Introdução às Tecnologias Web

O Ecossistema Web

O desenvolvimento web front-end é baseado em três tecnologias principais que trabalham em conjunto:

- HTML (HyperText Markup Language): Responsável pela estrutura e conteúdo da página
- CSS (Cascading Style Sheets): Responsável pela aparência e estilização
- JavaScript: Responsável pelo comportamento e interatividade

Analogia com Construção Civil

Para entender melhor a relação entre essas tecnologias, imagine a construção de uma casa:

- HTML seria a estrutura da casa: paredes, teto, portas e janelas
- CSS seria o acabamento: pintura, papéis de parede, cores e decoração
- JavaScript seria as instalações: eletricidade, encanamento e sistemas de automação

Cada tecnologia tem seu papel específico e todas trabalham juntas para criar uma experiência completa para o usuário.

HTML - A Estrutura da Página

Anatomia de uma Tag HTML

As tags HTML seguem uma estrutura específica:

```
<!-- Tag com abertura e fechamento -->
<nome-da-tag atributo="valor">Conteúdo</nome-da-tag>
```

```
<!-- Tag auto-fechável -->
<input type="text" />

<!-- Exemplo prático -->
<div class="container" id="main-container">
    <p>Este é um parágrafo dentro de uma div</p>
</div>
```

Explicação:

- Tag de abertura: `<nome-da-tag>`
- Tag de fechamento: `</nome-da-tag>` (para tags que contêm conteúdo)
- Atributos: Fornecem informações adicionais (sempre na tag de abertura)
- Conteúdo: O que fica entre as tags de abertura e fechamento

Estrutura Básica de um Documento HTML

Todo documento HTML5 segue uma estrutura padrão que o navegador espera encontrar:

```
<!DOCTYPE html>
<!--
  DOCTYPE: Não é uma tag HTML, mas uma declaração obrigatória
  Informa ao navegador que este é um documento HTML5
  Deve ser a primeira linha do documento
-->
<html lang="pt-br">
<!--
  <html>: Elemento raiz que envolve todo o conteúdo HTML
  lang: Atributo que define o idioma principal da página
  Importante para acessibilidade e motores de busca
-->
<head>
    <!--
      <head>: Contém metadados - informações sobre a página
      Estes elementos NÃO são visíveis para o usuário final
      Inclui título, codificação, links para CSS, etc.
    -->
    <meta charset="UTF-8">
    <!--
      charset: Define a codificação de caracteres
      UTF-8 suporta acentos, caracteres especiais e emojis
```

```
      Evita problemas com caracteres mal interpretados
    -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!--
      viewport: Configuração essencial para dispositivos móveis
      width=device-width: A largura da página iguala a largura do dispositivo
      initial-scale=1.0: Define o nível de zoom inicial como 100%
    -->
    <title>Meu Primeiro Site</title>
    <!--
      <title>: Define o título que aparece na aba do navegador
      Também é usado quando a página é salva nos favoritos
      Importante para SEO (Otimização para Motores de Busca)
    -->
</head>
<body>
    <!--
      <body>: Contém TODO o conteúdo visível da página
      Tudo que o usuário vê e interage está dentro do body
      Inclui textos, imagens, formulários, botões, etc.
    -->
    <h1>Olá, Mundo!</h1>
    <p>Minha primeira página web</p>
</body>
</html>
```

Principais Tags HTML e Suas Funções

Tags de Cabeçalho e Texto

As tags de texto definem a hierarquia e formatação do conteúdo:

```
<!-- Hierarquia de títulos - a SEMÂNTICA é crucial -->
<h1>Título Principal da Página</h1>    <!-- Único por página - mais importante para
SEO -->
<h2>Subtítulo da Seção Principal</h2>  <!-- Para seções principais do conteúdo -->
<h3>Subseção</h3>                      <!-- Para dividir seções em partes menores
-->
<h4>Tópico Específico</h4>             <!-- Para detalhamento adicional -->
<h5>Informação de Detalhe</h5>          <!-- Para informações menos importantes -->
<h6>Informação Menor</h6>               <!-- Raramente usado -->

<!-- Parágrafos e formatação de texto -->
```

```
<p>Este é um parágrafo de texto comum.</p>
<span>Texto inline sem quebra de linha</span>
<strong>Texto importante (negrito semântico)</strong>
<em>Texto enfatizado (itálico semântico)</em>
<br>                                    <!-- Quebra de linha forçada -->
<hr>                                    <!-- Linha horizontal divisória -->
```

