

APOSTILA 01 / 01 – MINI-CURSO DE 2H30

Seu primeiro site no *ar*.

Um cartão de visita digital, publicado na internet, com endereço próprio – em uma tarde, sem saber programar.

TEMA

HTML & CSS

DURAÇÃO

2h30

ETAPAS

07 práticas

NÍVEL

Iniciante

— ÍNDICE

Sumário.

12 capítulos, 7 etapas práticas. Use durante a aula pra acompanhar, e depois pra rever em casa quando quiser mexer no seu site.

01	O que você vai aprender	pg. 03
02	Como a web funciona (em 2 minutos)	pg. 04
03	As três ferramentas que você precisa	pg. 05
04	Etapa 1 — HTML cru	pg. 06
05	Etapa 2 — Conectando o CSS	pg. 12
06	Etapa 3 — Cabeçalho e Sobre	pg. 16
07	Etapa 4 — Serviços, Trabalhos e Contato	pg. 22
08	Etapa 5 — Páginas de detalhe	pg. 27
09	Etapa 6 — Polimento e celular	pg. 29
10	Etapa 7 — Publicando seu site (Netlify Drop)	pg. 31
11	Próximos passos	pg. 33
12	Glossário rápido	pg. 34

— CAPÍTULO 01 • OBJETIVOS

O que você vai aprender.

No fim destas 2h30 você vai sair daqui com um site seu, no ar, com endereço próprio — e a confiança pra abrir um editor de código e mexer.

Você vai sair com

- Um site seu, publicado na internet, com endereço próprio.
- Entendimento básico de HTML (o esqueleto do site).
- Entendimento básico de CSS (o visual do site).
- Confiança pra abrir um editor de código e mexer.

O que você NÃO precisa

- Saber programar.
- Ter computador potente.
- Falar inglês.
- Pagar nada por ferramenta.

O site que vamos fazer é um cartão de visita digital — o tipo de coisa que dá pra mandar no WhatsApp, colocar no perfil do Instagram, levar pra uma entrevista.

Uma página única com seu cartão no topo, um pouco sobre você, os serviços que você oferece, alguns trabalhos seus e formas de te contatar.

— CAPÍTULO 02 • FUNDAMENTOS

Como a web funciona.

Em 2 minutos: o que acontece quando você abre um site no celular ou no computador.

- 01 **Seu navegador pede.** Chrome, Safari, Firefox — eles pedem o site pra um computador na internet, que a gente chama de “servidor”.
- 02 **O servidor responde.** Manda de volta um monte de arquivos: HTML, CSS, imagens.
- 03 **O navegador monta.** Junta esses arquivos e desenha tudo na tela do seu jeito.

Hoje a gente vai criar esses arquivos no nosso computador, e depois colocar num servidor de graça (Netlify) pra qualquer pessoa do mundo poder abrir.

HTML, CSS e JavaScript — qual a diferença?

LINGUAGEM	PRA QUE SERVE	ANALOGIA
HTML	Conteúdo e estrutura (o que tem na página)	Esqueleto da casa
CSS	Visual (cor, tamanho, posição)	Pintura e decoração
JavaScript	Interação (botões que fazem algo)	Eletricidade e encanamento

HOJE

A gente vai mexer só com **HTML e CSS**. Dá pra fazer site profissional só com essas duas — e é o melhor jeito de começar.

— CAPÍTULO 03 • SETUP

As três ferramentas.

Tudo gratuito. Nada pra instalar além do editor.

01 • EDITOR

Visual Studio Code

Um bloco de notas melhorado, feito pra escrever código. Colore as palavras, ajuda a completar, avisa quando você erra.

`code.visualstudio.com`

02 • NAVEGADOR

O que você já usa

Chrome, Safari, Firefox, Edge — tanto faz. A gente vai abrir o site dando dois cliques no arquivo HTML.

03 • PASTA

`meu-site`

Crie uma pasta no seu computador chamada **meu-site**. Dentro dela vai todo o código que você vai escrever hoje.

DICA

Crie essa pasta na **Área de Trabalho** — fica fácil de achar.

