal
20  </a>
21  <div id="modal1" class="modal">
22    <div class="modal-content">
23      <h4>Meu Modal</h4>
24      <p>Este é um modal simples com o Materialize.</p>
25    </div>
26    <div class="modal-footer">
27      <a href="#" class="modal-close waves-effect waves-green btn-flat">
28        Fechar
29      </a>
30    </div>
31  </div>
32
33  <!-- Tabs -->
34  <div class="row">
35    <div class="col s12">
36      <ul class="tabs">
37        <li class="tab col s3"><a href="#tab1">Tab 1</a></li>
38        <li class="tab col s3"><a href="#tab2">Tab 2</a></li>
39        <li class="tab col s3"><a href="#tab3">Tab 3</a></li>
40      </ul>
41    </div>
42    <div id="tab1" class="col s12">
43      <h5>Conteúdo da Tab 1</h5>
44      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
45    </div>
46    <div id="tab2" class="col s12">
47      <h5>Conteúdo da Tab 2</h5>
48      <p>Sed do eiusmod tempor incididunt ut labore et dolore aliqua.</p>
49    </div>
50    <div id="tab3" class="col s12">
51      <h5>Conteúdo da Tab 3</h5>
52      <p>Ut enim ad minim veniam, quis nostrud exercitation.</p>
53    </div>
54  </div>
55 </div>
56
```

Seminário Materialize

```
1 <!-- Inicializando os componentes -->
2 <script>
3     document.addEventListener('DOMContentLoaded', function () {
4         // Inicializando o carrossel
5         var carouselElems = document.querySelectorAll('.carousel');
6         var carouselInstances = M.Carousel.init(carouselElems, {});
7
8         // Inicializando o modal
9         var modalElems = document.querySelectorAll('.modal');
10        var modalInstances = M.Modal.init(modalElems, {});
11
12        // Inicializando as tabs
13        var tabsElems = document.querySelectorAll('.tabs');
14        var tabsInstances = M.Tabs.init(tabsElems, {});
15    });
16 </script>
```

```
1 <!-- Estilos personalizados -->
2 <style>
3     /* Ajustes de margem para aprimorar o layout */
4     .carousel {
5         margin-top: 20px;
6     }
7
8     .tabs-content.carousel .carousel {
9         margin-top: 0;
10    }
11 </style>
```

Este exemplo integra um carrossel, um modal e tabs em uma única página. Os usuários podem interagir com o carrossel, abrir o modal e navegar pelas diferentes abas para acessar conteúdo específico.

Seminário Materialize

6. Integração com HTML5

- **Estratégias para incorporar recursos do Materialize em páginas HTML5:**

Incorporar recursos do Materialize em páginas HTML5 pode ser feito de várias maneiras. Aqui estão algumas estratégias:

- **Utilização de CDN:**

A maneira mais simples de incorporar o Materialize em páginas HTML5 é através do uso de um Content Delivery Network (CDN). Os arquivos CSS e JavaScript do Materialize podem ser incluídos diretamente no documento HTML5 usando URLs fornecidos por um CDN, como o Cloudflare ou o CDNJS.

Exemplo prático:



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Minha Página com Materialize</title>
8   <!-- Incluindo os arquivos CSS do Materialize via CDN -->
9   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
10 </head>
11
12 <body>
13   <!-- Conteúdo aqui -->
14
15   <!-- Incluindo o arquivo JavaScript do Materialize via CDN -->
16   <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
17 </body>
18
19 </html>
```

- **Instalação via Gerenciador de Pacotes:**

Para quem usa um gerenciador de pacotes como npm (Node Package Manager), o Materialize pode ser instalado como uma dependência. Isso permite gerenciar facilmente as atualizações e versões do Materialize. Depois da instalação, os arquivos CSS e JavaScript podem ser incluídos como recursos locais no HTML5.

Seminário Materialize

- Personalização do Tema:

O Materialize oferece a opção de personalizar um próprio tema usando o site oficial. Escolhendo cores primária e secundária, tipografia e outras opções, é possível criar um tema personalizado que combine com o estilo da aplicação. Após a personalização do tema, os arquivos CSS e JavaScript personalizados podem ser baixados e incluídos no HTML5.