Importante sobre hierarquia de títulos:

- Use apenas UM `<h1>` por página
- Não pule níveis (não vá de `<h1>` para `<h3>` sem `<h2>`)
- A hierarquia ajuda na acessibilidade e SEO

Tags de Lista

HTML oferece dois tipos principais de listas:

```
<!-- Lista não ordenada (com marcadores/bullets) -->
<ul>
    <li>Item da lista 1</li>
    <li>Item da lista 2</li>
    <li>Item da lista 3</li>
</ul>

<!-- Lista ordenada (com números ou letras) -->
<ol>
    <li>Primeiro item na sequência</li>
    <li>Segundo item na sequência</li>
    <li>Terceiro item na sequência</li>
</ol>
```

Tags de Formulário

Formulários permitem a interação do usuário com a página:

```
<form>
    <!-- Campo de texto básico -->
    <label for="nome">Nome Completo:</label>
    <input type="text" id="nome" name="nome">

    <!-- Campo de email com validação -->
    <label for="email">E-mail:</label>
```

```
    <input type="email" id="email" name="email">

    <!-- Campo de senha (texto oculto) -->
    <label for="senha">Senha:</label>
    <input type="password" id="senha" name="senha">

    <!-- Área de texto multilinha -->
    <label for="mensagem">Mensagem:</label>
    <textarea id="mensagem" name="mensagem" rows="4"></textarea>

    <!-- Lista suspensa (dropdown) -->
    <label for="cargo">Cargo:</label>
    <select id="cargo" name="cargo">
        <option value="">Selecione um cargo</option>
        <option value="dev">Desenvolvedor</option>
        <option value="design">Designer</option>
        <option value="gerente">Gerente</option>
    </select>

    <!-- Botões de rádio (seleção única) -->
    <label>
        <input type="radio" name="status" value="ativo"> Ativo
    </label>
    <label>
        <input type="radio" name="status" value="inativo"> Inativo
    </label>

    <!-- Checkboxes (seleção múltipla) -->
    <label>
        <input type="checkbox" name="skills" value="html"> HTML
    </label>
    <label>
        <input type="checkbox" name="skills" value="css"> CSS
    </label>
    <label>
        <input type="checkbox" name="skills" value="js"> JavaScript
    </label>

    <!-- Botão de envio -->
    <button type="submit">Enviar Formulário</button>
</form>
```

Tags Semânticas (IMPORTANTE PARA ACESSIBILIDADE E SEO)

As tags semânticas dão significado à estrutura da página:

```html
<header>
    <!-- Cabeçalho da página ou de uma seção -->
    <nav>
        <!-- Menu de navegação principal -->
        <a href="/">Página Inicial</a>
        <a href="/sobre">Sobre Nós</a>
        <a href="/contato">Contato</a>
    </nav>
</header>

<main>
    <!-- Conteúdo principal da página (DEVE SER ÚNICO) -->
    <article>
        <!-- Conteúdo autocontido e independente (post, artigo) -->
        <h2>Título do Artigo</h2>
        <section>
            <!-- Agrupamento temático de conteúdo -->
            <h3>Primeira Seção do Artigo</h3>
            <p>Conteúdo da primeira seção...</p>
        </section>
        <section>
            <h3>Segunda Seção do Artigo</h3>
            <p>Conteúdo da segunda seção...</p>
        </section>
    </article>

    <aside>
        <!-- Conteúdo relacionado indiretamente (sidebar) -->
        <h3>Links Relacionados</h3>
        <ul>
            <li><a href="#">Artigo Similar 1</a></li>
            <li><a href="#">Artigo Similar 2</a></li>
        </ul>
    </aside>
</main>

<footer>
    <!-- Rodapé da página ou seção -->
    <p>&copy; 2024 Minha Empresa. Todos os direitos reservados.</p>
</footer>
```

Vantagens das tags semânticas:

- Melhor acessibilidade para leitores de tela
- Melhor indexação por motores de busca
- Código mais legível e maintainable

