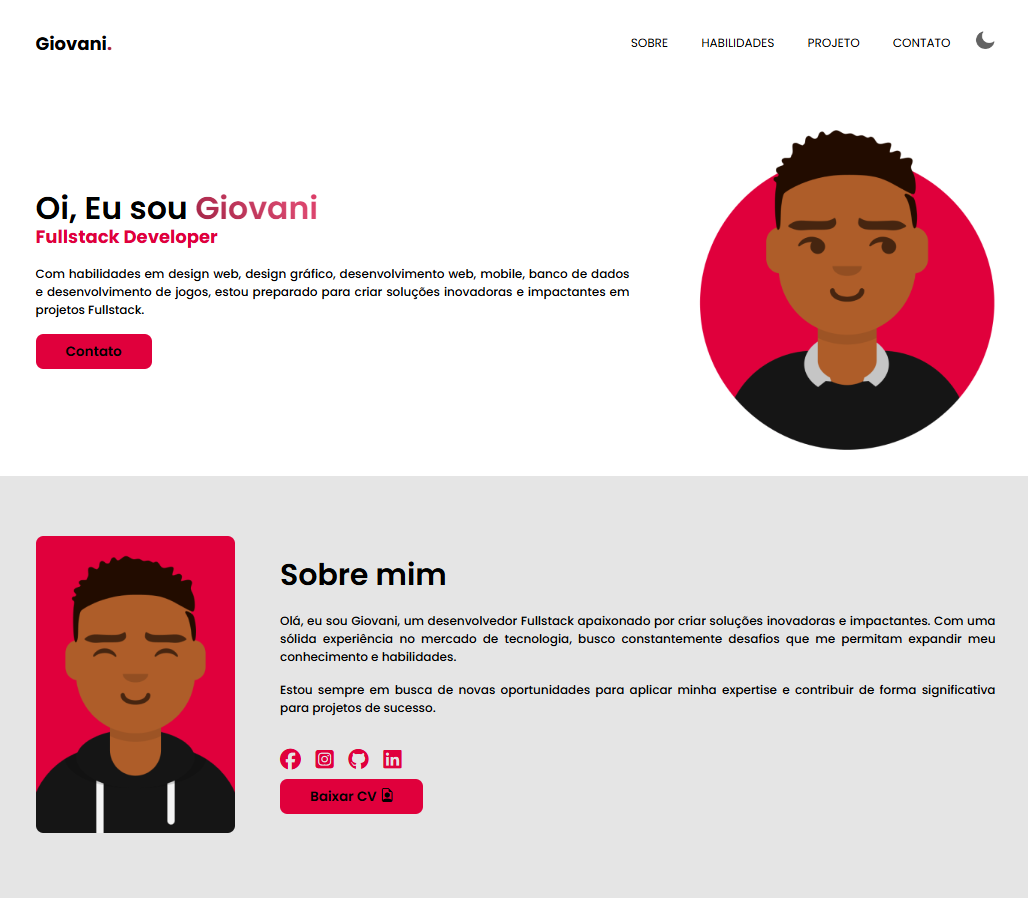
Relátorio técnico

Trabalho Final

Programaçao Web



By: Giovani dos Santos

Formador: Willian Pires

19/06/2024

#### Índice

[Introdução 3](#_Toc7385)

[Metodologia 3](#_Toc7385)

[Estrutura geral do Projeto 3](#_Toc7385)

[Comportamentos Dinâmicos com JavaScript: 4](#_Toc18429)

[Estrutura Detalhada do Projeto 7](#_Toc12135)

[Estrutura do Cabeçalho 7](#_Toc9308)

[Estrutura da Seção de Introdução 9](#_Toc6668)

[Estrutura da Seção "Sobre Mim" 11](#_Toc14435)

[Estrutura da Seção "Habilidades" 13](#_Toc22550)

[Estrutura da Tabela de Habilidades 14](#_Toc23493)

[Seção de Portfólio 14](#_Toc28561)

[Seção de Formulário de Contato 15](#_Toc8572)

[Análise Detalhada do CSS 16](#_Toc15445)

[Estilo do Cabeçalho 20](#_Toc18136)

[Estilo do Menu Mobile 21](#_Toc17307)

[Estilo do Topo do Site 24](#_Toc12323)

[Estilo do Sobre 24](#_Toc23302)

[Estilo dos Serviços 25](#_Toc32281)

[Estilo da Tabela 26](#_Toc2586)

[Estilo do Portfólio 28](#_Toc9502)

[Estilo do Formulário de Contato 30](#_Toc31465)

[Estilo do Footer 32](#_Toc6452)

[Dark theme 33](#_Toc25284)

[Estrutura das Media Queries 33](#_Toc20079)

[Conclusão 3](#_Toc20079)4

#### Introdução:

Este trabalho final tem como objetivo demonstrar os conhecimentos adquiridos ao longo da disciplina de Desenvolvimento Web, por meio da criação de um portfólio pessoal. Utilizando as tecnologias HTML, CSS e JavaScript, o portfólio será desenvolvido para exibir minhas habilidades, experiências e projetos de maneira organizada, acessível e interativa.

O portfólio incluirá diferentes níveis de cabeçalhos, parágrafos, listas, links, multimídia, elementos semânticos, tabelas e formulários, de acordo com os tópicos estudados durante a disciplina. Além disso, será utilizado CSS para estilizar o site, aplicando conceitos de Box Model e Flexbox para garantir um layout responsivo e moderno. JavaScript será empregado para adicionar funcionalidades interativas, como validação de formulários e animações.

**Metodologia:**

Tecnologias Utilizadas

**Linguagens de Programação:**

* **HTML/CSS:** Utilizados para estruturar e estilizar as páginas web, garantindo uma apresentação visual atraente e responsiva.
* **JavaScript:** Empregado para programação do lado do cliente, adicionando interatividade e dinamismo às funcionalidades da aplicação.

**Ferramentas de Desenvolvimento:**

* **Visual Studio Code:** Utilizado como ambiente de desenvolvimento integrado (IDE), oferecendo suporte para edição de código, depuração e controle de versão.

**Outras ferramentas:**

* **Font Awesome:** Utilizado para importar os icons
* **Figma:** Utilizado para encontrar as imagens presentes no site

#### Planejamento e Organização

Como o projeto foi desenvolvido individualmente, o planejamento foi estruturado em um cronograma pessoal, estabelecendo metas diárias para as diferentes fases do desenvolvimento. A organização do trabalho incluiu a divisão do tempo entre pesquisa, desenvolvimento de código e testes, garantindo uma progressão consistente e eficiente.

Esta metodologia permitiu um desenvolvimento organizado e eficaz do projeto, resultando em um produto final que atende aos requisitos estabelecidos e reflete as melhores práticas de desenvolvimento web.

