

# Guia de Preenchimento

## Documentação Técnica Projeto Final

### Módulo HTML5 e CSS3

---

Este guia explica, seção por seção, como preencher o arquivo “Template de Documentação do Trabalho — Curso Desenvolvedor Web (HTML5 & CSS3)”. Use-o como checklist durante todo o projeto.

#### 1. Visão geral

Objetivo da documentação: registrar decisões técnicas e evidências que demonstrem acessibilidade, responsividade e boas práticas ao longo dos 14 encontros do módulo.

Como será usada: a documentação é parte da avaliação (veja a rubrica). Cada item deve possuir uma evidência (link, imagem, commit, issue, relatório).

#### 2. Antes de começar

- Crie um repositório (Git) público/privado com acesso ao professor.
- Abra a estrutura mínima de pastas:

```
/
|-- index.html
|-- pages/
|-- css/estilos.css
|-- images/
|-- docs/ (coloque aqui capturas, PDFs e wireframes)
```

- Defina o nome do projeto e uma breve descrição (problema + solução).
- Padronize commits (ex.: Conventional Commits) e crie uma branch principal (`main`).

#### 3. Instruções por seção

##### 3.1 Identificação do Projeto

Preencha nome, equipe, links do repositório e da página publicada (GitHub Pages, Vercel, Netlify etc.).

Resumo: 2–3 linhas que expliquem para quem é o site, que valor entrega e quais páginas principais.

### 3.2 Arquitetura de Informação e Páginas

- Estrutura de diretórios: copie a árvore final do seu projeto. Se houver mudanças, atualize a árvore.
- Mapa do site: liste cada página prevista. Marque quando estiver concluída e publique o link.
- Links para wireframes/mockups: inclua URLs (Figma/Adobe XD/etc.) ou caminhos no repositório para as versões Desktop e Mobile de cada página.

**Tabela modelo — Links para wireframes/mockups**

Página/Tela	Breakpoint	Link	Versão	Data	Observações
Home	Desktop (1440px)	<a href="https://www.figma.com/file/ABC123?node-id=101">https://www.figma.com/file/ABC123?node-id=101</a>	v1.2	06/08	Fluxo de navegação incluído
Home	Mobile (375px)	repo:/docs/wireframes/home_mobile_v1.png	v1.0	06/08	Estados vazio/erro

### 3.3 Requisitos

- Escreva requisitos como frases testáveis. Separe funcionais e não funcionais.
- Funcionais descrevem o que o site faz (ex.: “o usuário consegue enviar o formulário de contato”).
- Não funcionais descrevem qualidades (ex.: “tempo de carregamento < 2s”, “contraste ≥ 4.5:1”).

**Tabela modelo — Requisitos**

ID	Descrição	Critério de Aceite	Evidência (link)
RF-01	Formulário de contato envia mensagem.	Ao preencher campos obrigatórios e clicar em Enviar, exibe confirmação.	<a href="https://deploy.site/contato">https://deploy.site/contato</a>   captura: /docs/testes/contato_ok.png
RNF-01	Contraste de texto mínimo 4.5:1.	Relatório do Lighthouse ≥ 90 em Acessibilidade.	/docs/relatorios/lighthouse_acesse.pdf

### 3.4 Informações Técnicas — HTML5

- Indique as principais decisões de marcação semântica e acessibilidade que você adotou.
- Inclua: uso de landmarks (`header/nav/main/footer`), ordem de headings, link “pular para o conteúdo”, textos alternativos em imagens, estrutura de formulários e tratamento de erros.
- Exemplo de trecho com skip-link e main:

```
<a class="skip-link" href="#conteudo-principal">Pular para o conteúdo</a>
<main id="conteudo-principal" tabindex="-1">
  <h1>Contato</h1>
  <form aria-describedby="status">
    <label for="email">E-mail</label>
    <input id="email" type="email" required aria-required="true">
    <button>Enviar</button>
    <div id="status" aria-live="polite"></div>
  </form>
</main>
```

- Registre também quaisquer exceções (por exemplo, uso de `role` específico) e o motivo.