- Melhor experiência do usuário

Atributos HTML Comuns

Atributos fornecem informações adicionais sobre os elementos:

```
<!-- Atributos globais (funcionam em quase todas as tags) -->
<div id="identificador-unico"
     class="grupo-de-estilos"
     title="Texto de dica"
     style="color: red;"
     hidden
     data-info="dados-personalizados">
    Conteúdo do elemento
</div>

<!-- Atributos específicos de tags -->
<img src="imagem.jpg" alt="Descrição da imagem para acessibilidade" width="300"
height="200">
<a href="https://exemplo.com" target="_blank">Abrir em nova janela</a>
<input type="text" required disabled placeholder="Digite seu nome">
```

Explicação dos atributos mais importantes:

- `id`: Identificador único (deve aparecer apenas uma vez na página)
- `class`: Agrupa elementos para estilização CSS (pode ser repetido)
- `src`: Especifica a fonte de mídia (imagens, vídeos, áudio)
- `href`: Define o URL de destino para links
- `alt`: Texto alternativo para imagens (essencial para acessibilidade)
- `required`: Torna o campo de formulário obrigatório
- `disabled`: Desabilita o elemento (não pode ser interagido)
- `placeholder`: Texto de exemplo dentro de campos de entrada

CSS - A Estilização da Página

Como Conectar CSS ao HTML

Existem três métodos para adicionar CSS a uma página HTML:

```html
<!-- Método 1: CSS Externo (RECOMENDADO para produção) -->
<link rel="stylesheet" href="estilos.css">

<!-- Método 2: CSS Interno (útil para pequenas quantidades) -->
<style>
    body {
        font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
    }
</style>

<!-- Método 3: CSS Inline (EVITAR - difícil de manter) -->
<p style="color: blue; font-size: 16px;">Texto azul</p>
```

Por que CSS externo é recomendado:

- Separação de concerns (conteúdo vs apresentação)
- Reutilização em múltiplas páginas
- Cache do navegador (melhor performance)
- Mais fácil de manter e escalar

Anatomia de uma Regra CSS

Uma regra CSS é composta por partes específicas:

```css
seletor {
    propriedade: valor;
    outra-propriedade: valor;
}

/* Exemplo prático */
h1 {
    color: blue;
    font-size: 24px;
    margin-bottom: 20px;
}
```

Componentes de uma regra CSS:

- Seletor: Especifica quais elementos HTML serão estilizados
- Bloco de declaração: Conjunto entre chaves `{}`

- Declaração: Par propriedade-valor terminado com `;`
- Propriedade: Característica a ser alterada (cor, tamanho, etc.)
- Valor: Configuração específica para a propriedade

Seletores CSS - Como Selecionar Elementos

Seletores Básicos

Os seletores básicos são os mais comuns e importantes:

```
/* Seletor por elemento/tag */
p { color: black; }                    /* Todos os parágrafos */
h1 { font-size: 2em; }                 /* Todos os títulos h1 */
div { background: white; }             /* Todas as divs */

/* Seletor por classe (.) - MAIS UTILIZADO */
.destaque { background: yellow; }      /* Elementos com class="destaque" */
.btn-primary { color: white; }         /* Elementos com class="btn-primary" */
.card { border: 1px solid #ccc; }      /* Elementos com class="card" */

/* Seletor por id (#) - para elementos únicos */
#header { height: 100px; }             /* Elemento com id="header" */
#main-content { width: 80%; }          /* Elemento com id="main-content" */

/* Seletor universal (*) - todos os elementos */
* { margin: 0; padding: 0; }           /* Reset básico */
```

Seletores Combinadores

Combinadores permitem selecionar elementos baseados em suas relações:

```
css
/* Descendente (espaço) - todos os descendentes */
div p { color: blue; }                 /* Todos os p dentro de div, em
qualquer nível */

/* Filho direto (>) - apenas filhos imediatos */
div > p { color: red; }                /* Apenas p que são filhos diretos
de div */

/* Irmão adjacente (+) - elemento imediatamente após */
```

```
h1 + p { margin-top: 0; }              /* p que vem imediatamente após h1
*/

/* Irmão geral (~) - todos os irmãos após */
h1 ~ p { color: green; }               /* Todos os p que são irmãos de h1
(depois dele) */
```

