

PROJETO FINAL PROGRAMAÇÃO WEB

FORMANDO: DERCIO Y. L. VIEIRA
FORMADOR: WILLIAM PIRES
MODULO: PROGRAMAÇÃO WEB

LOCAL: EIMC - MINDELO

Índice

Introdução:	3
Estrutura Principal do Projecto	3
<!DOCTYPE html>:	3
<i>Estrutura Geral</i>	3
<i>Cabeçalho</i>	3
<i>Seções de Conteúdo</i>	4
<i>Formulário de Contato</i>	4
<i>Galeria de Projetos</i>	5
<i>Modal de Visualização</i>	6
Estrutura de estilização do Projeto	6
CSS (style.css):	6
<i>Responsividade</i>	6
<i>Design das Abas e Botões</i>	8
Estrutura organizacional do Projeto	15
JavaScript (script.js)	15
<i>Modal de Imagem e Vídeo</i>	15
<i>Interação com o Formulário</i>	16
Desafios e Soluções:	17
Organização do Conteúdo	17
Responsividade	17
Integração JavaScript	17
Resultados e Conclusão	17

Introdução:

O meu projeto "Portfolio Dércio" consiste na criação do meu site pessoal, apresentando informações sobre sua carreira, experiência profissional, projetos finais e formas de contato. O site foi desenvolvido utilizando HTML, CSS e JavaScript para fornecer uma experiência interativa e informativa aos visitantes.

Estrutura Principal do Projeto

<!DOCTYPE html>:

Define o tipo de documento e a versão do HTML que o navegador deve esperar. Este é o primeiro elemento em um documento HTML padrão.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Portfolio Dércio</title>
  <link rel="stylesheet" href="style.css">
```

Estrutura Geral: O documento HTML utiliza elementos semânticos como <header>, <main>, <footer>, para estruturar o conteúdo de maneira organizada e acessível.

```
<body>
  <header></header>
  <main></main>
  <footer></footer>
</body>
```

Cabeçalho (<header>) inclui o nome de Dércio Vieira, abas de navegação e uma imagem de cabeçalho.

```
<header>
  <div class="abas">
    <h1 class="meunome">Dércio Vieira</h1>
    <div id="aba-1" class="aba ativa" data-aba="conteudo-1" onclick="mostrarAba(this.id)">Sobre min</div>
    <div id="aba-2" class="aba" data-aba="conteudo-2" onclick="mostrarAba(this.id)">Experiencia</div>
    <div id="aba-3" class="aba" data-aba="conteudo-3" onclick="mostrarAba(this.id)">Projectos</div>
    <div id="aba-4" class="aba" data-aba="conteudo-4" onclick="mostrarAba(this.id)">Contato</div>
    
  </div>
</header>
```

Seções de Conteúdo (<main>): Contém diferentes seções (DIV divisão genérica ou um contêiner que agrupa elementos) para Sobre mim, Experiência, Projetos e Contato. Cada seção é alternada através de **abas** que são ativadas dinamicamente pelo JavaScript.

Tabela de Experiencia.

<table> define uma tabela HTML.

<thead>, <tbody>são usados para agrupar os cabeçalhos, corpo e rodapé de uma tabela, respectivamente.

<tr> define uma linha na tabela.

<th> define um cabeçalho de célula na tabela.

<td> define uma célula de dados na tabela.

```
<div id="conteudo-2" class="conteudo">
  <table>
    <thead>
      <tr>
        <th>Entidade Empregadora</th>
        <th>Ocupação</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Dream Tours</td>
        <td>Guia Turístico</td>
      </tr>
    </tbody>
  </table>
</div>
```

Formulário de Contato: Permite aos visitantes enviar mensagens diretamente através do site .

<form> define um formulário HTML para coletar informações do usuário.

<label> define uma etiqueta descritiva para um controle de formulário.

<input> define um controle de entrada dentro de um formulário (como texto, e-mail, etc.).

<textarea> define uma área de texto multilinha dentro de um formulário.

<button> define um botão clicável dentro de um formulário.

**** um contêiner inline para texto ou elementos encaixados. Pode ser usado com classes e estilos para aplicar formatação específica.

<a> define um hiperlink usado para navegar para outra página, recurso externo ou local dentro da página.