### Estrutura geral do Projeto

O projeto é estruturado de maneira clara e organizada, seguindo as melhores práticas de desenvolvimento web. A estrutura é dividida em diferentes seções que compõem o layout geral do portfolio:

**Estrutura Geral do HTML:**

O HTML do projeto está organizado em várias seções principais, cada uma dedicada a um aspecto específico do portfolio. A estrutura básica inclui:

1. <header> **e** <nav>**:** Responsáveis pela navegação do site, com um menu principal contendo links para diferentes seções e uma logo no canto superior esquerdo que serve como link para a página inicial.
2. <section class="topo-do-site">**:** Esta seção é o topo da página, contendo um título principal, um texto descritivo, um botão contato e uma imagem ilustrativa.
3. <section class="sobre">**:** Uma seção dedicada a informações sobre mim, com uma imagem e um texto explicativo sobre sua experiência e qualificações.
4. <section class="servicos">**:** Esta seção apresenta os serviços oferecidos, cada um organizado em caixas destacadas com ícones e textos descritivos.
5. <section id="portfolio">**:** Uma seção que exibe uma coleção de cards de portfólio, cada um mostrando uma imagem representativa de um projeto, acompanhada de informações adicionais ao passar o mouse sobre elas.
6. <section class="contactForm">**:** A seção que contém um formulário de contato para que os visitantes possam enviar mensagens diretamente para mim.
7. <footer>**:** A seção final do site, contendo informações de contato, links para redes sociais, e outros links úteis.

**Estilo e Layout com CSS:**

**Estilo Geral (**\* {}**,** body {}**,** .container {}**, etc.):** Define estilos básicos para todos os elementos, como margens, preenchimentos, caixas de modelo, família de fontes e efeitos de transição.

**Estilo do Cabeçalho (**header {}**,** nav {}**,** .logo {}**, etc.):** Define o posicionamento, estilo e comportamento do cabeçalho e da navegação, incluindo o menu mobile e sua interação.

**Estilo do Topo do Site (**section.topo-do-site {}**,** .topo-do-site h1 {}**, etc.):** Define o layout e estilo do topo da página, incluindo o título principal, imagem flutuante, e botões de chamada para ação.

**Estilo da Seção Sobre (**section.sobre {}**,** .sobre .txt-sobre {}**, etc.):** Define o estilo da seção "Sobre", incluindo o layout da imagem, texto descritivo e ícones de redes sociais.

**Estilo dos Serviços (**section.servicos {}**,** .servicos .servicos-box {}**, etc.):** Define o layout e estilo das caixas de serviços, incluindo ícones, títulos e descrições.

**Estilo do Portfolio (**#portfolio {}**,** .portfolio\_card {}**,** .portfolio\_card\_informacao {}**, etc.):** Define o layout dos cards do portfólio, incluindo imagens representativas, informações adicionais ao passar o mouse e botões de interação.

**Estilo do Formulário de Contato (**.contactForm {}**,** .input-box {}**,** .btn-enviar {}**, etc.):** Define o layout e estilo do formulário de contato, incluindo caixas de entrada de texto, etiquetas flutuantes e botão de envio.

**Estilo do Rodapé (**footer {}**,** .container {}**,** .col {}**, etc.):** Define o layout e estilo do rodapé, incluindo informações de contato, links para redes sociais e outros links úteis.

### Comportamentos Dinâmicos com JavaScript:

O JavaScript desempenha um papel crucial no site, proporcionando funcionalidades interativas e dinâmicas que melhoram significativamente a experiência do usuário. Aqui estão os principais comportamentos dinâmicos implementados:

### Código 1: Alternar entre Tema Claro e Escuro

ar icon = document.getElementById("icon")

icon.onclick = function(){

    document.body.classList.toggle("dark-theme")

    if(document.body.classList.contains("dark-theme")){

        icon.src ="dark theme icon/sun.png"

    }else{

        icon.src ="dark theme icon/moon.png"

    }

}

**Estrutura:**

**Seleção de Elemento:** O código começa selecionando um elemento com o ID "icon".

**Evento de Clique:** Um evento onclick é adicionado ao elemento selecionado.

**Alternância de Classe:** Quando o ícone é clicado, a classe dark-theme é alternada no elemento body, mudando o tema entre claro e escuro.

**Atualização de Imagem:** Dependendo da presença da classe dark-theme, a imagem do ícone (sun.png ou moon.png) é atualizada para refletir o tema atual.

### Código 2: Menu Mobile Responsivo

let btnMenu = document.getElementById('btn-menu')

let menu = document.getElementById('menu-mobile')

let overlay = document.getElementById('overlay-menu')

btnMenu.addEventListener('click', ()=>{

    menu.classList.add('abrir-menu')

})

menu.addEventListener('click', ()=>{

    menu.classList.remove('abrir-menu')

})

overlay.addEventListener('click', ()=>{

    menu.classList.remove('abrir-menu')

})

**Estrutura:**

**Seleção de Elementos:** Os elementos com os IDs btn-menu, menu-mobile e overlay-menu são selecionados.

**Evento de Clique no Botão Menu:** Um evento click é adicionado ao botão de menu (btn-menu), que adiciona a classe abrir-menu ao elemento menu-mobile, exibindo o menu.

**Fechar Menu:** Dois eventos click são adicionados: um no próprio elemento menu-mobile e outro no elemento overlay-menu (provavelmente um overlay), ambos removendo a classe abrir-menu do menu-mobile, fechando o menu ao clicar fora dele ou no overlay.

### Código 3: Botão de Voltar ao Topo

let mybutton = document.getElementById("myBtn");

      window.onscroll = function() {scrollFunction()};

      function scrollFunction() {

        if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {

          mybutton.style.display = "block";

        } else {

          mybutton.style.display = "none";

        }

      }

      function topFunction() {

        document.body.scrollTop = 0;

        document.documentElement.scrollTop = 0;

      }

**Estrutura:**

**Seleção de Elemento:** O botão com o ID myBtn é selecionado.

**Evento de Rolagem da Janela:** Um evento onscroll é adicionado à janela, chamando a função scrollFunction() sempre que ocorre uma rolagem.

**Exibir Botão:** scrollFunction() verifica a posição da rolagem (scrollTop). Se a posição for maior do que 20 pixels, o botão é exibido (display = "block"); caso contrário, é ocultado (display = "none").

**Função** topFunction()**:** Quando o botão é clicado (onclick), a função topFunction() é chamada, fazendo a página rolar suavemente para o topo (scrollTop = 0).

