

Projeto para desenvolvimento de website para clínica odontológica

Davi Portes -10417139;

David Pessoa - 10402647;

Pedro Henrique -10737627;

Lucas Lago – 10436247;

A ideia do projeto

Este projeto tem como objetivo elaborar um website para uma clínica de odontologia, que seja capaz de: apresentar a clínica, mostrar os serviços prestados, exibir as informações para contato e permitir que o paciente agende uma consulta ao mesmo tempo em que cumpre os requisitos pedidos pela disciplina. Logo, o projeto apresenta seu caráter extensionista ao facilitar o acesso da comunidade local a serviços odontológicos de qualidade, melhorando a comunicação entre a clínica e seus pacientes.

Além disso, o código fonte deste projeto está disponível publicamente no GitHub, com um tutorial explicativo. Isso permite que outras clínicas e desenvolvedores, por exemplo, possam adaptá-lo, ampliando o impacto do mesmo para além da clínica específica.

Explicação do código desenvolvido:

Para a elaboração do código HTML, o mesmo foi dividido em 7 partes diferentes:

- Header: Fica no topo da página e contém a navbar e o logo
- Seção de Início: Apresenta uma imagem do consultório
- Seção de especialidades: Apresenta os serviços oferecidos pela profissional dentista
- Seção Sobre: Apresenta uma imagem da profissional ao lado de um texto exibindo sua trajetória profissional
- Status Bar: Apresenta dados quantitativos sobre a profissional
- Seção de depoimentos: apresenta um carrossel de depoimentos de alguns pacientes sobre o atendimento oferecido

- Rodapé: Apresenta um formulário para contato ao lado de um iframe exibindo a localização do consultório no Google Maps e informações de contato (endereço, telefone, whatsapp, etc.).

Sendo assim, utilizaram-se as tags: <header>, para a seção do header, <footer> para a seção do rodapé e <section> para as demais seções.

Header:

```
<!-- Cabeçalho -->
<header class="dark-background">
  <!-- Logo -->
  <figure class="logo">
    
  </figure>
  <nav>
    <ul>
      <li><a href="#inicio"><h4>Início</h4></a></li>
      <li><a href="#especialidades"><h4>Especialidades</h4></a></li>
      <li><a href="#sobre"><h4>Sobre</h4></a></li>
      <li><a href="#depoimentos"><h4>Depoimentos</h4></a></li>
      <li><a href="#contato"><h4>Contato</h4></a></li>
    </ul>
  </nav>
</header>
```

A estrutura do cabeçalho é definida dentro da tag <header>, que possui a classe dark-background para aplicar uma cor de fundo azul escuro. Dentro dele, temos:

- Uma <figure class="logo"> que contém a imagem do logotipo da clínica.
- Uma tag <nav> que agrupa os links de navegação.
- Uma lista não ordenada () onde cada item () contém um link (<a>). Esses links direcionam o usuário para as seções correspondentes na mesma página (ex: href="#especialidades").

CSS do Header

```
/* Header */
header{
  padding: 0.8em 2em;
  display: flex;
  align-items: center;
  justify-content: space-between;
}

/* Logo */
header .logo img {
  height: 5rem;
  width: auto;
}

header nav ul{
  display: flex;
  justify-content: center;
  list-style: none;
  gap: 2vw;
}

header nav ul li a{
  color: #FFF;
  text-decoration: none;
  transition: color 0.3s ease;
}

header nav ul li a:hover {
  color: #d6d6d6;
}
```

O CSS é responsável por posicionar o logotipo e os links de navegação, geralmente usando `display: flex` no `<header>`. A classe `.dark-background` aplica um fundo escuro, e outras regras definem as cores e o estilo dos links, garantindo que o cabeçalho seja visualmente coeso e funcional.