Seletores de Atributo

Selecionam elementos baseados em seus atributos:

```
css
/* Por atributo específico */
input[type="text"] { border: 1px solid #ccc; }    /* Inputs do tipo text */

/* Por atributo com valor específico */
a[target="_blank"] { color: purple; }             /* Links que abrem em nova janela
*/

/* Por atributo que contém valor */
div[class*="container"] { max-width: 1200px; }    /* Divs com "container" no class
*/

/* Por atributo que começa com valor */
a[href^="https"] { font-weight: bold; }           /* Links que começam com https */

/* Por atributo que termina com valor */
a[href$=".pdf"]::after { content: " (PDF)"; }     /* Links para arquivos PDF */
```

Pseudo-classes e Pseudo-elementos

Pseudo-classes selecionam elementos em estados específicos, enquanto pseudo-elementos selecionam partes de elementos:

```
css
/* Pseudo-classes (estados do elemento) */
a:hover { color: red; }                /* Quando mouse passa por cima */
input:focus { border-color: blue; }    /* Quando elemento está focado */
button:active { transform: scale(0.95); } /* Quando clicando/mantendo
pressionado */
li:first-child { font-weight: bold; }  /* Primeiro filho da lista */
```

```
li:last-child { margin-bottom: 0; }    /* Último filho da lista */
tr:nth-child(even) { background: #f9f9f9; } /* Linhas pares da tabela */
input:disabled { opacity: 0.5; }       /* Elementos desabilitados */

/* Pseudo-elementos (partes do elemento) */
p::first-line { font-weight: bold; }   /* Primeira linha do parágrafo */
p::first-letter { font-size: 2em; }    /* Primeira letra do parágrafo */
::selection { background: yellow; }     /* Texto selecionado pelo usuário
*/
li::before { content: "→ "; }          /* Adiciona conteúdo antes */
li::after { content: "!"; }            /* Adiciona conteúdo depois */
```

O Box Model - Conceito FUNDAMENTAL do CSS

O Box Model é um dos conceitos mais importantes do CSS. Todo elemento é tratado como uma caixa retangular:

```
css
.caixa {
    width: 300px;           /* Largura do CONTEÚDO */
    height: 200px;          /* Altura do CONTEÚDO */
    padding: 20px;          /* Espaço INTERNO entre conteúdo e borda */
    border: 2px solid black;/* Borda ao redor do padding */
    margin: 30px;           /* Espaço EXTERNO além da borda */

    /* PROPRIEDADE CRÍTICA: box-sizing */
    box-sizing: border-box; /* Faz width incluir padding e border */
}
```

Visualização do Box Model:

```
text
[ Margem Externa (margin) ]
  [ Borda (border) ]
    [ Preenchimento Interno (padding) ]
      [ Conteúdo (content) - width/height ]
```

Diferença entre box-sizing:

- `content-box` (padrão): width = apenas conteúdo
- `border-box`: width = conteúdo + padding + border

Recomendação: Sempre use `border-box` para layouts mais previsíveis:

```
css
* {
    box-sizing: border-box;
}
```

Propriedades CSS Mais Comuns

Cores e Fundos

CSS oferece várias formas de definir cores e fundos:

```
css
.elemento {
    /* Sistemas de cores */
    color: #ff0000;                 /* Hexadecimal - mais comum */
    color: rgb(255, 0, 0);          /* RGB - valores de 0-255 */
    color: rgba(255, 0, 0, 0.5);    /* RGB com transparência (0-1) */
    color: hsl(0, 100%, 50%);       /* HSL - matiz, saturação, luminosidade */
    color: hsla(0, 100%, 50%, 0.5); /* HSL com transparência */

    /* Propriedades de fundo */
    background-color: blue;          /* Cor de fundo sólida */
    background-image: url('fundo.jpg'); /* Imagem de fundo */
    background-size: cover;          /* Cobrir todo o elemento */
    background-position: center;     /* Posicionar imagem no centro */
    background-repeat: no-repeat;    /* Não repetir a imagem */
    background: linear-gradient(to right, red, blue); /* Gradiente */
}
```