### Estrutura Detalhada do Projeto

### Estrutura do Cabeçalho (<header class="header">)

 <header *class*="header">

        <div *class*="container">

            <div *class*="hero">

                <nav>

                    <h1 *class*="logo">Giovani<span>.</span></h1>

                    <ul *class*="menu-desktop">

                        <li><a *href*="#sobre">SOBRE</a></li>

                        <li><a *href*="#servicos">HABILIDADES</a></li>

                        <li><a *href*="#portfolio">PROJETO</a></li>

                        <li><a *href*="#contato">CONTATO</a></li>

                    </ul>

                    <div *class*="lateral">

                        <div>

                            <img *src*="dark theme icon/moon.png" *alt*="" *id*="icon">

                        </div>

                        <div *class*="btn-abrir-menu" *id*="btn-menu">

                            <i *class*="fa-solid fa-bars"></i>

                        </div>

                    </div>

                </nav>

            </div>

        </div>

    </header>

#### Descrição Detalhada:

**<header class="header">**

Define o cabeçalho principal do site, utilizando a classe header para estilização.

**<div class="container">**

Envolve o conteúdo do cabeçalho em uma div com a classe container, garantindo um controle melhor de largura e alinhamento do conteúdo.

**<div class="hero">**

Uma div com a classe hero, não está diretamente preenchida com conteúdo adicional além da navegação.

**<nav>**

Contém a estrutura de navegação do cabeçalho.

**<h1 class="logo">Giovani<span>.</span></h1>**

O título principal do site, com a classe logo, exibindo o nome "Giovani" com um ponto estilizado ao lado.

**<ul class="menu-desktop">**

Lista não ordenada (ul) que contém os links principais da navegação em desktop.

Cada item da lista (li) contém um link (<a>) para diferentes seções do site, utilizando href para âncoras nas seções correspondentes.

**Ícone de Modo Noturno e Ícone de Menu Mobile:**

**<img src="dark theme icon/moon.png" alt="" id="icon">: Ícone que representa o modo noturno do site.**

<div class="btn-abrir-menu" id="btn-menu">: Ícone de menu mobile, utilizando uma classe e um ID específicos para controle JavaScript.

### Menu Mobile (<div class="menu-mobile" id="menu-mobile">)

<div *class*="menu-mobile" *id*="menu-mobile">

        <div *class*="btn-fechar">

            <i *class*="fa-solid fa-xmark"></i>

        </div>

        <nav>

            <ul>

                <li><a *href*="#inicio">INÍCIO</a></li>

                <li><a *href*="#sobre">SOBRE</a></li>

                <li><a *href*="#servicos">HABILIDADES</a></li>

                <li><a *href*="#portfolio">PROJETOS</a></li>

                <li><a *href*="#contato">CONTATO</a></li>

            </ul>

        </nav>

    </div>

#### Descrição Detalhada:

**<div class="menu-mobile" id="menu-mobile">**

Define a estrutura do menu mobile, utilizando a classe menu-mobile para estilização e controle JavaScript.

**Ícone de Fechar Menu:**

<div class="btn-fechar">: Ícone utilizado para fechar o menu mobile.

<i class="fa-solid fa-xmark"></i>: Ícone X (marca de fechar) da biblioteca Font Awesome, estilizado por classes específicas.

<nav> **e** <ul>

Contém a lista de links para navegação em formato de lista não ordenada (ul), com âncoras (<a>) para diferentes seções do site.

### Overlay para Menu Mobile (<div class="overlay-menu" id="overlay-menu"></div>)

<div *class*="overlay-menu" *id*="overlay-menu"></div>

#### Descrição Detalhada:

Um elemento div usado como overlay para o menu mobile, cobrindo o restante do conteúdo do site quando o menu mobile está aberto. Este elemento é utilizado para fechar o menu ao clicar fora dele.

### Estrutura da Seção de Introdução (<section id="inicio" class="topo-do-site">)

<section *id*="inicio" *class*="topo-do-site">

            <div *class*="container">

                <div *class*="flex">

                    <div *class*="txt-top-site">

                        <h1>Oi, Eu sou <span>Giovani</span></h1>

                        <h2 *class*="msg">Fullstack Developer</h2>

<p>Com habilidades em design web, design gráfico, desenvolvimento web, mobile, banco de dados e desenvolvimento de jogos, estou preparado para criar soluções inovadoras e impactantes em projetos Fullstack.</p>

                        <div *class*="btn-contato">

                            <a *href*="#contato">

                                <button>Contato</button>

                            </a>

                        </div>

                    </div><!--txt-top-site-->

                    <div *class*="img-top-site">

                        <img *src*="imagens/me.png" *alt*="">

                    </div><!-- img-top-site -->

                </div><!-- flex -->

            </div><!--container-->

        </section><!--Topo-do-site-->

#### Descrição Detalhada:

**<section id="inicio" class="topo-do-site">**

Define uma seção principal do site com o ID "inicio" e a classe "topo-do-site". Este ID é utilizado para navegação por âncoras, permitindo que links direcionem diretamente a esta seção.

**<div class="container">**

Envolve o conteúdo da seção, garantindo um layout consistente e controlando a largura do conteúdo dentro da seção.

**<div class="flex">**

Utiliza a classe "flex" para aplicar um layout flexbox, organizando os elementos filhos em uma linha ou coluna, conforme especificado nas regras CSS.

**<div class="txt-top-site">**

Esta div contém o texto principal da seção de introdução.

**<h1>Oi, Eu sou <span>Giovani</span></h1>**

O título principal da seção, apresentando o nome "Giovani" em um span para aplicar estilizações específicas se necessário.

**<h2 class="msg">Fullstack Developer</h2>**

Subtítulo que descreve a profissão, com a classe "msg" para estilização.

**<div class="btn-contato">**

Contém um botão que, ao ser clicado, direciona o usuário para a seção de contato.

O botão é envolvido em um link (<a href="#contato">) para criar uma navegação de âncora suave.

<div class="img-top-site">

Contém a imagem de perfil do desenvolvedor.

A imagem (<img src="imagens/me.png" alt="">) é carregada a partir do diretório "imagens" e exibida na seção de introdução.

**Estrutura Flexbox:**

A classe "flex" aplicada à div pai garante que os elementos .txt-top-site e .img-top-site sejam dispostos lado a lado em um layout flexível, que pode se ajustar a diferentes tamanhos de tela.