Início:

```
<!-- Seção de Início -->
<section id="inicio">
  <!-- Imagem do consultório -->
</section>
```

A estrutura é minimalista: `<section id="inicio">`. Embora esteja vazia no HTML, ela serve como um contêiner que será totalmente estilizado pelo CSS.

```

/* Seção - Início imagem do consultório */
#inicio{
  background-image: url('/src/img/carrossel_consultorio/exterior.png');
  background-repeat: no-repeat;
  background-size: cover;
  background-position: center;
  height: 70vh;
}

```

O seletor #inicio no CSS é usado para aplicar um estilo visual marcante. Geralmente, isso inclui:

- background-image: Uma imagem de alta qualidade do consultório é definida como plano de fundo.
- height: Uma altura significativa de 70vh para que a seção ocupe a maior parte da tela inicial.
- background-size: cover: Garante que a imagem de fundo cubra toda a área da seção sem distorção.

```

alert(`
  Você ganhou um cupom de desconto!
  Use o cupom: NWEJKLWXJ!#
  E ganhe 30% no seu próximo atendimento!
`)

```

Ainda que não haja interatividade JavaScript específica para esta seção, foi criado um alert de cupom de desconto, executado pelo script.js, que aparece quando a página carrega e serve como uma forma de engajamento inicial com o visitante.

Especialidades:

```

<!-- Seção de Especialidades/Serviços -->
<section id="especialidades" class="white-background">
  <h1>Especialidades</h1>
  <ul id="listaEspecialidades">
    <!-- Lista renderizada dinamicamente pelo Javascript -->
  </ul>
</section>

```

A seção especialidades exibe os serviços numa lista de boxes para cada serviço oferecido em que há uma imagem demonstrativa e um texto explicativo, que aparece quando o usuário passa o mouse por cima. Os itens da lista são renderizados dinamicamente pelo Javascript. As informações de nome, descrição

da e caminho para a imagem ilustrativa de cada especialidade foram armazenadas no arquivo dados.json.

```
// Obtém dados do JSON com os depoimentos
fetch('./dados.json')
  .then(response => {
    if (!response.ok) {
      throw new Error("Erro ao carregar o JSON: " + response.status);
    }
    return response.json();
  })
  .then(data => {

    // Carrega boxes com as especialidades dinamicamente
    const servicos = data.especialidades;
    servicos.forEach((especialidade) => {
      const item = document.createElement('li');
      item.innerHTML = `
        <figure>
          
          <figcaption>
            <h4>${especialidade.nome}</h4>
            <p>${especialidade.texto}</p>
          </figcaption>
        </figure>
      `;
      listaEspecialidades.appendChild(item);
    });
  });
```

Esta é a parte mais importante. O script.js executa as seguintes ações:

1. Usa a função fetch para carregar os dados do arquivo dados.json.
2. Acessa o array especialidades dentro dos dados JSON.
3. Para cada especialidade, cria um elemento de lista () com o HTML correspondente (imagem, nome e descrição).
4. Adiciona (appendChild) cada novo item à listaEspecialidades no DOM, fazendo com que os cards apareçam na página.

```

/* Especialidades/Serviços */
#especialidades ul{
  display: flex;
  flex-wrap: wrap;
  list-style: none;
  max-width: 1440px;
  margin: 0 auto;
  gap: 3.188rem;
  justify-content: center;
  transition: all;
}

#especialidades figure{
  position: relative;
  display: flex;
  flex-flow: column;
  max-width: 25rem;
  overflow: hidden;
  box-shadow: 5px 5px 6px rgba(0,0,0,0.5);
}

#especialidades figure:hover img{
  display: block;
  border-radius: 7px;
}

#especialidades figure figcaption{
  background-color: rgba(4, 120, 139, 0.91);
  color: #FFF;
}

```

O CSS estiliza a aparência dos cards de especialidades. Regras para `#especialidades ul` usam `display: flex` e `flex-wrap: wrap` para organizar os itens em uma grade responsiva. Em cada item (``) da lista, é colocada uma tag `<figure>` contendo uma imagem ilustrativa (``) e um `<figcaption>` que possui o título e a descrição da especialidade ofertada. Os cards foram estilizados para que o usuário, ao passar o mouse por cima de uma figura, a descrição da respectiva especialidade apareça.