BAIXE OS ARQUIVOS

Pegue o material do curso em `minicursoseuprimeirosite.leoad.com.br`. Lá tem a **versão pronta de cada etapa** — se você travar em algum ponto da construção, abra a pasta daquela etapa e segue daí, sem atrasar.

Tudo o que você precisa cabe em três coisas — e nenhuma delas custa nada.

— ETAPA PRÁTICA

Etapa 01 / 07

01

HTML cru.

Ter uma página HTML funcionando no navegador, mesmo que ainda esteja feia.

TEMPO	ARQUIVO	FICA
~15 minutos	index.html	Funcional, feio

O que é HTML

HTML é uma linguagem de **tags**. Cada tag é um pedaço de texto entre `<` e `>` que diz pro navegador o que aquele conteúdo é.

HTML

EXEMPLO

```
<p>Este é um parágrafo.</p>
```

A tag `<p>` é "abre parágrafo". A tag `</p>` é "fecha parágrafo". O texto no meio é o conteúdo. Quase toda tag funciona em par: uma pra abrir, uma pra fechar.

As tags que vamos usar hoje

TAG	PRA QUE SERVE
<code><h1></code> a <code><h3></code>	Títulos (h1 = mais importante)
<code><p></code>	Parágrafo de texto
<code><a href></code>	Link clicável
<code></code>	Imagem
<code></code> + <code></code>	Lista com itens
<code><div></code>	Caixa genérica pra agrupar coisas
<code><header></code>	Topo da página
<code><section></code>	Pedaço de conteúdo
<code><footer></code>	Final da página
<code><blockquote></code>	Citação (frase de outra pessoa)

Estrutura mínima de toda página HTML

```
HTML INDEX.HTML

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Título que aparece na aba</title>
</head>
<body>
  <!-- aqui vai o conteúdo visível -->
</body>
</html>
```

`<!DOCTYPE html>`

Avisa que esta é uma página HTML moderna.

`<html lang="pt-BR">`

Envolve tudo. `lang` diz o idioma.

`<head>`

Informações invisíveis pro navegador.

`<meta charset="UTF-8">`

Faz acentos e ç funcionarem.

`<title>`

Texto da aba do navegador.

`<body>`

O que aparece na tela.

Vamos lá: crie o **index.html**

- 01 Abra o VS Code.
- 02 Vá em **File** → **Open Folder** e abra sua pasta `meu-site`.
- 03 Vá em **File** → **New File** e salve como `index.html`.
- 04 Cole a estrutura mínima da página anterior.
- 05 Abre o arquivo no navegador (dois cliques no arquivo na sua pasta) e veja a aba mudar com o título.

PRONTO

Você fez seu primeiro arquivo HTML.

Preencha com a estrutura do seu cartão

Dentro do `<body>` vamos colocar as grandes seções do cartão: cabeçalho (hero), sobre, serviços, trabalhos, contato e rodapé.

HTML

INDEX.HTML · <BODY> · PARTE 1/2

```
<header>
  <p>Cartão digital, 2026</p>
  
  <h1>Seu Nome.</h1>
  <p>Sua Profissão</p>
  <div>
    <p>Baseado em: Sua Cidade, BR</p>
    <p>Foco: 0 que você faz</p>
    <p>Idiomas: PT</p>
  </div>
  <p><a href="#contato">Entre em contato -</a></p>
  <blockquote>"Uma frase forte." – Quem disse</blockquote>
</header>

<section>
  <h2>Sobre.</h2>
  <p>Um parágrafo curto sobre você.</p>
</section>

<section>
  <h2>0 que eu ofereço.</h2>
  <ul>
    <li><h3>Serviço 01</h3><p>Descrição curta.</p></li>
    <li><h3>Serviço 02</h3><p>Descrição curta.</p></li>
    <li><h3>Serviço 03</h3><p>Descrição curta.</p></li>
  </ul>
</section>
```

HTML

INDEX.HTML · <BODY> · PARTE 2/2

```
<section>
  <h2>Trabalhos.</h2>
  <ul>
    <li>
      
      <h3>Trabalho · 01</h3>
      <p>Uma frase sobre o trabalho.</p>
    </li>
    <!-- repete pros outros trabalhos -->
  </ul>
</section>

<section id="contato">
  <h2>Contato.</h2>
  <p>E-mail: <a href="mailto:seu@email.com">seu@email.com</a></p>
</section>

<footer>
  <p>© 2026, Seu Nome</p>
</footer>
```

Salve (`Ctrl+S` no Windows, `Cmd+S` no Mac). Atualize a página no navegador (`F5`).