- Construção Customizada:

Para quem precisa de um controle mais granular sobre quais componentes do Materialize deseja incluir em um projeto, pode-se utilizar a ferramenta de construção customizada fornecida pelo Materialize. Com essa ferramenta, é possível selecionar apenas os componentes específicos necessários e gerar arquivos CSS e JavaScript personalizados para uso no HTML5.

Independentemente da estratégia escolhida, é importante seguir as práticas recomendadas de desenvolvimento web, como a otimização de arquivos estáticos e o uso de ferramentas de automação, para garantir um desempenho e uma experiência de usuário otimizados em páginas HTML5.

Seminário Materialize

- Exemplos de códigos de integração:

Exemplo 1: Página simples com Materialize

```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <title>Página com Materialize</title>
5   <!-- Incluindo os arquivos CSS do Materialize via CDN -->
6   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
7 </head>
8
9 <body>
10  <div class="container">
11    <h1>Bem-vindo ao Materialize!</h1>
12    <p>Este é um exemplo simples de integração do Materialize em uma página HTML.</p>
13    <a class="waves-effect waves-light btn">Botão Materialize</a>
14  </div>
15
16  <!-- Incluindo o arquivo JavaScript do Materialize via CDN -->
17  <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
18 </body>
```

Exemplo 2: Navbar responsiva com Materialize

```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <title>Navbar com Materialize</title>
5   <!-- Incluindo os arquivos CSS do Materialize via CDN -->
6   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
7 </head>
8
9 <body>
10  <nav>
11    <div class="nav-wrapper">
12      <a href="#" class="brand-logo">Logo</a>
13      <ul id="nav-mobile" class="right hide-on-med-and-down">
14        <li><a href="#">Item 1</a></li>
15        <li><a href="#">Item 2</a></li>
16        <li><a href="#">Item 3</a></li>
17      </ul>
18    </div>
19  </nav>
20
21  <!-- Incluindo o arquivo JavaScript do Materialize via CDN -->
22  <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
23 </body>
```

7. Aplicações Práticas

- Apresentação de Projetos Práticos com Materialize e HTML5:

Exemplo - Projeto 1: Página de Perfil de Usuário

```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <title>Perfil de Usuário</title>
5   <!-- Incluindo os arquivos CSS do Materialize via CDN -->
6   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
7 </head>
8
9 <body>
10   <div class="container">
11     <div class="row">
12       <div class="col s12 m6">
13         <div class="card">
14           <div class="card-image">
15             
16             <span class="card-title">John Doe</span>
17           </div>
18           <div class="card-content">
19             <p>Email: john@example.com</p>
20             <p>Telefone: 123-456-7890</p>
21           </div>
22           <div class="card-action">
23             <a href="#">Editar Perfil</a>
24           </div>
25         </div>
26       </div>
27     </div>
28   </div>
29
30   <!-- Incluindo o arquivo JavaScript do Materialize via CDN -->
31   <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
32 </body>
```

Seminário Materialize

Exemplo - Projeto 2: Lista de Tarefas

```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <title>Lista de Tarefas</title>
5   <!-- Incluindo os arquivos CSS do Materialize via CDN -->
6   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
7 </head>
8
9 <body>
10  <div class="container">
11    <h4>Lista de Tarefas</h4>
12    <ul class="collection">
13      <li class="collection-item">Estudar para o exame</li>
14      <li class="collection-item">Fazer compras no mercado</li>
15      <li class="collection-item">Ligar para o cliente</li>
16    </ul>
17    <div class="input-field">
18      <input id="tarefa" type="text" class="validate">
19      <label for="tarefa">Adicionar Nova Tarefa</label>
20    </div>
21    <button class="btn waves-effect waves-light" onclick="adicionarTarefa()">Adicionar</button>
22  </div>
```

```
1 <!-- Incluindo o arquivo JavaScript do Materialize via CDN -->
2 <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
3 <script>
4   function adicionarTarefa() {
5     var tarefa = document.getElementById('tarefa').value;
6     var lista = document.querySelector('.collection');
7     var novaTarefa = document.createElement('li');
8     novaTarefa.classList.add('collection-item');
9     novaTarefa.textContent = tarefa;
10    lista.appendChild(novaTarefa);
11    document.getElementById('tarefa').value = '';
12  }
13 </script>
14 </body>
```

- **Discussão sobre desafios enfrentados e soluções implementadas:**

- **Desafio 1: Responsividade da Página de Perfil de Usuário:**

Um dos desafios enfrentados foi garantir que a página de perfil de usuário fosse responsiva e se adaptasse bem a diferentes tamanhos de tela. A solução foi usar as classes de grid do Materialize (por exemplo, `col s12 m6`) para definir o layout da página e garantir que os elementos fossem exibidos corretamente em dispositivos móveis e desktops.