Sobre:

```
<!-- Seção Sobre -->
<section id="sobre" class="blue-background">
  <h1>Sobre</h1>
  <article class="sobre-container">
    <aside class="sobre-coluna">
      <h3>A dentista</h3>
      <figure class="sobre-figura">
        
        <figcaption>
          <h5 class="medium">Dra. Alzira Varão Lima</h5>
          <h5>Periodontista formada pela USP</h5>
        </figcaption>
      </figure>
    </aside>

    <p class="sobre-coluna">
      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque viverra sem viverra,
      vulputate augue quis, interdum urna. Etiam pharetra ex non auctor tincidunt.
      Etiam ornare leo in rutrum elementum. Curabitur in urna dolor. Sed feugiat ullamcorper massa in egestas.
      Sed consequat, sapien quis ornare porta, diam orci interdum diam, eget vehicula ipsum nulla vel risus.
      Phasellus efficitur hendrerit nisl, nec ultrices augue mollis at.
      Nunc sagittis interdum urna vitae fringilla. Sed iaculis in urna efficitur mattis.
      Vivamus eget posuere nibh, id elementum eros. Mauris bibendum erat ut eros faucibus,
      nec elementum velit tincidunt. Morbi ut nunc mauris. Duis feugiat ultrices aliquet.
      Maecenas sed ullamcorper sapien. Etiam sit amet feugiat magna. In hac habitasse platea dictumst.
      Vivamus congue justo nec euismod scelerisque. Duis pretium iaculis erat.
    </p>
  </article>
</section>
```

A seção `<section id="sobre">` contém um `<article class="sobre-container">` que organiza o conteúdo em duas colunas:

- Uma tag `<aside>` para exibir a foto e o nome da dentista.
- Um parágrafo (`<p>`) com o texto descritivo sobre a profissional.

```

/*sobre */
#sobre{
  padding: 2.5rem 0;
}

#sobre h1{
  margin: 0 0 1.25rem;
}

#sobre .sobre-container{
  display: flex;
  justify-content: center;
  gap: 7rem;
  align-items: center;
}

/*“cartão” da dentista */
#sobre .sobre-coluna{
  display: flex;
  flex-direction: column;
}

#sobre .sobre-figura{
  padding: 1rem;
  text-align: center;
  display: flex;
  flex-direction: column;
}

#sobre .sobre-figura img{
  max-height: 17rem;
}

```

A classe `.sobre-container` usa `display: flex` para criar o layout de duas colunas. O CSS também estiliza a imagem (`.sobre-figura img`), a legenda (`figcaption`) e o texto, garantindo uma leitura agradável e uma apresentação profissional. A classe `.blue-background` define uma cor azul claro como cor de fundo da seção.

Status Bar:

```
<!-- Barra de status -->
<section id="status-bar">
  <ul>
    <li class="estatistica">
      <h1 class="numero">33</h1>
      <span class="texto">Anos Atendendo</span>
    </li>
    <li class="estatistica">
      <h1 class="numero">+500</h1>
      <span class="texto">Pacientes</span>
    </li>
    <li class="estatistica">
      <h1 class="numero">+8k</h1>
      <span class="texto">Restaurações Feitas</span>
    </li>
    <li class="estatistica">
      <h1 class="numero">+30k</h1>
      <span class="texto">Consultas Feitas</span>
    </li>
    <li class="estatistica">
      <h1 class="numero">+15</h1>
      <span class="texto">Certificações</span>
    </li>
  </ul>
</section>
```

A status bar exibe numa lista não ordenada `` os dados quantitativos como: número de anos atendendo, quantidade de pacientes atendidos, quantidade de certificações e número de procedimentos realizados.

CSS da status bar:

```
/* Status-bar */
#status-bar {
  width: 100%;
  background-color: var(--dark-blue);
  padding: 1.25rem 0;
}

#status-bar ul{
  list-style: none;
  display: flex;
  flex-wrap: wrap;
  justify-content: space-evenly;
  align-items: center;
}

#status-bar .estatistica {
  text-align: center;
  margin: 20px;
}

#status-bar .numero {
  color: #FFF;
  font-size: 2.83rem; /* tamanho dos números */
  font-weight: 500;
  margin-bottom: 0.7rem;
}

#status-bar .texto {
  font-family: "Montserrat", sans-serif;
  color: #FFF;
  font-size: 1.5rem; /* tamanho do texto */
  display: block; /* faz o texto aparecer abaixo do número */
  font-weight: 300;
}
```

O seletor #status-bar ul usa display: flex para alinhar as estatísticas horizontalmente no Desktop e verticalmente em dispositivos mobile. O CSS define a cor de fundo, a tipografia dos números (.numero) e dos textos (.texto), criando um forte contraste visual que chama a atenção para os dados, com números em branco num fundo azul escuro.