### Estrutura da Seção "Sobre Mim" (<section id="sobre" class="sobre">)

<section *id*="sobre" *class*="sobre">

            <div *class*="container">

                <div *class*="flex">

                    <div *class*="img-sobre">

                        <img *src*="imagens/image 6.png" *alt*="">

                    </div><!--img-sobre-->

                    <div *class*="txt-sobre">

                        <h2 *class*="sobre">Sobre mim</h2>

                        <P>Olá, eu sou Giovani, um desenvolvedor Fullstack apaixonado por criar soluções inovadoras e impactantes…</P>

<P>Estou sempre em busca de novas oportunidades para aplicar minha expertise e contribuir…</P>

                        <div *class*="sobre-btn">

                            <div *class*="social">

                                <a *href*="https://pt-br.facebook.com/login.php/"><i *class*="fa-brands fa-facebook"></i></a>

                                <a *href*="https://www.instagram.com/"><i *class*="fa-brands fa-square-instagram"></i></a>

                                <a *href*="https://github.com/GiovaniSantos9"><i *class*="fa-brands fa-github"></i></a>

                                <a *href*="https://www.linkedin.com/"><i *class*="fa-brands fa-linkedin"></i></a>

                            </div>

                            <div *class*="btn-cv2">

                                <a *href*="#">

                                    <button *class*="btn-cv">Baixar CV <i *class*="bi bi-file-earmark-person"></i></button>

                                </a>

                            </div>

                        </div>

                    </div><!--txt-sobre-->

                </div><!--flex-->

            </div><!--container-->

        </section><!--sobre-->

#### Descrição Detalhada:

**<section id="sobre" class="sobre">**

Esta seção é identificada com o ID "sobre" e a classe "sobre", permitindo que links de navegação e estilos específicos sejam aplicados a ela.

**<div class="container">**

Um contêiner que centraliza e restringe a largura do conteúdo para manter uma estrutura organizada e responsiva.

**<div class="flex">**

A aplicação de flexbox para organizar os elementos filhos horizontalmente ou verticalmente, conforme necessário.

**<div class="img-sobre">**

Contém a imagem ilustrativa sobre o autor da página.

A imagem (<img src="imagens/image 6.png" alt="">) proporciona um toque pessoal e visual à seção "Sobre".

**<div class="txt-sobre">**

Contém todo o texto explicativo e botões interativos relacionados ao autor.

**<h2 class="sobre">Sobre mim</h2>**

Um título destacado da seção que descreve seu conteúdo.

**<div class="sobre-btn">**

Contém botões de ação que incentivam a interação com a seção.

**<div class="social">**

Inclui ícones de redes sociais que linkam para os perfis do autor no Facebook, Instagram, GitHub e LinkedIn.

Os ícones são carregados a partir do FontAwesome, adicionando uma representação visual fácil de reconhecer.

**<div class="btn-cv2">**

Contém um botão que permite aos usuários baixar o CV do autor.

A estrutura com link (<a href="#">) e botão (<button class="btn-cv">) proporciona uma chamada clara para ação.

### Estrutura da Seção "Habilidades" (<section id="servicos" class="servicos">)

<section *id*="servicos" *class*="servicos">

            <div *class*="container">

                <h2 *class*="titulo">Minhas<span> Habilidades</span></h2>

                <p *class*="subtitulo">Principais Habilidades</p>

                <div *class*="flex">

                    <div *class*="servicos-box">

                        <i *class*="fa-regular fa-object-group"></i>

                        <h3>Web Design</h3>

                        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus optio unde omnis eius! </p>

                        <a *href*="#">Ler Mais</a>

                    </div><!--servicos-box-->

                </div><!--flex-->

            </div><!--container-->

        </section><!--servicos-->

#### Descrição Detalhada:

**<section id="servicos" class="servicos">**

Esta seção é identificada com o ID "servicos" e a classe "servicos", tornando-a facilmente referenciável e estilizável.

<p class="subtitulo">Principais Habilidades</p>

Um subtítulo que introduz a lista de habilidades.

<div class="servicos-box">

Cada caixa de habilidades (servicos-box) contém um ícone, título, descrição e um link para mais informações.

Existem seis dessas caixas, cada uma representando uma habilidade diferente: Web Design, Graphic Design, Web Developer, Mobile, Data Base, e Games Developer.

Os ícones são carregados a partir do FontAwesome, proporcionando uma identificação visual rápida das habilidades.

### Estrutura da Tabela de Habilidades (<div class="table-container">)

#### Descrição Detalhada:

**<table class="portfolio-table">**

A tabela principal que contém as informações sobre as habilidades do desenvolvedor.

A classe portfolio-table é utilizada para aplicar estilos específicos à tabela.

**Linha (**<tr>**):**

Contém os cabeçalhos das colunas.

### Seção de Portfólio

A seção de portfólio é uma parte crucial do site, onde são exibidos os projetos desenvolvidos pelo autor. Esta seção destaca os trabalhos realizados e proporciona aos visitantes uma visão clara das habilidades e experiências do desenvolvedor. A seguir, uma descrição detalhada da estrutura e elementos desta seção:

#### Estrutura HTML da Seção de Portfólio

### Descrição Detalhada:

**Contêiner de Todos os Cards de Portfólio (**<div class="portfolio\_all\_card">**)**

Um contêiner flexível que engloba todos os cards individuais de projetos.

**Cards de Projeto**

Cada card é representado por um div com a classe portfolio\_card:

**Imagem do Projeto (**<img src="...">**)**: Mostra uma imagem representativa do projeto.

**Informações do Card (**<div class="portfolio\_card\_informacao">**)**:

**Título do Projeto (**<h2>**)**: O nome do projeto.

**Descrição do Projeto (**<p>**)**: Uma breve descrição do projeto.

**Link "Ver Mais" (**<a href="#" class="button">Ver m ais</a>**)**: Um botão que leva a mais informações sobre o projeto.

**Botão "Ver Mais" (**<div class="more\_button">**)**

Um botão adicional fora dos cards que permite ao usuário acessar mais projetos, sugerindo a existência de mais conteúdo além do exibido.

### Seção de Formulário de Contato

A seção de formulário de contato é essencial para permitir que visitantes entrem em contato diretamente com o desenvolvedor. Esta seção é projetada para ser clara e intuitiva, facilitando a comunicação. A seguir, uma descrição detalhada da estrutura e dos elementos desta seção:

#### Estrutura HTML da Seção de Formulário de Contato

**<section id="contato" class="formulario">**

Delimita a seção de formulário de contato, proporcionando um ponto de ancoragem para navegação interna.

**<div class="contactForm">**

Um contêiner que mantém o formulário centralizado e estilizado dentro do layout geral do site.

**<form>**

Elemento HTML que encapsula todos os campos do formulário.

**Campos de Entrada (**input-box**)**

Cada campo de entrada é envolvido por uma div com a classe input-box para estilização e organização.

**Campo Nome (**<input type="text" id="nome-box" autocomplete="off" required>**)**:

<input>: Campo de entrada de texto para o nome do usuário, com preenchimento obrigatório (required) e autocomplete desativado (autocomplete="off").

<label for="">Seu nome completo</label>: Rótulo descritivo para o campo.

