

Documentação Completa - Landing Page Alura+

Visão Geral do Projeto

Este projeto é uma **landing page moderna e responsiva** para o serviço "Alura+", desenvolvida com HTML5 e CSS3. A página utiliza um **design elegante com tema escuro**, focado em apresentar os benefícios do combo de serviços educacionais da Alura de forma atrativa e profissional.

Características Principais:

- Layout responsivo com CSS Grid
- Design moderno com tema escuro
- Efeitos hover interativos
- Otimizada para dispositivos móveis
- Estrutura semântica bem definida

Análise do CSS

1. Variáveis CSS (Custom Properties)

```
css



:root {
  --branco-principal: #FFFFFF;
  --cinza-secundario: #C0C0C0;
  --botao-azul: #167BF7;
  --cor-de-fundo: #00030C;
  --fonte-principal: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unic
  --botao-azul-efeito: #3c92fa;
}
```

Conceitos de Aprendizado:

- **CSS Custom Properties:** Permitem reutilizar valores em todo o projeto
- **Manutenibilidade:** Facilita mudanças de tema e cores
- **Nomenclatura Semântica:** Usa nomes descritivos que facilitam o entendimento

Paleta de Cores:

- **Fundo Principal:**  #00030C (azul escuro quase preto)
- **Texto Principal:**  #FFFFFF (branco puro)
- **Texto Secundário:**  #C0C0C0 (cinza claro)

- **Botão Primário:**  #167BF7 (azul vibrante)
 - **Hover Effect:**  #3c92fa (azul mais claro para feedback visual)
-

2. Reset e Configurações Globais

```
css

* {
  margin: 0;
  padding: 0;
}

body {
  background-color: var(--cor-de-fundo);
  color: var(--branco-principal);
  font-family: var(--fonte-principal);
  font-size: 16px;
  font-weight: 400;
}
```

Conceitos Importantes:

- **Universal Reset:** Remove espaçamentos padrão dos navegadores
 - **Tipografia Base:** Define fonte, tamanho e peso padrão
 - **Tema Escuro:** Fundo escuro com texto claro para melhor legibilidade
-

3. Layout Principal com CSS Grid

```
css

.principal {
  background-image: url("../img/Background.png");
  background-repeat: no-repeat;
  background-size: contain;
  align-items: center;
  text-align: center;
}

.container {
  height: 100vh;
  display: grid;
  grid-template-columns: 50% 50%;
}
```

Técnicas Utilizadas:

- **CSS Grid:** Layout moderno para dividir conteúdo em colunas
 - **Viewport Height (100vh):** Ocupa toda a altura da tela
 - **Background Image:** Imagem de fundo sem repetição
 - **Grid 50/50:** Divisão equilibrada do espaço
-

4. Estilização de Botões

Botão Primário:

CSS

```
.container__botao {  
  background-color: var(--botao-azul);  
  border-radius: 5px;  
  padding: 1em;  
  color: var(--branco-principal);  
  display: block;  
  text-decoration: none;  
  margin-bottom: 1em;  
}
```

Botão Secundário:

CSS

```
.botao_secundario {  
  background-color: transparent;  
  color: var(--branco-principal);  
  border-radius: 5px;  
  padding: 1em;  
  display: block;  
  border: 2px solid var(--branco-principal);  
}
```

Design Pattern:

- **Hierarquia Visual:** Botão primário mais chamativo, secundário mais sutil
 - **Consistência:** Mesmo padding e border-radius
 - **Call-to-Action:** Botões bem definidos para conversão
-

5. Efeitos Interativos (Hover)

CSS

```
.container__botao:hover {  
    background-color: var(--botao-azul-efeito);  
    color: var(--cor-de-fundo);  
}  
  
.lista__link a:hover {  
    color: var(--botao-azul);  
}
```

🌟 UX/UI Benefits:

- **Feedback Visual:** Usuário recebe resposta imediata
 - **Engajamento:** Torna a interface mais interativa
 - **Consistência:** Cores de hover seguem o padrão do design
-

🏗️ Análise da Estrutura HTML

1. Head e Configurações Básicas

```
html  
  
<head>  
    <meta charset="UTF-8" />  
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
    <link rel="stylesheet" href="./styles.css" />  
    <title>Gabys Plus</title>  
</head>
```

📱 Responsividade:

- **Viewport Meta:** Essencial para design responsivo
- **UTF-8:** Suporte completo a caracteres especiais
- **External CSS:** Separação de responsabilidades

2. Seção Principal (Hero Section)

html

```
<section class="container_principal">
  <div class="container_caixa">
    <h1 class="container_titulo">
      Com o Combo+, você pode aproveitar a Alura+ e o Alura Língua por um preço único.
    </h1>
    
    <a href="https://www.alura.com.br/" class="container_botao">
      Assine por 12x de R$ 120,00*
    </a>
    <a href="https://www.alura.com.br/" class="container_botao botao_secundario">
      Assinar somente o Alura+
    </a>
    <p class="container_aviso">
      *O preço pode variar caso a assinatura seja feita em outros planos.
    </p>
  </div>
</section>
```