Depoimentos:

```
<!-- Depoimentos -->
<section id="depoimentos" class="blue-background">
  <h1>Depoimentos</h1>
  <article class="swiper">
    <ul id="listaDepoimentos" class="swiper-wrapper">
      <!-- Lista renderizada dinamicamente pelo Javascript -->
    </ul>
    <!-- If we need pagination -->
    <span class="swiper-pagination"></span>
  </article>

  <!-- If we need navigation buttons -->
  <button class="swiper-button-prev"></button>
  <button class="swiper-button-next"></button>
</section>
```

Os depoimentos são exibidos dentro de uma tag como boxes num carrossel. A lista é renderizada dinamicamente pelo Javascript, já que as informações sobre cada um deles (texto, autor e avaliação) estão guardados no arquivo dados.json. Para aplicar o carrossel, foi usada a biblioteca Swiper do Javascript.

- : Elemento para os indicadores de paginação (bolinhas).
- <button class="swiper-button-prev/next">: Botões de navegação para avançar e retroceder os slides.

```

// Carrega Boxes com os depoimentos dinamicamente
const depoimentos = data.depoimentos;
depoimentos.forEach((depoimento) => {
  const item = document.createElement("li");
  item.className = "dark-background swiper-slide";

  // Adiciona as estrelas
  let estrelas = depoimento.estrelas;
  let starLine = "<span class='stars'>";
  for(let i = 0; i < 5; i++)
  {
    if(estrelas > 0)
      starLine += '<i class="fa-solid fa-star"></i>';
    else
      starLine += '<i class="fa-regular fa-star"></i>';

    estrelas--;
  }
  starLine += "</span>";

```

```

// Adiciona depoimento
item.innerHTML = `
  <span class="quotes">"</span>
    <blockquote>
      ${depoimento.texto}
    </blockquote>
  <span class="quotes below">"</span>
  <cite>${depoimento.pessoa}</cite>
  ` + starLine;
listaDepoimentos.appendChild(item);
});
})
.catch(error => console.error(error));

```

```
// Lógica do Carrossel de depoimentos
const swiper = new Swiper('.swiper', {
  // Optional parameters
  direction: 'horizontal',
  slidesPerView: "auto",
  centeredSlides: false,
  loop: true,

  // If we need pagination
  pagination: {
    el: '.swiper-pagination',
    clickable: true
  },

  // using "ratio" endpoints
  breakpoints: {
    600: {
      slidesPerView: "auto",
      spaceBetween: 50,
      centeredSlides: true,
    },
    300: {
      slidesPerView: "auto",
      spaceBetween: 30,
      centeredSlides: true,
    },
  },

  // Navigation arrows
  navigation: {
    nextEl: '.swiper-button-next',
    prevEl: '.swiper-button-prev',
  },
});
```

O JavaScript tem duas funções principais na seção de depoimentos:

1. Carregar Depoimentos: Assim como na seção de especialidades, ele busca os dados de dados.json, percorre o array depoimentos, cria o HTML para cada slide (incluindo as estrelas de avaliação) e os adiciona à listaDepoimentos.
2. Inicializar o Carrossel: Ele cria uma nova instância do Swiper, passando as configurações de comportamento, como direção, número de slides visíveis (slidesPerView), loop infinito e os elementos de navegação e paginação. Os breakpoints ajustam essas configurações para diferentes tamanhos de tela.

```

/* Depoimentos */
#depoimentos{
  margin-top: 10vh;
  padding-top: 3vh;
  padding-bottom: 3vh;
}

#depoimentos ul{
  display: flex;
  padding-bottom: 20px;
}

#depoimentos .swiper .swiper-pagination{
  margin-bottom: 5px;
}

#depoimentos .swiper-button-prev{
  left: var(--swiper-navigation-sides-offset, 60px);
  right: auto;
}

#depoimentos .swiper-button-next{
  left: auto;
  right: var(--swiper-navigation-sides-offset, 60px);
}

#depoimentos .swiper-button-prev, .swiper-button-next.swiper-button-next{
  color: #FFF;
  background-color: transparent;
  border: none;
}

```

O CSS estiliza a aparência dos slides (#depoimentos ul li e .dark-background), como tamanho, cor de fundo e preenchimento. As Media Queries ajustam o tamanho dos slides e a posição dos botões de navegação para telas menores, garantindo a responsividade do carrossel.