**Campo E-mail (**<input type="text" id="email-box" autocomplete="off" required>**)**:

**Campo Assunto (**<input type="text" id="assunto-box" autocomplete="off" required>**)**:

**Campo Mensagem (**<textarea id="mensagem-box" required autocomplete="off"></textarea>**)**:

#### Estrutura HTML do Botão "Voltar ao Topo"

 <!-- Botão voltar -->

        <button *onclick*="topFunction()" *id*="myBtn" *title*="Go to top"><i *class*="fa-solid fa-arrow-up"></i></i></button>

        <script *src*="voltarBtn.js"></script>

Associa a função topFunction() ao evento de clique no botão, que será executada quando o botão for clicado.

**Atributo** id**:** "myBtn"

Atribui um identificador único ao botão, permitindo estilização e manipulação via CSS e JavaScript.

**Atributo** title**:** "Go to top"

Exibe um texto explicativo quando o usuário passa o cursor sobre o botão.

**Elemento:** <script> src="voltarBtn.js"></script>

Inclui o arquivo JavaScript externo voltarBtn.js, que contém o código necessário para controlar a exibição e funcionalidade do botão.

### Análise Detalhada do CSS

**Global Style (**\***)**:

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    font-family: "Poppins", sans-serif;

    scroll-behavior: smooth;

}

**Margin e Padding**: Remove todas as margens e preenchimentos padrão de todos os elementos.

**Box-sizing**: Define border-box, o que faz com que o padding e a borda sejam incluídos na largura e altura totais de um elemento.

**Font-family**: Define a fonte "Poppins" para todos os elementos.

**Scroll-behavior**: Habilita uma rolagem suave para ancoragens e outras ações de rolagem.

**Estilo do Corpo (**body**)**:

body{

    background-color: var(--primary-color);

    height: 100vh;

    transition: .5s

}

**Background-color**: Usa uma variável CSS --primary-color para definir a cor de fundo.

**Height**: Define a altura do corpo como 100% da altura da janela de visualização.

**Transition**: Aplica uma transição suave de 0,5 segundos para qualquer mudança no estilo do corpo.

**Container Geral (**.container**)**:

*.container*{

    max-width: 1280px;

    margin: 0 auto;

}

**Max-width**: Define a largura máxima do container como 1280px.

**Margin**: Centraliza o container horizontalmente.

**Flex Container (**.flex**)**:

**Display**: Define o elemento como um contêiner flexível, permitindo layouts flexíveis.

**Botão de Contato (**.btn-contato button**)**:

**Padding**: Define o preenchimento interno do botão.

**Font-size e Font-weight**: Define o tamanho e peso da fonte.

**Background-color**: Define a cor de fundo do botão.

**Border**: Remove a borda padrão e define a largura da borda.

**Border-radius**: Arredonda os cantos do botão.

**Cursor**: Altera o cursor para um ponteiro ao passar o mouse sobre o botão.

**Transition**: Aplica uma transição suave de 0,2 segundos para mudanças de estilo.

**Título (**h2.titulo**)**:

h2*.titulo*{

    color: var(--secondary-color);

    font-size: 38px;

    text-align: center;

    font-weight: 600;

}

**Color**: Define a cor do texto usando a variável CSS --secondary-color.

**Font-size**: Define o tamanho da fonte.

**Text-align**: Centraliza o texto.

**Font-weight**: Define o peso da fonte.

**Título Secundário (**h2.sobre**)**:

**Font-weight**: Define o peso da fonte para títulos de seção "Sobre".

**Parágrafos (**p**)**:

**Font-weight**: Define o peso da fonte para parágrafos.

Esta seção do CSS configura estilos globais e padrões de layout, garantindo uma aparência consistente e fluida em toda a página. A utilização de variáveis CSS (--primary-color e --secondary-color) facilita a manutenção e a atualização das cores do site.

#### Efeito no span

span{

    background: linear-gradient(45deg, #e0003c, #861332, #da486f, #e0003c);

    background-size: 400% 100%;

    animation: degrade 10s linear infinite alternate;

*-webkit-background-clip*: text;

*-webkit-text-fill-color*: transparent;

**Background**: Aplica um gradiente linear de 45 graus, com as cores especificadas (#e0003c, #861332, #da486f, #e0003c).

**Background-size**: Define o tamanho do plano de fundo como 400% da largura do elemento e 100% da altura. Isso permite que o gradiente se mova horizontalmente de uma borda à outra.

**Animation**: Aplica a animação chamada degrade por 10 segundos, linearmente e infinitamente, alternando entre as direções (avança e retrocede).

**Webkit-background-clip**: Define que o plano de fundo será aplicado somente ao texto.

**Webkit-text-fill-color**: Define que a cor de preenchimento do texto será transparente, permitindo que o plano de fundo gradiente apareça como a cor do texto.

**Animação** degrade:

@keyframes *degrade* {

    0%{

        background-position-x: 0%;

    }

    100%{

        background-position: 400%;

    }

}

**@keyframes**: Define uma animação chamada degrade.

**0%**: No início da animação, a posição do plano de fundo é 0% na direção X.

**100%**: No final da animação, a posição do plano de fundo é 400% na direção X, fazendo com que o gradiente se mova completamente de um lado para o outro.

#### Efeito Visual

**Gradiente**: O span apresenta um gradiente dinâmico que muda continuamente devido à animação.

**Texto Transparente**: O texto no span é transparente, mas como o plano de fundo é cortado pelo texto, o gradiente aparece como a cor do texto.

**Animação Suave**: A animação faz com que o gradiente se mova de um lado para o outro de forma suave e contínua, criando um efeito visual atrativo.

#### Efeito de Hover nos Botões

button*:hover*, form *.btn-enviar* input*:hover*{

    box-shadow: 0px 0px 5px #e0003c;

    transform: scale(1.05);

}

**Seletores**:

button:hover: Aplica os estilos especificados quando qualquer elemento <button> está em estado de hover (quando o usuário passa o mouse sobre ele).

form .btn-enviar input:hover: Aplica os estilos especificados quando qualquer elemento <input> dentro de um formulário e com a classe .btn-enviar está em estado de hover.

**Estilos Aplicados no Hover**:

**Box-shadow**:

box-shadow: 0px 0px 5px #e0003c;transform: scale(1.05);

0px 0px 5px #e0003c: Adiciona uma sombra ao redor do elemento com um desfoque de 5px e cor #e0003c.

Este efeito cria um destaque ao redor do botão, fazendo-o parecer mais elevado e pronunciado.

**Transform**:

scale(1.05): Aumenta o tamanho do elemento em 5%.

Este efeito de escala faz com que o botão pareça crescer ligeiramente quando o usuário passa o mouse sobre ele, adicionando um feedback visual interativo.

#### Efeito Visual

**Sombra**: Quando o usuário passa o mouse sobre um botão ou um input dentro de .btn-enviar, ele ganha uma sombra vermelha (#e0003c) com um desfoque de 5px, destacando-o visualmente.