Elementos Principais:

- **H1:** Título principal com proposta de valor
- **CTA Duplo:** Duas opções de assinatura
- **Disclaimer:** Aviso legal sobre preços
- **Imagem:** Visual do produto/combo

3. Seções Secundárias (Features)

O projeto utiliza um padrão consistente para apresentar features:

html

```
<section class="container_secundario">
  
  <div class="container_descricao">
    <h2 class="descricao_titulo">Assista do seu jeito</h2>
    <p class="descricao_texto">
      Aproveite a tela grande da TV ou assista no tablet, laptop, celular e outros aparel
    </p>
  </div>
</section>
```

Padrão de Layout:

- **Alternância:** Imagem-texto, texto-imagem para ritmo visual

- **Grid Consistency:** Mantém estrutura 50/50
- **Hierarchy:** H2 para títulos, parágrafos para descrições

4. Seção de Dispositivos

html

```
<section class="dispositivos">
  <h2 class="dispositivos__titulo">Disponível nos seus dispositivos favoritos</h2>
  <ul class="dispositivos__lista">
    <li>
      
      <h3 class="lista__item">TV</h3>
    </li>
    <!-- Mais itens... -->
  </ul>
</section>
```

Características:

- **Lista Semântica:** Uso correto de `` e ``
- **Acessibilidade:** Alt texts descritivos
- **Flexbox Layout:** Para distribuição horizontal

Responsividade e Media Queries

Breakpoint Principal: 768px

```
@media screen and (max-width: 768px) {  
  .container {  
    display: block;  
    background-image: none;  
  }  
  
  .container__imagem {  
    width: 50vw;  
  }  
  
  .container__caixa {  
    margin: 2em 2em;  
  }  
  
  .dispositivos__lista {  
    display: block;  
  }  
  
  .rodape__lista {  
    display: block;  
  }  
}
```

Adaptações Mobile:

- **Grid → Block:** Layout em coluna única
 - **Background Removal:** Remove imagem de fundo
 - **Viewport Width:** Imagens responsivas com `(vw)`
 - **Flex → Block:** Listas empilhadas verticalmente
-

Estrutura Semântica

Hierarquia de Headings:

- **H1:** Título principal (único por página)
- **H2:** Títulos de seções
- **H3:** Títulos de itens da lista

Elementos Semânticos:

- `<section>`: Agrupa conteúdo relacionado
- `<footer>`: Informações finais e links

- `/`: Listas estruturadas
 - `` **com alt**: Acessibilidade para imagens
-

Metodologia CSS (BEM-like)

O projeto utiliza uma nomenclatura inspirada no BEM:

CSS

```
.container           /* Bloco */  
.container__titulo   /* Elemento */  
.container__botao    /* Elemento */  
.botao_secundario    /* Modificador */
```

✓ Vantagens:

- **Clareza**: Nomes descritivos
 - **Escalabilidade**: Fácil manutenção
 - **Organização**: Estrutura lógica
-

Melhores Práticas Implementadas

CSS:




- ✓ Uso de variáveis CSS
- ✓ Reset de estilos
- ✓ Layout moderno (Grid/Flexbox)
- ✓ Design responsivo
- ✓ Efeitos hover
- ✓ Nomenclatura consistente

HTML:

- ✓ Estrutura semântica
- ✓ Meta tags essenciais
- ✓ Alt texts nas imagens
- ✓ Links com HTTPS
- ✓ Hierarquia de headings

UX/UI:

- ✓ Contraste adequado

-  Botões bem definidos
 -  Layout limpo e organizado
 -  Feedback visual (hover)
-

Pontos de Aprendizado Principais

1. CSS Grid Layout

- Como criar layouts em grade
- Divisão proporcional do espaço
- Responsividade com grid

2. CSS Custom Properties

- Organização de valores
- Facilidade de manutenção
- Temas consistentes

3. Design Responsivo

- Media queries eficazes
- Adaptação mobile-first
- Unidades relativas (vh, vw, em)

4. Interatividade CSS

- Estados hover
- Transições suaves
- Feedback visual

5. HTML Semântico

- Estrutura lógica
 - Acessibilidade
 - SEO-friendly
-

Sugestões de Melhorias

Performance:

- Otimizar imagens (WebP, lazy loading)
- Minificar CSS

- Usar CDN para assets

Acessibilidade:

- Adicionar `aria-labels`
- Melhorar contraste em alguns elementos
- Incluir skip navigation

Funcionalidade:

- Adicionar animações CSS
- Implementar scroll suave
- Formulário de contato

SEO:

- Meta description
- Structured data
- Open Graph tags



Conclusão

Este projeto demonstra um **excelente domínio dos conceitos fundamentais** de desenvolvimento front-end moderno. A combinação de HTML semântico, CSS bem estruturado e design responsivo resulta em uma landing page profissional e funcional.

Pontos Fortes:

- Código limpo e organizado
- Design consistente e atraente
- Responsividade bem implementada
- Boas práticas de nomenclatura
- Estrutura escalável

Este é um **projeto sólido** que serve como excelente base para estudos e referência em desenvolvimento web moderno.