Texto e Tipografia

Controle da aparência do texto:

```
css
.texto {
    /* Família de fontes */
    font-family: Arial, Helvetica, sans-serif;  /* Fallbacks em caso de falha */

    /* Tamanho e peso da fonte */
```

```
    font-size: 16px;                /* Tamanho base (px, em, rem) */
    font-weight: bold;              /* Negrito (normal, bold, 100-900) */
    font-style: italic;             /* Itálico ou normal */

    /* Alinhamento e espaçamento */
    text-align: center;             /* left, center, right, justify */
    line-height: 1.5;               /* Espaçamento entre linhas */
    letter-spacing: 1px;            /* Espaçamento entre letras */
    text-decoration: underline;     /* underline, overline, line-through */

    /* Transformação de texto */
    text-transform: uppercase;      /* uppercase, lowercase, capitalize */
}
```

Layout e Posicionamento

Controle do fluxo e posição dos elementos:

```
css
.container {
    /* Display - como o elemento se comporta no fluxo */
    display: block;          /* Ocupa toda a linha (div, p, h1-h6) */
    display: inline;         /* Ocupa só o necessário (span, a, strong) */
    display: inline-block;   /* Híbrido - inline mas aceita width/height */
    display: none;           /* Elemento escondido e removido do fluxo */
    display: flex;           /* Ativa Flexbox (layout moderno) */
    display: grid;           /* Ativa CSS Grid (layout bidimensional) */

    /* Flexbox (explicação detalhada depois) */
    display: flex;
    justify-content: center; /* Alinhamento horizontal */
    align-items: center;     /* Alinhamento vertical */
    flex-direction: column;  /* Direção dos itens */

    /* Posicionamento */
    position: static;        /* Padrão - segue o fluxo normal */
    position: relative;      /* Relativo à sua posição original */
    position: absolute;      /* Relativo ao ancestral posicionado mais próximo */
    position: fixed;         /* Relativo à viewport (fica fixo na tela) */
    position: sticky;        /* Híbrido entre relative e fixed */

    /* Coordenadas (usadas com position não-static) */
    top: 10px;              /* Distância do topo */
    left: 20px;             /* Distância da esquerda */
    right: 30px;            /* Distância da direita */
```

```
    bottom: 40px;           /* Distância da base */
}
```

Espaçamento (Margin e Padding)

Controle dos espaços internos e externos:

```
css
.elemento {
    /* Margin - espaço EXTERNO (fora da borda) */
    margin: 10px;                    /* Todos os lados = 10px */
    margin: 10px 20px;               /* Top/Bottom = 10px, Left/Right = 20px */
    margin: 10px 20px 15px 5px;      /* Top, Right, Bottom, Left (sentido horário)
*/

    /* Propriedades individuais */
    margin-top: 10px;
    margin-right: 20px;
    margin-bottom: 15px;
    margin-left: 5px;

    /* Padding - espaço INTERNO (dentro da borda) - mesma sintaxe */
    padding: 20px;
    padding: 10px 15px;
    padding: 10px 15px 20px 5px;

    padding-top: 10px;
    padding-right: 15px;
    padding-bottom: 20px;
    padding-left: 5px;
}
```

Dica para lembrar a ordem:

- 1 valor: todos os lados
- 2 valores: vertical horizontal
- 3 valores: top horizontal bottom
- 4 valores: top right bottom left (sentido horário)