**Aumento de Tamanho**: Simultaneamente, o elemento aumenta de tamanho em 5%, criando um efeito de zoom suave.

### Estilo do Cabeçalho

#### header

**Descrição**: Define o espaçamento interno do cabeçalho. Adiciona 40 pixels de padding na parte superior e inferior, e 4% nas laterais.

#### nav

**Descrição**: Utiliza Flexbox para alinhar os itens dentro do elemento nav. Os itens são alinhados verticalmente ao centro (align-items: center) e distribuídos com espaço igual entre eles (justify-content: space-between).

#### .logo

**Descrição**: Define a largura da logo, muda o cursor para pointer ao passar o mouse (indicando que é clicável) e define a cor do texto com uma variável CSS --secondary-color.

#### nav ul

**Descrição**: Adiciona uma margem direita de 10 pixels para o elemento ul dentro do nav.

#### nav ul li

**Descrição**: Define que os itens da lista sejam exibidos em linha (display: inline), remove os marcadores da lista (list-style: none) e adiciona uma margem de 10 pixels vertical e 20 pixels horizontal.

#### nav ul li a

**Descrição**: Remove a sublinhado dos links (text-decoration: none) e define a cor do texto com a variável --secondary-color.

#### nav ul li a:hover

**Descrição**: Altera a cor do texto para #e0003c ao passar o mouse, aumenta o tamanho do link em 5% (transform: scale(1.05)), e aplica uma transição suave de 0.2 segundos (transition: .2s).

### Estilo do Menu Mobile

#### .btn-abrir-menu i

**Descrição**: Define a cor dos ícones no botão de abrir menu para #e0003c e o tamanho da fonte para 30 pixels.

#### .btn-abrir-menu

**Descrição**: Oculta o botão de abrir menu por padrão (display: none) e adiciona uma margem inferior de 5 pixels.

#### .menu-mobile

*.menu-mobile*{

    background-color: var(--primary-color);

    height: 450px;

    position: fixed;

    top: 0;

    right: 0;

    z-index: 99999;

    width: 0%;

    overflow: hidden;

    transition: .4s;

    border-bottom-left-radius: 10px;

    border-top-left-radius: 10px;

}

**Descrição**: Define o estilo do menu mobile. A cor de fundo é definida pela variável --primary-color. O menu tem altura de 450 pixels, é posicionado fixo no canto superior direito (position: fixed; top: 0; right: 0), e tem um índice z alto para garantir que fique sobre outros elementos (z-index: 99999). A largura inicial é 0%, com transição suave de 0.4 segundos. Também tem cantos arredondados no lado esquerdo superior e inferior.

#### .menu-mobile.abrir-menu

*.menu-mobile.abrir-menu*{

    width: 40%;

}

**Descrição**: Define a largura do menu para 40% quando a classe abrir-menu é adicionada, o que faz com que o menu se expanda ao ser aberto.

#### .menu-mobile.abrir-menu ~ .overlay-menu

*.menu-mobile.abrir-menu* ~ *.overlay-menu*{

    display: block;

}

**Descrição**: Exibe o overlay de fundo quando o menu é aberto.

#### .menu-mobile nav ul li a

*.menu-mobile* nav ul li a{

    color: var(--secondary-color);

    font-size: 16px;

    font-weight: 500;

    padding: 5px 8%;

    display: block;

    text-align: center;

}

**Descrição**: Define o estilo dos links dentro do menu mobile. A cor do texto é definida pela variável --secondary-color, o tamanho da fonte é 16 pixels, e o peso da fonte é 500. Adiciona padding de 5 pixels vertical e 8% horizontal, define o display como block, e centraliza o texto.

#### .menu-mobile nav ul li a;hover

*.menu-mobile* nav ul li a*:hover*{

    background-color: #e0003c;

    color: var(--secondary-color);

    border-radius: 10px;

}

**Descrição**: Altera a cor de fundo do link para #e0003c, a cor do texto para --secondary-color, e adiciona bordas arredondadas de 10 pixels ao passar o mouse.

#### .overlay-menu

*.overlay-menu*{

    background-color: #000000a4;

    width: 100%;

    height: 100%;

    position: fixed;

    top: 0;

    left: 0;

    z-index: 88888;

    display: none;

}

**Descrição**: Define o estilo do overlay de fundo. A cor de fundo é preta com 64% de opacidade (#000000a4). A largura e altura são 100%, cobrindo toda a tela. A posição é fixa, com um índice z menor que o menu mobile (z-index: 88888). Por padrão, está oculto (display: none).

### Estilo do Topo do Site

#### section.topo-do-site .flex

section*.topo-do-site* *.flex*{

    align-items: center;

    justify-content: center;

    text-align: justify;

    gap: 90px;

}

**Descrição**: Utiliza Flexbox para centralizar verticalmente (align-items: center) e horizontalmente (justify-content: center) os itens dentro da seção de topo do site. Define o texto para alinhamento justificado (text-align: justify) e adiciona um espaço de 90 pixels entre os itens (gap: 90px).

.topo-do-site .img-top-site img

*.topo-do-site* *.img-top-site* img{

    position: relative;

    animation: flutuar 2s ease-in-out infinite alternate;

}

**Descrição**: Define a posição relativa para a imagem dentro da classe img-top-site e aplica uma animação contínua e alternada chamada flutuar, com duração de 2 segundos e suavização ease-in-out.

#### @keyframes flutuar

@keyframes *flutuar*{

    0%{

        top: 0;

    }

    100%{

        top: 30px;

    }

}

**Descrição**: Define a animação flutuar, onde a posição top da imagem começa em 0 pixels e vai até 30 pixels, criando um efeito de flutuação vertical.

### Estilo do Sobre

#### .sobre-btn .social a i

*.sobre-btn* *.social* a i{

    font-size: 28px;

    color:#e0003c;

    margin-top:2rem;

    margin-right: 15px;

    transition: 0.3s ease;

}

**Descrição:** Define o estilo para os ícones de redes sociais dentro da classe .social na seção "Sobre", com tamanho de fonte de 28 pixels, cor #e0003c, margem superior de 2 rem e margem direita de 15 pixels, além de uma transição suave de 0.3 segundos..

#### .social a i:hover

*.social* a i*:hover*{

    transform: scale(1.05);

    filter:grayscale(25);

}

**Descrição:** Define o efeito de escala (aumento em 5%) e aplicação de filtro de escala de cinza em 25% quando o mouse estiver sobre os ícones de redes sociais na seção "Sobre".