Seminário Materialize

- Desafio 2: Adição Dinâmica de Tarefas na Lista de Tarefas:

O desafio aqui era permitir que os usuários adicionassem novas tarefas à lista de forma dinâmica, sem recarregar a página. A solução foi usar JavaScript para capturar o valor do campo de entrada, criar um novo elemento de lista e adicioná-lo à lista existente quando o botão "Adicionar" fosse clicado. Isso proporcionou uma experiência de usuário mais fluida e interativa.

8. Boas Práticas de Desenvolvimento

- **Destaque para dicas e recomendações para escrever código limpo e eficiente:**
 - **Nomenclatura Descritiva:**

Use nomes de variáveis, funções e classes que descrevam claramente seu propósito e função no código. Isso facilita a compreensão do código por outros desenvolvedores e torna sua manutenção mais fácil.
 - **Comentários Significativos:**

Use comentários para explicar partes complicadas do código, fornecer contexto sobre decisões de design e documentar funções e classes. Comentários bem colocados podem ajudar outros desenvolvedores a entender rapidamente o que seu código faz.
 - **Organização Estrutural:**

Divida seu código em módulos ou arquivos separados, cada um com uma responsabilidade clara. Isso ajuda a manter o código organizado e facilita a localização de partes específicas do código quando necessário.
 - **Evitar Duplicação de Código:**

Procure por padrões repetitivos em seu código e busque maneiras de reutilizar código através de funções, classes ou componentes. Isso reduz a duplicação e facilita futuras alterações e manutenção.
 - **Manter o Código Simples:**

Evite complexidade desnecessária. Prefira soluções simples e diretas sempre que possível. Isso torna o código mais fácil de entender, depurar e manter no longo prazo.
 - **Testes Unitários:**

Escreva testes unitários para suas funções e componentes. Isso ajuda a garantir que seu código funcione conforme esperado e permite fazer alterações com confiança, sabendo que os testes irão identificar regressões.

Seminário Materialize

- Exemplos de Boas Práticas Aplicadas em um Projeto:

Exemplo - Projeto: Calculadora Simples

```
1 <div class="container">
2   <h4>Calculadora Simples</h4>
3   <div class="row">
4     <div class="col s12 m6">
5       <div class="input-field">
6         <input id="numero1" type="number" class="validate">
7         <label for="numero1">Número 1</label>
8       </div>
9       <div class="input-field">
10        <input id="numero2" type="number" class="validate">
11        <label for="numero2">Número 2</label>
12      </div>
13      <button class="btn waves-effect waves-light" onclick="calcular()">
14        Calcular
15      </button>
16    </div>
17  </div>
18  <div id="resultado" class="row hide">
19    <div class="col s12">
20      <h5>Resultado:</h5>
21      <p id="valorResultado"></p>
22    </div>
23  </div>
24 </div>
```

```
1 <!-- Incluindo o arquivo JavaScript -->
2 <script>
3   // Função para calcular a soma dos dois números
4   function calcular() {
5     // Obtendo os valores dos números a partir dos campos de entrada
6     var numero1 = parseFloat(document.getElementById('numero1').value);
7     var numero2 = parseFloat(document.getElementById('numero2').value);
8     // Calculando a soma dos números
9     var resultado = numero1 + numero2;
10    // Atualizando o elemento de resultado com o valor calculado
11    document.getElementById('valorResultado').textContent = resultado;
12    // Exibindo o resultado ao usuário removendo a classe 'hide'
13    document.getElementById('resultado').classList.remove('hide');
14  }
15 </script>
16
17 <!-- Incluindo o arquivo JavaScript do Materialize via CDN -->
18 <script src="https://cdn.jsdelivr.net/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
```

Seminário Materialize

Este projeto de calculadora simples demonstra algumas boas práticas:

- **Nomenclatura Descritiva:** Os IDs dos elementos HTML são descritivos, facilitando a identificação e manipulação no JavaScript.
- **Organização Estrutural:** O código HTML está dividido em seções claras, facilitando a compreensão e manutenção.
- **Evitar Duplicação de Código:** A função `calcular()` encapsula a lógica de cálculo, evitando repetição de código.