Pronto. Você tem o esqueleto do site. Feio? Sim. Mas funciona.

CONFIRA

No material que você baixou, `etapas/v1-htm1-cru/` tem essa etapa pronta.

— ETAPA PRÁTICA

Etapa 02 / 07

02

Conectando o CSS.

Dar cor, fonte e respiro à página.

TEMPO	ARQUIVO	FICA
~25 minutos	style.css	Com cara

O que é CSS

CSS é a linguagem que controla o visual do site. Você escreve regras tipo “todo parágrafo tem texto cinza” ou “todo título usa a fonte tal”.

CSS

ESTRUTURA DE UMA REGRA

```
seletor {  
  propriedade: valor;  
}
```

CSS

EXEMPLO

```
p {  
  color: red;  
}
```

Isso quer dizer: “todo `<p>` da página tem cor vermelha”.

Crie o arquivo **style.css**

A organização que vamos usar:

```
ESTRUTURA MEU-SITE/  
  
meu-site/  
├── index.html  
├── assets/  
│   ├── css/  
│   └── style.css
```

- 01 No VS Code, crie a pasta `assets`.
- 02 Dentro dela, crie outra pasta `css`.
- 03 Dentro de `css`, crie o arquivo `style.css`.

Conecte o CSS ao HTML

Dentro do `<head>` do `index.html`, adicione:

```
HTML INDEX.HTML • <HEAD>  
  
<link rel="stylesheet" href="assets/css/style.css">
```

Adicione as fontes do Google

Ainda no `<head>`, antes do `<link>` do CSS:

```
HTML INDEX.HTML • <HEAD>  
  
<link rel="preconnect" href="https://fonts.googleapis.com">  
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
  
<!-- uma família por linha: mais fácil de ler e de mexer -->  
<link rel="stylesheet" href="https://fonts.googleapis.com/css2?  
family=Plus+Jakarta+Sans:wght@500;700;800&display=swap">  
<link rel="stylesheet" href="https://fonts.googleapis.com/css2?  
family=Inter:wght@400;500;600&display=swap">  
<link rel="stylesheet" href="https://fonts.googleapis.com/css2?  
family=JetBrains+Mono:wght@400;500&display=swap">
```

São **3 famílias: Plus Jakarta Sans** (títulos), **Inter** (texto), **JetBrains Mono** (rótulos pequenos).

Escreva o CSS básico

Cole isto dentro do `style.css`:

```
CSS ASSETS/CSS/STYLE.CSS

/* Variáveis: define cores e fontes em um lugar só.
   Mudou aqui, mudou no site inteiro. */
:root {
  --paper:      #ece6db;
  --paper-2:   #e3dcce;
  --ink:       #1c1a17;
  --ink-soft:  #2b2925;
  --muted:    #6e6859;
  --accent:    #7a2018;
  --accent-2: #a13328;
  --font-display: 'Plus Jakarta Sans', sans-serif;
  --font-text:  'Inter', sans-serif;
  --font-mono:  'JetBrains Mono', monospace;
}

/* Reset básico: zera margens estranhas dos navegadores. */
* { box-sizing: border-box; }

body {
  margin: 0;
  font-family: var(--font-text);
  font-size: 16px;
  line-height: 1.65;
  color: var(--ink);
  background-color: var(--paper);
}

img { max-width: 100%; display: block; }
a { color: inherit; text-decoration: none; }

h1, h2, h3 {
  font-family: var(--font-display);
  font-weight: 800;
  letter-spacing: -0.025em;
  line-height: 1.1;
  margin: 0;
}
```

Salve. Atualize o navegador. A página ainda está em coluna vertical, mas agora tem fundo cor de papel, texto escuro e fontes bonitas.

Entendendo o que aconteceu

:root

Lugar onde a gente define variáveis (cores, fontes).

var(--paper)

Usa a variável `--paper` definida lá em cima.