Rodapé:

```

<!-- Rodapé com formulário, iframe do Google Maps e Informações de Contato -->
<footer id="contato">
  <section class="contato-container">
    <form class="formulario">
      <h3>Formulário de Contato</h3>

      <fieldset class="input-field">
        <label for="nome">Nome</label>
        <input type="text" id="nome" name="nome">
      </fieldset>

      <fieldset class="input-field">
        <label for="email">Email</label>
        <input type="email" id="email" name="email">
      </fieldset>

      <fieldset class="input-field">
        <label for="assunto">Assunto</label>
        <input type="text" id="assunto" name="assunto">
      </fieldset>
    </form>
  </section>
</footer>

```

```

<fieldset class="input-field">
  <label for="especialidade">Selecione uma especialidade</label>
  <select id="especialidade" name="especialidade">
    <option>Selecione</option>
    <option>Estética</option>
    <option>Dentística</option>
    <option>Periodontia</option>
    <option>Endodontia</option>
    <option>Ortodontia</option>
    <option>Implantodontia</option>
  </select>
</fieldset>

<fieldset class="input-field">
  <label for="modalidade">Modalidade</label>
  <select id="modalidade" name="modalidade">
    <option>Presencial</option>
    <option>Teleconsulta</option>
  </select>
</fieldset>

<fieldset class="input-field">
  <label for="mensagem">Mensagem</label>
  <textarea id="mensagem" name="mensagem"></textarea>
</fieldset>

<button type="submit" class="medium">Enviar</button>
</form>

```

```

<iframe class="mapa"
  src="https://maps.google.com/maps?q=Rua%20Chemin%20Del%20Pra,%20172,%20São%20Paulo&t=m&z=15&output=embed&iwloc=near"
  title="Mapa da clínica">
</iframe>

<address class="info">
  <h3>Fale Conosco</h3>
  <ul>
    <li><i class="fa-2x fa fa-phone"></i><h5>(11) 98839-8134</h5></li>
    <li><i class="fa-2x fab fa-whatsapp"></i><h5>(11) 98839-8134</h5></li>
    <li><i class="fa-2x fa fa-envelope"></i><h5>consultoria.alzira@gmail.com</h5></li>
    <li><i class="fa-2x fa fa-location-dot"></i><h5>Rua Chemin Del Pra, nº 172, São Paulo, Santana</h5></li>
  </ul>
</address>
</section>
</footer>

<script src="https://cdn.jsdelivr.net/npm/swiper@11/swiper-bundle.min.js"></script>
<script src="./src/js/script.js"></script>
</body>
</html>

```

O formulário do rodapé é envolvido pela tag <form>, em que cada entrada é representada por um <input> (textos curtos), <textarea> (textos longos) <select> (lista de opções). Cada uma das entradas, assim como suas <labels> (explicação do input) é envolvida num <fieldset> e no fim do formulário há o botão de submit. Ao lado do formulário é possível encontrar o iframe que exibe a localização do consultório no Google Maps por meio da tag <iframe>. Do lado direito, há uma lista das informações de contato envolvidas pela tag <address>.

Além disso, foi usada a biblioteca Parsley do Javascript para realizar a validação dos campos do formulário quando o usuário submetê-lo para envio.

```
/* contato */
#contato{
  padding: 2.5rem 1.5rem 1rem 1.5rem;
  display: flex;
  justify-content: center;
}

#contato h3{
  color: #000;
  margin: 0 0 1.25rem 0;
}

/* container contato */
#contato .contato-container{
  display: flex;
  gap: 2rem;
  padding: 0 0.7rem;
  align-items: center;
}

/* formulário */
#contato .formulario{
  background: #FFF;
  border-radius: 0.875rem;
  padding: 1.5rem 3%;
  box-shadow: 0 0.375rem 1.125rem rgba(0,0,0,0.08);
  width: 30%;
  display: flex;
  flex-direction: column;
  align-items: center;
}
```

O seletor `.contato-container` usa `display: flex` para organizar o formulário, o mapa e as informações lado a lado em telas grandes. Em telas pequenas, uma Media Query altera `flex-direction` para `column`, empilhando os elementos verticalmente para melhor visualização em dispositivos móveis. O CSS também estiliza todos os elementos do formulário e as informações de contato. No arquivo `parsley.css`, são realizadas as estilizações das mensagens de erro vindas do parsley.