```
<div id="conteudo-4" class="conteudo">
  <nz>Contato</nz>
  <form action="#" method="post">
    <label for="nome">Nome:</label><br>
    <input type="text" id="nome" name="nome" required><br><br>

    <label for="email">Email:</label><br>
    <input type="email" id="email" name="email" required><br><br>

    <label for="comentario">Comentário:</label><br>
    <textarea id="comentario" name="comentario" rows="4" cols="50" required></textarea><br><br>

    <input type="submit" value="Enviar">
    <a href="https://www.facebook.com/seu_perfil" target="_blank" class="facebook-link">
    <span class="icon-facebook"></span>
  </a>
```

Galeria de Projetos: Mostra uma variedade de projetos finais utilizando imagens e vídeos, com a capacidade de abrir uma modal para visualização detalhada.

**** Define uma imagem para ser exibida na página HTML. O atributo src especifica o caminho para a imagem e o atributo alt fornece um texto alternativo para acessibilidade.

<video> Define um vídeo ou um contêiner de mídia para conteúdo de vídeo.

<source> Especifica a localização de um arquivo de mídia de vídeo, e o tipo MIME para o navegador escolher o melhor formato suportado.

```

<div id="conteudo-3" class="conteudo">
  <h2>Projectos Finais</h2>
  <div class="galeria">

    <div class="img-container">
      
      <p>Projectos final de <br>Disigner de algoritimo.</p>
    </div>
    <br>

    <div class="img-container">
      
      <p>Projectos final de <br>Disigner de Interface.</p>
    </div>
  </div>
</div>

```

Modal de Visualização: Utiliza JavaScript para exibir imagens e vídeos em tamanho ampliado com descrição associada.

```

<div id="myModal" class="modal">
  <div class="modal-content">
    <img id="modal-img" src="" alt="imagem">
    <p id="modal-description"></p>
  </div>
</div>

```

Estrutura de estilização do Projeto

CSS (style.css):

```

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

```

Define a aparência visual do site, incluindo fontes, cores, layout responsivo e espaçamento, que são estilizadas nos elementos como <head>, <body>, <header>, <main>, <footer>, nas class e id.

Responsividade: Garante que o site seja amigável para dispositivos móveis através de técnicas como flexbox e media queries.

Em que no elemento **body**; ulitizei:

Font-family- Define o estilo do corpo da página.

Margin- Define A margem é removida.

Padding- Define preenchimento de 8%.

Background image- Difine a imagem de fundo.

Letter-spacing- Define o espaçamento entre letras.

Word-spacing- Define o espaçamento entre palavras.

Font-size-Define o tamanho da fonte.

```
body{  
  
    font-family: sans-serif;  
    margin: 0;  
    padding: 8%;  
    background-image: url('imagem/abstract-1891289_1920.jpg');  
    font-size: 20px;  
    letter-spacing: 2px;  
    word-spacing: 0.4px;  
    color: white;
```

Em que no elemento **Table**; ulitizei:

Width-define a largura em 50%

Border-collapse-Define que as bordas são colapsadas.

Background-color-Define a cor de Fundo da Table

Box-shadow- Define sombra é aplicada à tabela.

Margin-left.A tabela é centralizada horizontalmente com margem de 25%

```
table {  
  
    width: 50%;  
    border-collapse: collapse;  
    background-color: white;  
    box-shadow: 0 2px 4px rgb(134, 6, 6);  
    color: black;  
    margin-left: 25%;
```

Em que no elemento **Th, td, tr: hover**; ulitizei:

Th, td, O estilo das células de cabeçalho (th) e dados (td). O preenchimento é de 12px, e as bordas são definidas com uma cor específica.

Th, especifica o estilo das células de cabeçalho (th). A cor de fundo é vermelha, e a cor do texto é branca.

Tr: hover; aplica estilos às linhas (tr) quando o mouse passa sobre elas. A cor de fundo e a cor do texto mudam.

```
th, td {
  padding: 12px;
  border: 1px solid #8B0000;
}

th {
  background-color: #8B0000;
  color: white;
}

tr:hover {
  background-color: #8B0000;
  color: white;
}
```

Em que no elemento class **imgcabecalho**; se estiliza uma imagem de cabeçalho. Define a altura e a largura para 100px, adiciona uma borda preta e torna a imagem circular com border-radius: 50%.

