

# Módulo 1 – Fundamentos

**Tipo de projeto:** P1 – Exercício Tradicional

**Nível:** 1

**Tema sugerido:** “Minha primeira página pessoal”

---

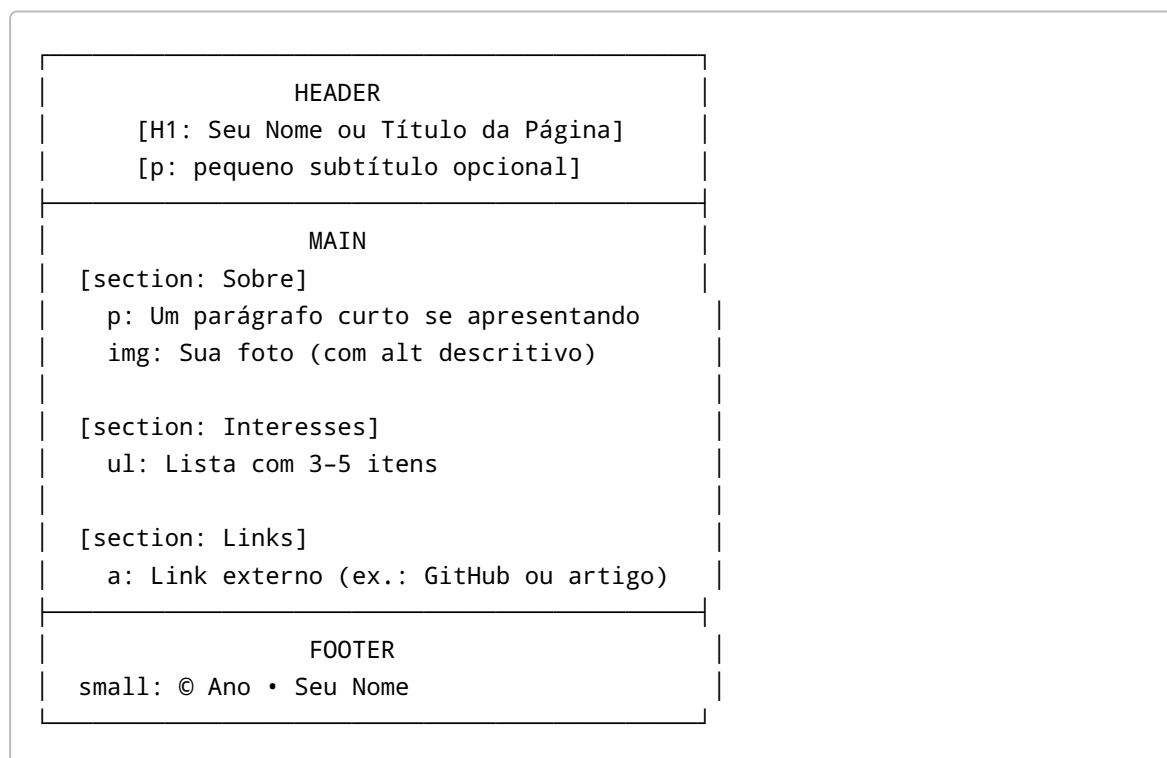
## (0) Objetivo deste nível

Criar **uma página HTML única** e totalmente semântica contendo: **um cabeçalho (header)** com título, **um parágrafo**, **uma lista**, **um link** e **uma imagem**, além de **meta tags básicas** (charset, viewport e description). Sem CSS avançado e sem JavaScript – foco total na **estrutura** e **semântica**.

---

## (1) Descrição visual (wireframe em texto)

A página é **minimalista**, centralizada e com largura máxima confortável para leitura. Pense em um cartão vertical simples:



**Tipografia:** fontes do sistema (sem importar nada).

**Cores:** padrão do navegador (sem styling extra).

**Espaçamentos:** uma linha em branco lógica entre seções (via quebras de bloco HTML).

**Imagens:** 1 imagem estática (por exemplo, `img/perfil.jpg`), largura natural; não forçar tamanho neste nível.

**Acessibilidade inicial:** texto alternativo significativo na imagem; hierarquia de heading correta (H1 único no header).

