

Avaliação Web 1 – Desenvolvimento de Página Estática com HTML e CSS (Flexbox)

✓ Observações

- A atividade exige **atenção a cores, espaçamentos, tipografia, estrutura, containers, alinhamentos e elementos visuais**, mas **não utiliza JavaScript**.
 - A página **não deve ser responsiva** — ela pode ficar bonita apenas em telas largas ($\geq 1200\text{px}$).
 - Todo o layout deve ser construído usando **Flexbox** sempre que houver elementos lado a lado ou alinhamento horizontal/vertical.
 - Use **elementos semânticos de HTML5**, como: `header`, `nav`, `main`, `section`, `article`, `footer`, `h1-h6`, `ul`, `p`, etc.
 - Não use IDs para estilização generalizada. Prefira **classes reutilizáveis** (`.container`, `.card`, `.section`, `.cta-button`, etc).
 - IDs podem ser usados apenas quando forem **âncoras de navegação** (ex.: `id="packages"`).
 - Organize o CSS separando estilos gerais e específicos. Use convenções simples como:
 - Classes genéricas para padrões: `.container`, `.section`, `.title`, `.card`.
 - Classes específicas apenas quando necessário: `.hero-content`, `.package-card`.
-

🔍 Boas práticas adicionais que devem ser seguidas

1) Estrutura e organização do HTML

- Utilize indentação correta (2 ou 4 espaços).
 - Evite elementos extras desnecessários (divs sem função).
 - Mantenha a hierarquia correta de títulos:
 - A página deve ter **apenas um** `<h1>` (no Hero).
 - As demais seções usam `<h2>` e subtítulos `<h3>`.
 - Use parágrafos (`<p>`) para textos corridos, nunca `<div>`.
 - Use listas (``, ``) para itens de benefícios, pacotes, links, etc.
-

2) Organização e boas práticas no CSS

- Centralize o estilo geral no início:
 - Paleta de cores em variáveis (`:root`).
 - Fonte padrão.
 - Reset de margens e paddings.
- Mantenha uma ordem lógica dos seletores no arquivo CSS:

1. Reset
2. Variáveis
3. Elementos base (`body` , `h1` , `p` , etc.)
4. Layout geral (`.container` , `.section`)
5. Header
6. Hero
7. Seções subsequentes
8. Footer

- Evite repetir estilos idênticos em vários lugares (use classes reutilizáveis).
 - Para espaçamento, prefira padrões consistentes:
 - **20px / 30px / 60px / 100px**
 - Use nomes de classe claros e descritivos (inglês é preferido como convenção de mercado):
 - Correto: `.package-card` , `.section-title` , `.benefits-list`
 - Evitar: `.card1` , `.coisa` , `.teste123`
-

3) Boas práticas visuais

- Mantenha um alinhamento visual consistente entre seções.
 - Use a mesma proporção de margens e paddings em elementos semelhantes.
 - Respeite a mesma paleta de cores em toda a página (não invente cores novas).
 - Mantenha consistência nos estilos:
 - Todos os botões devem parecer iguais.
 - Todos os cards devem ter mesma largura e altura aproximada.
 - Todos os títulos de seção devem seguir o mesmo estilo.
 - Evite fontes muito pequenas (mínimo: 14–16px em textos e 2rem+ em títulos).
 - Sempre certifique-se de que o contraste de cores é legível (texto claro no fundo escuro ou vice-versa).
-

4) Estrutura do projeto e arquivos

- Separe o CSS em um arquivo externo (`styles.css`).
 - Organize as imagens em uma pasta própria: `assets/images/`
 - Se usar outras páginas (ex.: contato), mantenha pastas organizadas: `assets/css/` , `assets/images/` , `assets/js/` (mesmo que vazia).
 - Os ícones ficam em uma pasta própria `assets/icons/`
 - Não deixe arquivos soltos ou nomes confusos (ex.: `imagem.png` , `aaaa.jpg`).
 - Nomeie arquivos de forma clara:
 - `hero-background.jpg`
 - `evento-card.png`
-

5) Acessibilidade mínima

- Todas as imagens devem ter **atributo alt**.

- Os links devem ter texto compreensível (evitar “clique aqui”).
 - Botões devem ter textos dentro deles (nunca usar apenas ícones, nesta atividade).
-

6) Comentários úteis

Adicionar pequenos comentários no HTML é bom **quando realmente necessário**, como:

```
<!-- Packages Section -->
<!-- Footer -->
```

Evitar comentários supérfluos como:

```
<!-- DIV 1 -->
<!-- Testando -->
```

× 7) O que NÃO fazer

- Não usar `position: absolute` para montar o layout.
- Não usar tabelas para layout.
- Não usar IDs para estilizar tudo.
- Não criar um CSS gigante repetindo estilos.
- Não usar 20px em um lugar e 23px em outro “só porque sim”.
- Não estilizar elementos semânticos importantes, use classes, pense sempre no reuso (ex.: estilizar `<h1>` direto para algo diferente do padrão de seção).