Flexbox - Sistema de Layout Moderno

Flexbox revolucionou o layout CSS, tornando-o mais intuitivo e poderoso:

```css
.container {
    /* Ativa o Flexbox */
    display: flex;

    /* Direção dos itens */
    flex-direction: row;            /* Padrão: esquerda para direita */
    flex-direction: row-reverse;    /* Direita para esquerda */
    flex-direction: column;         /* Topo para base */
    flex-direction: column-reverse; /* Base para topo */

    /* Alinhamento no eixo principal */
    justify-content: flex-start;    /* Início do container (padrão) */
    justify-content: center;        /* Centro do container */
    justify-content: flex-end;      /* Final do container */
    justify-content: space-between; /* Espaço igual entre itens */
    justify-content: space-around;  /* Espaço igual ao redor dos itens */
    justify-content: space-evenly;  /* Espaço igual entre e ao redor */

    /* Alinhamento no eixo cruzado */
    align-items: stretch;           /* Estica para preencher (padrão) */
    align-items: flex-start;        /* Topo do container */
    align-items: center;            /* Centro do container */
    align-items: flex-end;          /* Base do container */
    align-items: baseline;          /* Alinha pela linha de base do texto */

    /* Quebra de linha */
    flex-wrap: nowrap;              /* Todos os itens em uma linha (padrão) */
    flex-wrap: wrap;                /* Quebra em múltiplas linhas */
    flex-wrap: wrap-reverse;        /* Quebra em múltiplas linhas (invertido) */

    /* Espaço entre itens (substitui margin tricks) */
    gap: 10px;                      /* Espaço entre linhas e colunas */
    row-gap: 10px;                  /* Espaço apenas entre linhas */
    column-gap: 15px;               /* Espaço apenas entre colunas */
}

.item {
    /* Capacidade de crescimento */
    flex-grow: 0;                   /* Não cresce (padrão) */
    flex-grow: 1;                   /* Cresce para preencher espaço disponível */

    /* Capacidade de encolhimento */
    flex-shrink: 1;                 /* Encolhe se necessário (padrão) */
    flex-shrink: 0;                 /* Não encolhe (mantém tamanho) */

    /* Tamanho base */
```

```
    flex-basis: auto;               /* Tamanho base automático (padrão) */
    flex-basis: 200px;              /* Tamanho base fixo */
    flex-basis: 50%;                /* Tamanho base percentual */

    /* Shorthand flex */
    flex: 1;                        /* flex-grow: 1, flex-shrink: 1, flex-basis: 0%
*/
    flex: 0 0 200px;                /* flex-grow: 0, flex-shrink: 0, flex-basis:
200px */

    /* Ordem dos itens */
    order: 0;                       /* Ordem natural (padrão) */
    order: 1;                       /* Aparece depois dos itens com order: 0 */
    order: -1;                      /* Aparece antes dos itens com order: 0 */
}
```

Responsividade com Media Queries

Media Queries permitem criar designs que se adaptam a diferentes tamanhos de tela:

```
css
/* Mobile First - comece pelos estilos mobile */

/* Estilos base (para mobile) */
.container {
    width: 100%;
    padding: 10px;
}

/* Tablet (768px ou mais) */
@media (min-width: 768px) {
    .container {
        width: 750px;
        padding: 20px;
    }

    .menu {
        display: flex;
    }
}

/* Desktop (1024px ou mais) */
```

```css
@media (min-width: 1024px) {
    .container {
        width: 980px;
    }

    .sidebar {
        display: block;
    }
}

/* Large Desktop (1200px ou mais) */
@media (min-width: 1200px) {
    .container {
        width: 1140px;
    }
}

/* Outros tipos de media queries */
@media (max-width: 767px) {
    /* Apenas para telas menores que 768px */
}

@media (orientation: landscape) {
    /* Quando o dispositivo está na horizontal */
}

@media (prefers-color-scheme: dark) {
    /* Quando o usuário prefere modo escuro */
}
```

Abordagem Mobile First:

1. Comece com estilos para dispositivos móveis
2. Use `min-width` para adicionar estilos para telas maiores
3. Progressivamente melhore a experiência para dispositivos com mais recursos

Agora Vamos Praticar!