---

## (2) Git/GitHub — estrutura, commits e README

### Estrutura de pastas (nível 1)

```
meu-site-fundamentos/  
├─ index.html  
├─ css/  
│   └─ styles.css      # vazio ou com reset mínimo opcional  
├─ img/  
│   └─ perfil.jpg      # placeholder local  
└─ README.md
```

**Nota:** Embora o nível 1 não exija CSS, já deixamos a pasta `css/` criada para acostumar o fluxo de projeto.

### Commits de exemplo (Convencional Commits)

1. `feat: scaffold do projeto e estrutura de pastas`
2. `feat(html): cria index com semântica básica (header, main, footer)`
3. `docs(readme): adiciona instruções de uso e objetivos do nível 1`
4. `chore(assets): adiciona imagem placeholder em /img`

### Exemplo de comandos

```
# 1) iniciar repositório  
git init  
  
git add .  
  
git commit -m "feat: scaffold do projeto e estrutura de pastas"  
  
# 2) iterar no HTML e commitar  
git add index.html  
  
git commit -m "feat(html): cria index com semântica básica (header, main, footer)"  
  
# 3) adicionar README  
git add README.md  
  
git commit -m "docs(readme): adiciona instruções de uso e objetivos do nível 1"  
  
# 4) adicionar imagem placeholder  
git add img/perfil.jpg  
  
git commit -m "chore(assets): adiciona imagem placeholder em /img"
```

```
# 5) publicar no GitHub (após criar o repositório remoto)
# substitua <URL_DO_REPO> pela URL do seu repo
git branch -M main
git remote add origin <URL_DO_REPO>

git push -u origin main
```

## README.md (modelo)

```
# Minha Primeira Página Pessoal – Fundamentos (P1 • Nível 1)

Projeto do Módulo 1 – Fundamentos. Objetivo: criar uma página HTML simples e
semântica com header, parágrafo, lista, link, imagem e meta tags básicas.

## Como visualizar
Abra o arquivo `index.html` no navegador.

## Estrutura
- `index.html`: página principal.
- `css/styles.css`: arquivo de estilos (opcional neste nível).
- `img/perfil.jpg`: imagem de perfil (placeholder).

## Requisitos atendidos
- [x] `<!doctype html>` e `lang="pt-BR"`
- [x] `meta charset`, `meta viewport`, `meta description`
- [x] `header`, `main`, `footer`
- [x] `h1` único e descritivo
- [x] parágrafo introdutório
- [x] lista não ordenada com 3-5 itens
- [x] link externo (com `target="_blank"` + `rel="noopener noreferrer"`)
- [x] imagem com `alt` descritivo

## Licença
MIT
```

## (3) Código do projeto (HTML/CSS) — comentado

**Observação:** CSS é opcional neste nível. Abaixo, `styles.css` tem apenas um reset mínimo *opcional* para tornar o visual mais limpo sem desviar do foco (semântica).

`index.html`

```
<!doctype html>
<html lang="pt-BR">
  <head>
    <!-- Define a codificação de caracteres para acentuação correta -->
    <meta charset="utf-8" />
```

```

<!-- Garante boa renderização e zoom responsivo em dispositivos móveis -->
<meta name="viewport" content="width=device-width, initial-scale=1" />

<!-- Descrição curta da página (SEO básico) -->
<meta name="description" content="Minha primeira página pessoal usando
HTML semântico: apresentação rápida, interesses e link externo." />

<title>Minha Página Pessoal – Fundamentos</title>

<!-- CSS opcional do nível 1 (reset leve). Pode remover se preferir 100%
padrão do navegador. -->
<link rel="stylesheet" href="css/styles.css" />
</head>
<body>
  <!-- Cabeçalho da página: contém o H1 (título principal) -->
  <header>
    <h1>Seu Nome – Página Pessoal</h1>
    <!-- Subtítulo opcional para contexto -->
    <p>Olá! Esta é minha primeira página feita apenas com HTML semântico.</
p>
  </header>

  <!-- Conteúdo principal da página -->
  <main>
    <!-- Seção "Sobre" -->
    <section aria-labelledby="sobre-titulo">
      <!-- Use um heading por seção para manter a hierarquia clara -->
      <h2 id="sobre-titulo">Sobre mim</h2>
      <p>
        Sou uma pessoa curiosa aprendendo desenvolvimento web. Este projeto
        foi criado para praticar a estrutura semântica do HTML.
      </p>

      <!-- Imagem com alt descritivo. Use uma foto sua ou um placeholder.
-->
      <figure>
        
        <figcaption>Minha foto de perfil (placeholder).</figcaption>
      </figure>
    </section>

    <!-- Seção "Interesses" -->
    <section aria-labelledby="interesses-titulo">
      <h2 id="interesses-titulo">Interesses</h2>
      <!-- Lista não ordenada com 3-5 itens -->
      <ul>
        <li>HTML e semântica</li>
        <li>CSS e design de interfaces</li>

```

```

        <li>Acessibilidade na web</li>
        <li>Leitura e escrita</li>
        <li>Aprender algo novo todo dia</li>
    </ul>
</section>

<!-- Seção "Links" -->
<section aria-labelledby="links-titulo">
    <h2 id="links-titulo">Links</h2>
    <!-- Link externo: abra em nova aba com segurança -->
    <p>
        Visite meu perfil no
        <a href="https://github.com/" target="_blank" rel="noopener
noreferrer">GitHub</a>
        para acompanhar minha evolução.
    </p>
</section>
</main>

<!-- Rodapé com informação breve de direitos/autor -->
<footer>
    <small>© <span id="ano-atual"></span> Seu Nome. Todos os direitos
reservados.</small>
</footer>

<!-- JS opcional apenas para preencher o ano automaticamente (pode
remover) -->
<script>
    // Este pequeno script não é necessário para o nível 1, mas ajuda a
manter o ano atualizado.
    // Se preferir zerar o JavaScript neste nível, remova totalmente este
bloco.
    document.getElementById('ano-atual').textContent = new
Date().getFullYear();
</script>
</body>
</html>