### 3.5 Informações Técnicas — CSS3

- Explique tokens (cores, espaçamentos), breakpoints, estratégia de responsividade (mobile-first), uso de Flexbox/Grid, política de fontes e foco visível.
- Exemplo de tokens e navegação responsiva (hamburger):

```
:root{ --text:#0a0a0a; --bg:#fff; --brand:#0b57d0; --
space:clamp(.75rem,2vw,1.25rem); }
.navbar{ display:flex; justify-content:space-between; align-items:center; }
.nav-toggle{ display:none; }
.nav-menu{ display:flex; gap:.75rem; }
@media (max-width:768px){
  .nav-toggle{ display:inline-flex; }
  .nav-menu{ display:none; flex-direction:column; }
  .nav-menu[data-open="true"]{ display:flex; }
}
```

- Documente como garante alto contraste e como trata `prefers-reduced-motion`.

### 3.6 Plano de Acessibilidade (Checklist)

- Marque [x] apenas quando houver evidência. Anexe capturas, relatório do Lighthouse/axe e observações.
- Para contraste, registre o par de cores e a razão medida.

### 3.7 Plano de Responsividade

- Liste breakpoints e descreva como o layout muda em cada um. Inclua evidências (capturas) em 360×640, 768×1024 e ≥1366×768, além do funcionamento do menu hamburger.

### 3.8 Plano de Testes

- Cada caso deve ter passos claros, resultado esperado e evidência. Use 'Status' = Em execução / Aprovado / Reprovado.

### 3.9 Como atingir a nota máxima na rubrica

- Estrutura & Semântica: 100% das páginas com landmarks e headings corretos.
- Acessibilidade: contraste ≥ 4.5:1, navegação por teclado, alt em todas as imagens, formulário com `aria-live`.
- Responsividade: sem scroll horizontal; Grid/Flex adaptam conteúdo; menu hamburger funcional.
- Boas práticas CSS: variáveis, baixa especificidade, componentes reutilizáveis.
- Formulários & Feedback: mensagens claras e focáveis; estados de erro e sucesso.
- Documentação: este guia preenchido com links válidos e capturas.

### 3.10 Registro de Alterações (Changelog)

- Preencha a cada mudança relevante. Recomenda-se mensagens de commit no padrão Conventional Commits (ex.: `feat: adiciona formulário de contato`).

### 3.11 Declaração do Estudante

- Preencha nome, data e assine digitalmente ou à mão quando finalizar o trabalho.

## 4. Padrões de nomenclatura e organização

- Pastas/arquivos: `kebab-case` sem espaços (ex.: `contato\_form.html`, `home-hero.jpg`).
- Imagens: prefixe por página (ex.: `home-banner-hero.jpg`).
- CSS: se usar BEM, siga `bloco\_elemento--modificador`.
- Commits: `feat`, `fix`, `docs`, `style`, `refactor`, `test`, `chore`.

## 5. Entregáveis e checklist final

- ☐ Site publicado (link acessível).
- ☐ Repositório com código, README e licença.
- ☐ Template de documentação preenchido com evidências.
- ☐ Relatórios de acessibilidade/responsividade anexados.
- ☐ Tabela de links de wireframes por página (desktop e mobile).

## 6. FAQ e erros comuns

- “Preciso de ícone no botão do menu?” — Opcional; mantenha o rótulo textual “Menu” para acessibilidade.
- “Posso usar framework (Bootstrap)?” — Sim, desde que registre o que foi usado e personalize com foco em acessibilidade e contraste.
- Erros comuns: mais de um `<h1>` por página; falta de `alt`; navegação sem foco visível; textos com contraste baixo; media queries apenas para desktop (não mobile-first).

## 7. Apêndice — Exemplos

Exemplo de ‘Home — Heading map’ (como checar organização de títulos):

```
<h1>Home</h1>
  <h2>Seção Destaque</h2>
  <h2>Serviços</h2>
    <h3>Serviço A</h3>
    <h3>Serviço B</h3>
  <h2>Depoimentos</h2>
  <h2>Contato</h2>
```

Modelo de commit: `feat: implementa menu hamburger com aria-controls`

Medir contraste: use ferramentas (Lighthouse, axe, contrast-ratio.com) e registre o par de cores com a razão obtida ( $\geq 4.5:1$ ).