Vamos Construir Juntos o Sistema de Funcionários

Agora que entendemos os conceitos teóricos, vamos aplicar esse conhecimento

criando nosso projeto prático. Vamos construir um sistema de cadastro de funcionários que incorpora tudo o que aprendemos:

```
html
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sistema de Funcionários</title>
    <style>
        /* Vamos aplicar TODO o CSS que aprendemos! */

        /* Reset e configurações globais */
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        /* Estilos base para o body */
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            line-height: 1.6;
            color: #333;
            background-color: #f4f4f4;
            padding: 20px;
        }

        /* Container principal */
        .container {
            max-width: 800px;
            margin: 0 auto;
            background: white;
            padding: 30px;
            border-radius: 8px;
            box-shadow: 0 2px 10px rgba(0,0,0,0.1);
        }

        /* Cabeçalho */
        h1 {
            color: #2c3e50;
            text-align: center;
            margin-bottom: 30px;
        }
```

```
        /* Grupos de formulário */
        .form-group {
            margin-bottom: 20px;
        }

        /* Labels */
        label {
            display: block;
            margin-bottom: 5px;
            font-weight: bold;
        }

        /* Campos de entrada */
        input, select {
            width: 100%;
            padding: 10px;
            border: 1px solid #ddd;
            border-radius: 4px;
            font-size: 16px;
        }

        /* Botões */
        button {
            background: #3498db;
            color: white;
            border: none;
            padding: 12px 20px;
            border-radius: 4px;
            cursor: pointer;
            font-size: 16px;
        }

        /* Efeito hover nos botões */
        button:hover {
            background: #2980b9;
        }
    </style>
</head>
<body>
    <div class="container">
        <header>
            <h1>Cadastro de Funcionários</h1>
        </header>

        <main>
            <section class="form-section">
                <h2>Novo Funcionário</h2>
```

```
                <form id="employee-form">
                    <div class="form-group">
                        <label for="name">Nome:</label>
                        <input type="text" id="name" name="name" required>
                    </div>

                    <div class="form-group">
                        <label for="position">Cargo:</label>
                        <select id="position" name="position" required>
                            <option value="">Selecione um cargo</option>
                            <option value="Desenvolvedor">Desenvolvedor</option>
                            <option value="Designer">Designer</option>
                            <option value="Gerente">Gerente</option>
                        </select>
                    </div>

                    <div class="form-group">
                        <label for="department">Departamento:</label>
                        <input type="text" id="department" name="department"
required>
                    </div>

                    <button type="submit">Cadastrar</button>
                </form>
            </section>

            <section class="employees-section">
                <h2>Lista de Funcionários</h2>
                <div id="employees-list">
                    <p>Nenhum funcionário cadastrado</p>
                </div>
            </section>
        </main>
    </div>

    <script>
        // JavaScript básico para funcionalidade
        // (O foco da aula é HTML/CSS, mas vamos ver o básico do JavaScript)

        const form = document.getElementById('employee-form');
        const employeesList = document.getElementById('employees-list');

        form.addEventListener('submit', function(e) {
            e.preventDefault();

            const name = document.getElementById('name').value;
            const position = document.getElementById('position').value;
```

```
            const department = document.getElementById('department').value;

            // Criar elemento para o novo funcionário
            const employeeDiv = document.createElement('div');
            employeeDiv.className = 'employee-card';
            employeeDiv.innerHTML = `
                <h3>${name}</h3>
                <p><strong>Cargo:</strong> ${position}</p>
                <p><strong>Departamento:</strong> ${department}</p>
            `;

            // Adicionar à lista
            employeesList.appendChild(employeeDiv);

            // Limpar formulário
            form.reset();
        });
    </script>
</body>
</html>
```

Exercícios Práticos Durante a Aula

Exercício 1: Adicionar Novos Campos ao Formulário

- Adicione campos para email (type="email") e telefone (type="tel")
- Implemente um campo para data de admissão (type="date")
- Adicione radio buttons para o status (Ativo/Inativo)

Exercício 2: Melhorar a Estilização

- Adicione cores diferentes para os cards baseados no cargo
- Implemente hover effects nos cards de funcionário
- Adicione bordas arredondadas e sombras para melhor visual

Exercício 3: Implementar Responsividade

- Use media queries para ajustar o layout em tablets
- Adapte o design para mobile (stack vertical)
- Ajuste tamanhos de fonte e espaçamento para diferentes telas

Exercício 4: Utilizar Flexbox no Layout

- Reorganize o formulário usando flexbox para layout em colunas
- Use flexbox para alinhar os cards de funcionário
- Implemente um header flexbox com logo e navegação

Boas Práticas e Dicas Finais

HTML Semântico

- Sempre prefira tags semânticas (`<header>`, `<nav>`, `<main>`, etc.)
- Mantenha uma hierarquia lógica de cabeçalhos (h1 > h2 > h3)
- Use `alt` em todas as imagens para acessibilidade
- Valide seu HTML regularmente

CSS Organizado

- Use nomes de classes descritivas e consistentes (.btn-primary, .card-header)
- Agrupe propriedades relacionadas (layout, typography, colors)
- Comente seções do seu CSS para melhor organização
- Considere usar metodologias como BEM para nomenclatura