### Estilo dos Serviços

#### .servicos .servicos-box

*.servicos* *.servicos-box*{

    color:var(--secondary-color);

    padding: 30px 30px;

    border-radius: 20px ;

    margin-top: 7px;

    transition: .2s;

    background-color: #2424241e;

    text-align: center;

    max-width: 350px;

}

**Descrição:** Define o estilo para os blocos de serviços dentro da seção "Serviços". Aplica uma cor de texto usando a variável --secondary-color, padding interno de 30 pixels na vertical e 30 pixels na horizontal, bordas arredondadas de 20 pixels, uma margem superior de 7 pixels, transição suave de 0.2 segundos, fundo semi-transparente com a cor #2424241e, alinhamento centralizado do texto e largura máxima de 350 pixels.

#### .servicos-box a

*.servicos-box* a {

    color:var(--secondary-color);

    font-weight: 600;

    font-size: 20;

    margin-top: 15px;

    cursor: pointer;

    text-decoration: none;

    padding: 1px 7px;

}

}

**Descrição:** Define o estilo para os links dentro dos blocos de serviços na seção "Serviços". Aplica uma cor de texto usando a variável --secondary-color, peso da fonte de 600, tamanho de fonte de 20 pixels, margem superior de 15 pixels, cursor em forma de ponteiro, sem decoração de texto (text-decoration: none) e um pequeno padding interno de 1 pixel na vertical e 7 pixels na horizontal.

### Estilo da Tabela

#### .table-container

*.table-container* {

    margin-bottom: 50px;

    overflow-x: auto;

    margin-top: 20px;

    display: flex;

    justify-content: center;

    align-items: center;

}

**Descrição:** Define o estilo para o contêiner da tabela. Adiciona uma margem inferior de 50 pixels e uma margem superior de 20 pixels. Define um overflow-x para permitir rolagem horizontal caso o conteúdo da tabela seja maior que a largura disponível. Utiliza Flexbox para centralizar tanto horizontalmente (justify-content: center) quanto verticalmente (align-items: center) o conteúdo da tabela.

#### .portfolio-table

*.portfolio-table* {

    max-width: 600px;

    width: 70%;

    border-collapse: collapse;

}

**Descrição:** Define o estilo para a tabela com a classe .portfolio-table. Define uma largura máxima de 600 pixels e uma largura de 70% do contêiner pai. border-collapse: collapse; remove espaços entre as células da tabela, criando uma borda única contínua.

#### .portfolio-table th, .portfolio-table td

*.portfolio-table* th, *.portfolio-table* td {

    border: 1px solid #ddd;

    padding: 10px;

    text-align: center;

    word-break: break-all;

}

**Descrição:** Define o estilo para cabeçalhos (th) e células (td) da tabela .portfolio-table. Adiciona uma borda de 1 pixel sólida cinza (#ddd), padding interno de 10 pixels, alinha o texto ao centro (text-align: center) e permite que palavras longas sejam quebradas (word-break: break-all).

#### .portfolio-table th

*.portfolio-table* th {

    color: var(--secondary-color);

    background-color: #2424241e;

}

**Descrição:** Define o estilo para os cabeçalhos (th) da tabela .portfolio-table. Define a cor do texto usando a variável --secondary-color e define um fundo semi-transparente com a cor #2424241e.

#### .portfolio-table tr:nth-child(even)

*.portfolio-table* tr*:nth-child*(even) {

    background-color: transparent;

    color: var(--secondary-color);

}

**Descrição:** Define o estilo para as linhas pares da tabela .portfolio-table. Define um fundo transparente e define a cor do texto usando a variável --secondary-color.

### Estilo do Portfólio

#### .portfolio\_all\_card

*.portfolio\_all\_card*{

    display: flex;

    flex-wrap: wrap;

    align-items: center;

    justify-content: space-evenly;

}

**Descrição:** Define o estilo para o contêiner que envolve todos os cartões de portfólio. Utiliza Flexbox para exibir os cartões em uma linha flexível (flex-wrap: wrap). Alinha os itens verticalmente no centro (align-items: center) e distribui os itens com espaçamento igual entre eles (justify-content: space-evenly).

#### .portfolio\_card

*.portfolio\_card*{

    width:360px;

    height: auto;

    margin-top: 20px;

    position: relative;

**Descrição:** Define o estilo para cada cartão individual de portfólio. Define a largura como 360 pixels, altura automática (auto), adiciona margem superior de 20 pixels e posicionamento relativo (position: relative).

#### .portfolio\_card img

*.portfolio\_card* img{

    width: 100%;

    display: block;

    margin: auto;

    border-radius: 8px;

}

**Descrição:** Define o estilo para as imagens dentro dos cartões de portfólio. Define a largura como 100% para ocupar todo o espaço disponível, exibe a imagem como bloco (display: block), centraliza a imagem horizontalmente (margin: auto) e adiciona bordas arredondadas com raio de 8 pixels (border-radius: 8px).

#### .portfolio\_card\_informacao

*.portfolio\_card\_informacao*{

    position: absolute;

    width: 100%;

    height: 100%;

    top:0;

    background: rgb(0, 0, 0,0.6);

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    padding: 0px 70px;

    text-align: center;

    opacity: 0;

    transition: 0.6s;

}

**Descrição:** Define o estilo para a informação do cartão de portfólio que aparece ao passar o mouse. Define posicionamento absoluto (position: absolute) para ocupar todo o espaço do cartão pai. Define um fundo preto semi-transparente (background: rgb(0, 0, 0,0.6)), utiliza Flexbox para alinhar itens no centro vertical e horizontal (align-items: center; justify-content: center). Define padding interno de 70 pixels nos lados, alinha o texto ao centro (text-align: center), define opacidade inicial como 0 e adiciona uma transição suave de 0.6 segundos para efeito de fade-in.

#### .portfolio\_card\_informacao:hover

*.portfolio\_card\_informacao:hover*{

    opacity: 1;

    border-radius: 8px;

}

**Descrição:** Define o estilo quando o cartão de portfólio é hoverado (passagem do mouse). Aumenta a opacidade para 1 (tornando visível), e arredonda as bordas com raio de 8 pixels (border-radius: 8px).

#### .portfolio\_card\_informacao>\*

*.portfolio\_card\_informacao*>\*{

    transform: translateY(25px);

    transition: transform 0.6s;

}

**Descrição:** Define a transição para todos os elementos filhos diretos (> \*) dentro do elemento com a classe .portfolio\_card\_informacao. Define uma transformação que desloca os elementos verticalmente para baixo em 25 pixels (transform: translateY(25px)). Adiciona uma transição suave de 0.6 segundos para a transformação (transition: transform 0.6s), criando um efeito de movimento suave para esses elementos.

#### .portfolio\_card\_informacao:hover>\*

*.portfolio\_card\_informacao:hover*>\*{

    transform: translateY(0px);

}

**Descrição:** Define o comportamento dos elementos filhos diretos (> \*) dentro do elemento com a classe .portfolio\_card\_informacao quando o mouse passa sobre (:hover). Define uma transformação que restaura a posição original dos elementos, movendo-os verticalmente para cima em 0 pixels (transform: translateY(0px)), removendo o deslocamento anteriormente definido e retornando os elementos à sua posição inicial.