```
.imgcabecalho{
  height: 100px;
  width: 100px;
  border: 1px solid black;
  border-radius: 50%;
}
```

Design das Abas e Botões: Estiliza abas de navegação, botões de formulário e eleme

.abas

```
.abas{
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  align-self: center;
  align-items: center;
  padding-bottom: 5%;
}
```


Descrição: Estiliza um contêiner flexível para abas. Os itens são distribuídos uniformemente, alinhados ao centro, e há um preenchimento inferior de 5%.

.aba

```
.aba{  
    margin-right: 10px;  
    padding: 10px 15px;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
    cursor: pointer;  
}
```

Descrição: Define o estilo de uma aba individual. Inclui margem, preenchimento, borda, bordas arredondadas e um cursor de ponteiro.

.aba:hover

```
.aba:hover{  
    transform: scale(1.1);  
    background-color: white;  
    box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.5);  
    color: black;  
}
```

Descrição: Aplica estilos à aba quando o mouse passa sobre ela. A aba aumenta de tamanho, a cor de fundo muda, uma sombra é adicionada, e a cor do texto muda.

.aba.ativa

```
.aba.ativa{  
    background-color: rgb(128, 32, 32);  
    color: #fff;  
}
```

Descrição: Define o estilo de uma aba ativa. A cor de fundo é um vermelho escuro, e a cor do texto é branca.

.conteudo

```
.conteudo{
  padding: 20px;

  border-radius: 5px;
  display: none;
  text-align: center;

  margin: 20px;
}
```

Descrição: Estiliza o contêiner de conteúdo. Inclui preenchimento, bordas arredondadas, alinhamento de texto ao centro, e margem. O contêiner está oculto por padrão.

.conteudo.ativa

```
.con MDN Reference
  display: block;
}
```

Descrição: Mostra o contêiner de conteúdo quando a classe ativa é aplicada.

.galeria

```
.galeria{
  Click to collapse the range.
  flex-wrap: wrap;
}
```

Descrição: Define um contêiner flexível para a galeria de imagens. Os itens são organizados em múltiplas linhas, se necessário.

.img-container

```
.img-container{
  Click to collapse the range.
  flex-direction: row;
  margin: 10px;
  border: solid 1px #ccc;
  border-radius: 5px;
  text-align: center;
  background-color: #808080;
}
```

Descrição: Estiliza o contêiner de imagem. Inclui flexibilidade, margens, borda, bordas arredondadas, alinhamento de texto ao centro e cor de fundo.

.img-container img

```
.img-container img {  
  margin-top: 20px;  
  width: 200px;  
  height: 150px;  
  object-fit: cover;  
  cursor: pointer;  
}
```

Descrição: Define o estilo das imagens dentro do contêiner. Inclui margens, dimensões específicas, ajuste da imagem para cobrir o contêiner e um cursor de ponteiro.

.img-container img

```
.img-container img:hover{  
  transform: scale(0.96);  
}
```

Descrição: Reduz o tamanho da imagem levemente quando o mouse passa sobre ela.

.modal

```
.modal{  
  position: fixed;  
  z-index: 1;  
  left: 0;  
  top: 0;  
  width: 100%;  
  height: 100%;  
  overflow: auto;  
  background-color: rgba(0, 0, 0, 0.8);  
}
```

Descrição: Estiliza o modal (janela pop-up). O modal está oculto por padrão, ocupa toda a tela, tem um fundo semi-transparente preto e é posicionado de forma fixa.

.modal-content

```
.modal-content{
  background-color: rgb(128, 32, 32);
  margin: 15% auto;
  padding: 20px;
  border: 1px solid #888;
  width: 120%;
  max-width: 600px;
```

Descrição: Estiliza o conteúdo dentro do modal. Inclui cor de fundo, margens automáticas para centralização, preenchimento, borda, e dimensões específicas.

Form

```
form {
  background-color: rgb(128, 32, 32);
  padding: 20px;
  border-radius: 5px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  max-width: 500px;
  margin: auto;
```

Descrição: Define o estilo dos formulários. Inclui cor de fundo, preenchimento, bordas arredondadas, sombra, largura máxima e centralização.

label, input, textarea