```

css/styles.css (opcional)

```

/* Reset mínimo e leitura confortável. Opcional para nível 1. */
* { box-sizing: border-box; }
html { line-height: 1.5; }
body {
    margin: 1.5rem;
    max-width: 60ch; /* largura confortável de leitura */
    font-family: system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-serif;
}
header, main, footer { margin-block: 1rem; }

```

```
figure { margin: 0; }  
img { display: block; max-width: 100%; height: auto; }
```

## (4) Briefing para Figma — passo a passo para replicar o layout

**Objetivo:** reproduzir exatamente o resultado visual do HTML/CSS acima.

### 1) Arquivo & Frame

- Crie um arquivo chamado: **"P1 • Nível 1 — Página Pessoal"**.
- Insira um **Frame** desktop: Desktop (1440).
- Adicione um **Layout Grid** ao frame com **1 coluna**, margem lateral de **auto** (sem colunas múltiplas neste nível). Use a grade apenas como referência de centralização.

### 2) Estilos tipográficos

- **Fonte:** *System UI* (no Figma pode usar **Inter** como proxy).
- **Base:** 16 px.
- **H1:** 32 px, peso 700, line-height ~ 120%.
- **H2:** 24 px, peso 700, line-height ~ 130%.
- **Parágrafo e lista:** 16 px, line-height ~ 150%.
- **Small (footer):** 14 px.

### 3) Estrutura (layers)

No Frame principal, crie grupos/layers nesta ordem: 1. **Header**

- **H1** (Seu Nome — Página Pessoal)
- **p** subtítulo (opcional) 2. **Main**
- **Section: Sobre**
- **H2** "Sobre mim"
- **p** parágrafo
- **Figure**
- **Rectangle** 240×240 (placeholder de imagem)
- **Caption:** "Minha foto de perfil (placeholder)." - **Section: Interesses**
- **H2** "Interesses"
- Lista com 5 itens (5 linhas de texto) - **Section: Links**
- **H2** "Links"
- Linha de texto com palavra "GitHub" sublinhada para simular link 3. **Footer**
- **small** "© 2025 Seu Nome. Todos os direitos reservados."

### 4) Espaçamentos

- Espaço vertical entre **Header/Main/Footer:** **16 px**.
- Espaço entre **seções** dentro do main: **16-24 px**.
- Espaço entre um **heading** e seu conteúdo: **8-12 px**.
- **Largura máxima** do conteúdo (container) no centro: **~60ch** (simule com um **Frame** interno de **600-700 px** de largura).

## 5) Alinhamentos

- Todo o conteúdo **centralizado no eixo horizontal** do frame.
- Texto alinhado à esquerda.
- A imagem deve respeitar a largura máxima do container (não ultrapassar 600–700 px).

## 6) Cores

- Mantém cores padrão (preto no texto, fundo branco). Sem paleta custom neste nível.

## 7) Export / Inspeção

- Marque o retângulo da imagem como exportável em `@1x` e `@2x` (PNG).
- Use o painel **Inspect** para anotar tamanhos, line-heights e espaçamentos definidos.

## 8) Checklist de equivalência (Figma ↔ HTML)

- [ ] H1 único no Header
  - [ ] Hierarquia H1 → H2 nas seções
  - [ ] Parágrafo “Sobre” com conteúdo semelhante
  - [ ] Figura com legenda
  - [ ] Lista com 5 itens
  - [ ] Link “GitHub” indicado
  - [ ] Footer com small/ano
- 

## (5) O que praticar/observar ao revisar

- **Semântica:** cada bloco dentro da tag correta (`header`, `main`, `section`, `footer`).
  - **Acessibilidade:** `alt` descritivo na imagem; `id/aria-labelledby` relacionando títulos e seções.
  - **SEO básico:** `title` e `meta description` claros.
  - **Estrutura do repo:** manter organização mínima desde o início (pasta `img`, `css`).
  - **Commits pequenos e descritivos.**
- 

## (6) Próximos passos (preparando o Nível 2)

- Adicionar **mais artigos/seções** e praticar `<article>`, `<section>`, `<header>`, `<footer>` internos.
- Introduzir **múltiplas imagens** e **links externos** adicionais.
- Começar a **nomear classes** (mesmo sem CSS avançado) para futura estilização.