Seminário Materialize

Exercício Final – Seminário Materialize

Devido ao tamanho do código, seria inviável inseri-lo diretamente a este arquivo.

Links:

- [Repositório Github](#)
- [Website Live](#)

- **Visão geral:**

A documentação a seguir descreve a estrutura, funcionalidades e aspectos técnicos do software desenvolvido para o Seminário Materialize. Este software foi criado para apresentar um projeto fictício utilizando o framework materialize para o desenvolvimento de um site responsivo e dinâmico.

- **Estrutura do Código Fonte:**

O software é composto por várias páginas HTML, cada uma representando uma seção diferente do site. Cada página possui sua estrutura específica e funcionalidades próprias.

A seguir, são apresentadas as principais páginas do site:

- **Página "Home":**

A página inicial do site, que apresenta uma breve introdução ao projeto e destaca seus principais recursos.

- Estrutura: A página contém uma seção de introdução, seguida por cartões destacando os principais recursos do projeto.
- Funcionalidades Interativas: Os cartões têm efeitos visuais ao passar o mouse sobre eles para proporcionar uma experiência interativa ao usuário.
- Essa página diferente das demais como teste fizemos um carrossel próprio para ser adaptado à quando o usuário abre a página no celular.

Seminário Materialize

- **Página "Sobre o Projeto":**

Esta página fornece uma visão geral do projeto, destacando seus objetivos, funcionalidades e equipe de desenvolvimento.

- Estrutura: A página inclui um carrossel de imagens no topo, seguida por cartões que descrevem aspectos importantes do projeto, como documentação, contato e links úteis.
- Formulário de Contato: Os visitantes podem utilizar um formulário de contato para enviar mensagens diretamente à equipe de desenvolvimento.

- **Página "Contato":**

Esta página apresenta informações de contato da equipe de desenvolvimento e fornece um formulário para os visitantes entrarem em contato.

- Estrutura: Além das informações de contato, a página contém um formulário simples para inserir nome, sobrenome, Email e mensagem.

- **Página "Equipe":**

A página da equipe destaca os membros responsáveis pelo desenvolvimento do projeto. Estrutura: Além das informações de contato, a página contém um formulário simples para inserir nome, sobrenome, Email e mensagem.

- Estrutura: Apresenta cards individuais para cada membro da equipe, com uma foto, nome e links para seus perfis online.

- **Estilos Personalizados:**

O design do site é melhorado com estilos personalizados que garantem uma apresentação visualmente atraente e uma experiência de usuário consistente.

- Cartões Estilizados: Os cartões são estilizados com sombras sutis e efeitos de escala ao passar o mouse para uma aparência moderna e interativa.
- Carrossel de Imagens Atraente: O carrossel de imagens é estilizado para ser centralizado e exibir imagens de forma dinâmica.

Seminário Materialize

- **Funcionalidades Interativas:**

O software incorpora diversas funcionalidades interativas para melhorar a experiência do usuário e fornecer uma navegação fluida pelo site.

- **Navegação Fácil:** Links de navegação são disponibilizados na barra superior e na barra lateral para uma transição suave entre as diferentes lacunas do site.
- **Carrossel Automático:** O carrossel de imagens na página "Sobre o Projeto" avança automaticamente, destacando os aspectos importantes do projeto.
- **Formulário de Contato:** O formulário na página "Contato" permite que os visitantes enviem mensagens diretamente à equipe de desenvolvimento.

- **Conclusão:**

O software desenvolvido para o Seminário Materialize é uma representação prática do uso do framework materialize para criar um site moderno e responsivo. Com uma estrutura organizada, estilos personalizados e funcionalidades interativas, o site proporciona uma experiência de usuário atraente e funcional. Este documento visa fornecer uma visão abrangente do software e suas características principais.