```
label, input, textarea {
  display: flex;
  width: 100%;
  margin-bottom: 10px;
  justify-content: center;
```

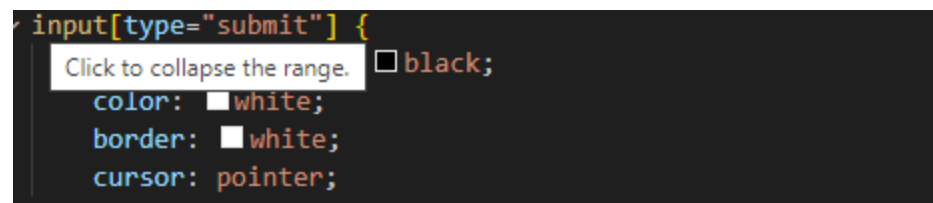
Descrição: Estiliza rótulos, campos de entrada e áreas de texto. Todos são definidos como flexíveis, com largura total, margem inferior e alinhamento centralizado.

input, textarea

```
input, textarea {
  padding-top: 5px;
  padding-bottom: 5px;
  border: 1px solid #ddd;
  border-radius: 4px;
```

Descrição: Especifica preenchimento superior e inferior, bordas e bordas arredondadas para campos de entrada e áreas de texto.

Input(type-submit)



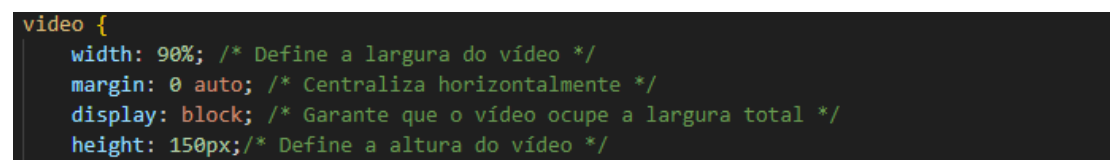
Descrição: Define o estilo dos botões de envio. Inclui cor de fundo preta, cor do texto branca, sem borda visível e um cursor de ponteiro.

in Input(type-submit): hover



Descrição: Altera a cor de fundo e a cor do texto dos botões de envio quando o mouse passa sobre eles.

Video



Descrição: Estiliza elementos de vídeo. A largura é definida como 90%, o vídeo é centralizado horizontalmente, e dimensões específicas são aplicadas.

#conteudo-1

```
#conteudo-1{
  width: 50%;
  margin-left: 25%;
}
```

Descrição: Define o estilo de um elemento com ID conteudo-1. A largura é de 50% e é centralizado horizontalmente com margin-left.

.facebook-link

```
.facebook-link {
  display: inline-block;
  width: 40px; /* Largura do ícone, ajuste conforme necessário */
  height: 40px; /* Altura do ícone, ajuste conforme necessário */

  text-decoration: none; /* decoração do texto é none, nenhum */
  border-radius: 50%; /* Para criar um círculo */
}
```

Descrição: Estiliza links do Facebook. Eles são exibidos como blocos em linha, com dimensões específicas, sem sublinhado e bordas arredondadas.

.icon-facebook

```
.icon-facebook {
  display: block;
  width: 100%;
  height: 100%;
  background-image: url(imagem/Vector-Facebook-Logo-Black-And-White-PNG-Background-Image.png) ;
  background-size: cover;
}
```

Descrição: Estiliza o ícone do Facebook. O ícone é exibido como um bloco, com a imagem de fundo ajustada para cobrir o contêiner.

Footer

```
footer {
  border: 1px solid ■ white;

  background-color: ■ black;
  height: 100px;
  width: 60%;
  margin-left: 20%;
  background-image: url(imagem/th-removebg-preview.png);
}
```

Descrição: Define o estilo do rodapé. Inclui borda branca, cor de fundo preta, dimensões específicas, centralização horizontal e uma imagem de fundo.

Estrutura organizacional do Projeto

JavaScript (script.js): Gerencia a exibição dinâmica de conteúdo através de funções como **mostrarAba** para alternar entre diferentes seções.

Modal de Imagem e Vídeo: Permite a visualização expandida de imagens e vídeos quando clicados na galeria de projetos.