### Estilo do Formulário de Contato

#### .contactForm

*.contactForm*{

    max-width: 1080px;

    margin: 0 auto;

    justify-content: center;

    align-items: center;

    display: flex;

    padding: 50px 8%;

}

**Descrição:** Define o contêiner principal para o formulário de contato. Estabelece largura máxima de 1080 pixels, centraliza horizontalmente (margin: 0 auto), utiliza flex para alinhar e justificar conteúdos centralizados. Aplica um preenchimento de 50 pixels na parte superior e inferior e 8% nas laterais.

#### .contactForm .input-box

*.contactForm* *.input-box*{

    position: relative;

    width: 100%;

    margin-top: 10px;

}

**Descrição:** Estilo para a caixa de entrada do formulário dentro do .contactForm. Define posição relativa para que os elementos filhos possam ser posicionados relativamente a este container. A largura é definida como 100% do seu container pai e aplica um espaço superior de 10 pixels..

#### .contactForm .input-box label

*.contactForm* *.input-box*  label{

    position: absolute;

    left: 0;

    padding: 5px 0;

    font-size: 16px;

    margin: 10px 0;

    pointer-events: none;

    transition: 0.3s;

}

**Descrição:** Estiliza as etiquetas (label) dentro da .input-box no formulário de contato. Define a posição absoluta em relação ao seu container pai, alinhando à esquerda (left: 0). Aplica preenchimento de 5 pixels na parte superior e inferior, tamanho da fonte de 16 pixels e margem de 10 pixels na parte superior e inferior. Desativa os eventos do ponteiro (pointer-events: none) para que o clique não afete a etiqueta. Define uma transição suave de 0.3 segundos para efeitos de transição.

*.contactForm* *.input-box* input*:focus* ~ label,

*.contactForm* *.input-box* input*:valid* ~ label,

*.contactForm* *.input-box* textarea*:focus* ~ label,

*.contactForm* *.input-box* textarea*:valid* ~ label{

    transform: translateY(-20px);

    font-size: 12px;

}

**Descrição:**

.contactForm .input-box input:focus ~ label: Define o estilo da etiqueta (label) dentro de .input-box quando o campo de entrada (input) está em foco (:focus). Move a etiqueta para cima por meio de transform: translateY(-20px) para criar um efeito de flutuação. Define o tamanho da fonte como 12 pixels (font-size: 12px).

.contactForm .input-box input:valid ~ label: Define o estilo da etiqueta (label) dentro de .input-box quando o campo de entrada (input) é válido (:valid). Isso ocorre quando o usuário preenche corretamente o campo, por exemplo, ao inserir um endereço de e-mail válido. Assim como o anterior, move a etiqueta para cima e define o tamanho da fonte como 12 pixels.

.contactForm .input-box textarea:focus ~ label: Define o estilo da etiqueta (label) dentro de .input-box quando a área de texto (textarea) está em foco (:focus). Move a etiqueta para cima e define o tamanho da fonte como 12 pixels.

.contactForm .input-box textarea:valid ~ label: Define o estilo da etiqueta (label) dentro de .input-box quando a área de texto (textarea) é válida (:valid). Move a etiqueta para cima e define o tamanho da fonte como 12 pixels..

#### form input[type=submit]

form input[*type*=submit]{

    padding: 10px 40px;

    border: 0;

    border-radius: 10px;

    background-color: #e0003c;

    font-weight: 600;

    transition: .2s;

    cursor: pointer;

}

**Descrição:** Estiliza o botão de envio do formulário. Define padding de 10 pixels na parte superior e inferior e 40 pixels nas laterais, sem borda, bordas arredondadas de 10 pixels, cor de fundo #e0003c, peso da fonte em 600, transição suave de 0.2 segundos e define o cursor como ponteiro.

### Estilo do Footer

* footer: Define o estilo do rodapé, com plano de fundo semi-transparente, largura total, altura mínima de 100 pixels, preenchimento interno de 3 rem na parte superior e inferior e 1 rem nas laterais. Adiciona bordas arredondadas no canto superior esquerdo e direito e uma margem superior de 80 pixels.
* .row: Estilo para as linhas dentro do rodapé, usando flexbox para distribuir os itens com espaçamento uniforme entre eles.
* .col: Define o estilo para as colunas individuais dentro do rodapé, com largura mínima de 250 pixels, texto colorido com a variável --secondary-color, fonte poppins e preenchimento interno nas laterais.

### Dark theme

**:root: Define as variáveis CSS globais que podem ser acessadas em todo o documento CSS.**

* --primary-color: Define a cor primária como branco (#fff) por padrão.
* --secondary-color: Define a cor secundária como preto (#000) por padrão.

.dark-theme: Modifica as variáveis de cor para refletir um tema escuro.

* --primary-color: Altera a cor primária para preto (#000).
* --secondary-color: Altera a cor secundária para branco (#fff).

### Estrutura das Media Queries

As media queries são um recurso poderoso no CSS utilizado para criar designs responsivos, ajustando o layout e os estilos de um site com base no tamanho da tela do dispositivo do usuário. A estrutura geral das media queries envolve a definição de regras CSS específicas que se aplicam somente quando determinadas condições de mídia são atendidas, como a largura da tela.

#### Estrutura Básica

A estrutura básica de uma media query no CSS é a seguinte:

@media screen and (max-width: 1080px){

    }

#### Condições Comuns

**Max-width**: Define estilos para dispositivos com largura máxima especificada.

(max-width: 1080px)

窗体底端

#### Descrição Geral das Adaptações

1. **Layout Flexível**: As flex containers são configuradas para mudar de direção (linha para coluna) para melhor uso do espaço vertical em telas menores.
2. **Centralização de Texto**: Muitos elementos de texto, como cabeçalhos e descrições, são centralizados para melhor legibilidade.
3. **Redimensionamento de Elementos**: Imagens e botões são redimensionados para se adequarem a telas menores.
4. **Ocultação de Elementos**: Alguns menus e outros elementos são ocultados ou exibidos conforme o tamanho da tela.

### Conclusão

Ao longo deste projeto de desenvolvimento web, foquei na criação de um site moderno e responsivo. Utilizei HTML semântico para estruturação, CSS avançado para estilização flexível e JavaScript para interatividade dinâmica. A aplicação de técnicas como media queries garantiu que o site fosse adaptável a diferentes dispositivos, proporcionando uma experiência de usuário consistente. Com isso, o projeto não apenas atendeu aos requisitos funcionais e estéticos, mas também priorizou a otimização de desempenho e a acessibilidade, visando oferecer uma navegação fluida e eficiente aos usuários finais.

By: Giovani dos Santos

2024