```
function mostrarAba(abaSelecionada){  
    // 1- apanhar todos os elementos com a classe "aba"  
    const abas = document.querySelectorAll(".aba");  
    // 2 - Remover a classe "ativa" dos elementos com a classe "aba"  
    abas.forEach((aba) => aba.classList.remove("ativa"));  
    // 3 - Adicionar a classe "ativa" ao elemento que se deseja ativar  
    // O elemento que se deseja ativar é passado para a função através do atributo "abaSelecionada"  
    document.getElementById(abaSelecionada).classList.add("ativa");  
  
    // 4 - apanhar todos os elementos com a classe "conteudo"  
    const conteudos = document.querySelectorAll(".conteudo");  
    // 5 - remover a classe "ativa" dos elementos com a classe "conteudo"  
    conteudos.forEach((conteudo) => conteudo.classList.remove("ativa"));  
  
    // 6 - apanhar o id do conteúdo que se deseja apresentar, através da propriedade "data-aba" do elemento referenciado pelo id abaSele  
    const abaConteudo = document.getElementById(abaSelecionada).getAttribute("data-aba");  
    // 7 - adiciona a classe "ativa" ao elemento que se deseja mostrar  
    document.getElementById(abaConteudo).classList.add("ativa");  
}
```

Função para Abrir Modal

A função **abrirModal** é responsável por abrir um modal com uma imagem e uma descrição específicas. Esta função recebe dois parâmetros: **imgSrc** e **imgDescription**.

Descrição dos parâmetros:

imgSrc: URL da imagem que será exibida no modal.

imgDescription: Descrição da imagem que será exibida no modal.

A função realiza as seguintes operações:

Define a fonte da imagem do modal.

Define o conteúdo da descrição da imagem do modal.

Exibe o modal, alterando seu estilo para **display: block**.

```
function abrirModal(imgSrc, imgDescription){
    document.getElementById("modal-img").src = imgSrc;
    document.getElementById("modal-description").innerHTML = imgDescription;
    document.getElementById("myModal").style.display="block";
}
```

Fechar Modal ao Clicar Fora

Um evento é adicionado ao objeto window para fechar o modal quando o usuário clica fora da área do modal.

Este evento verifica se o elemento clicado é o modal. Se for, o modal é fechado.

```
window.onclick = function(event){
    if(event.target == document.getElementById("myModal")){
        document.getElementById("myModal").style.display = "none";
    }
}
```

Interação com o Formulário: Validação básica de campos de entrada e envio de dados através do formulário de contato.

Função de Validação de Formulário

A função validateForm valida os campos de um formulário antes de permitir o envio. Ela verifica o nome, email e comentário.

```
function validateForm() {
    var nome = document.getElementById('nome').value;
    var email = document.getElementById('email').value;
    var comentario = document.getElementById('comentario').value;

    if (!nome.match(/[A-Za-z\s]{1,}/)) {
        alert('Por favor, insira um nome válido.');
```

```
        return false;
    }

    if (!email.match(/^([a-zA-Z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$)/)) {
        alert('Por favor, insira um email válido.');
```

```
        return false;
    }

    if (comentario.trim() === '') {
        alert('Por favor, insira um comentário.');
```

```
        return false;
    }

    return true;
}
```


Validações realizadas:

Nome: Verifica se o nome contém apenas letras e espaços.

Email: Verifica se o email está no formato correto.

Comentário: Verifica se o campo de comentário não está vazio.

Se qualquer uma dessas validações falhar, um alerta é exibido e a função retorna false, impedindo o envio do formulário.

Desafios e Soluções:

Organização do Conteúdo: Garantir que as informações estivessem claramente organizadas e acessíveis foi um desafio. Utilizamos elementos semânticos e estruturação adequada para melhorar a navegação.

Responsividade: Assegurar que o site fosse igualmente funcional em dispositivos móveis exigiu testes e ajustes contínuos no CSS.

Integração JavaScript: Implementar transições suaves entre abas e funcionalidades de modalidade exigiu um bom entendimento de eventos e manipulação do DOM.

Resultados e Conclusão:

O projeto "Portfolio Dércio" foi feito para demonstrar as minhas habilidades, experiências aprendidas durante esses modulo de “programação web”. E usa-la de forma harmónica para HTML, CSS e JavaScript que permitiu criar uma experiência interativa e informativa para os visitantes do site.