Todos os elementos da página.

box-sizing: border-box

Faz a matemática de tamanhos ficar intuitiva (sem isso, tamanho + borda + padding viram contas chatas).

letter-spacing: -0.025em

Aperta as letras dos títulos pra ficarem mais elegantes.

line-height: 1.65

Espaço vertical entre linhas de texto. Mais alto = mais respiro.

Variáveis são o coração do CSS moderno. Você define uma cor uma vez, usa em 50 lugares. Quer trocar a cor? Muda em um lugar só.

CONFIRA

No material que você baixou, `etapas/v2-css-conectado/` tem essa etapa pronta.

— ETAPA PRÁTICA

Etapa 03 / 07

03

Cabeçalho e Sobre.

Transformar o cabeçalho (hero) e a seção Sobre em algo profissional.

TEMPO	FOCO	FICA
~30 minutos	Classes & flex	Profissional

Adicionando classes ao HTML

Classe é uma “etiqueta” que a gente põe num elemento HTML pra dar um nome a ele. Aí no CSS a gente fala “estilize todo elemento com a classe tal”.

No HTML:

```
HTML                                     CLASSE
<header class="hero">
  ...
</header>
```

No CSS — ponto antes do nome:

```
CSS                                     CLASSE
.hero {
  background-color: var(--ink);
}
```

Helpers reutilizáveis

Antes de estilizar o hero, vamos criar 3 ajudantes que vamos reutilizar no site inteiro:

CSS

STYLE.CSS · HELPERS

```
.container {
  max-width: 1100px;
  margin: 0 auto;
  padding: 0 24px;
}

.label {
  font-family: var(--font-mono);
  font-size: 11px;
  letter-spacing: 0.16em;
  text-transform: uppercase;
  color: var(--muted);
}

.num {
  font-family: var(--font-mono);
  font-size: 12px;
  color: var(--accent);
  letter-spacing: 0.08em;
}

.dot { color: var(--accent-2); }
```

max-width + margin: 0 auto

Limita a largura e centraliza.

text-transform: uppercase

Deixa tudo MAIÚSCULO.

letter-spacing

Espaça as letras.

Atualize o HTML do cabeçalho

HTML

INDEX.HTML · <HEADER>

```
<header class="hero">
  <div class="container">
    <div class="hero-top">
      <span class="label">Cartão digital, 2026</span>
      <span class="mark">X</span>
    </div>
    
    <h1>Seu Nome<span class="dot">.</span></h1>
    <p class="role">Sua Profissão</p>

    <div class="hero-meta">
      <div>
        <span class="label">Baseado em</span>
        <strong>Sua Cidade, BR</strong>
      </div>
      <div>
        <span class="label">Foco</span>
        <strong>0 que você faz</strong>
      </div>
      <div>
        <span class="label">Idiomas</span>
        <strong>PT</strong>
      </div>
    </div>

    <a class="cta" href="#contato">Entre em contato →</a>
    <p class="thought">"Uma frase forte." <span>Quem disse</span></p>
  </div>
</header>
```

Estilize o cabeçalho

CSS

STYLE.CSS · HERO

```
.hero {
  background-color: var(--ink);
  color: var(--paper);
  padding: 80px 0 100px;
}

.hero-top {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 60px;
}

.avatar {
  width: 96px; height: 96px;
  border-radius: 50%;
  border: 2px solid var(--paper);
  object-fit: cover;
  margin-bottom: 28px;
}

.hero h1 {
  font-size: 64px;
  color: var(--paper);
  letter-spacing: -0.035em;
  margin-bottom: 16px;
}

.hero-meta {
  display: flex;
  flex-wrap: wrap;
  gap: 32px;
  padding: 24px 0;
  border-top: 1px solid rgba(236, 230, 219, 0.15);
  border-bottom: 1px solid rgba(236, 230, 219, 0.15);
}

.cta {
  display: inline-block;
  background-color: var(--accent);
  color: var(--paper);
  padding: 16px 24px;
  font-weight: 700;
  transition: background-color 0.2s;
}

.cta:hover { background-color: var(--accent-2); }
```

Entendendo o CSS do cabeçalho

display: flex

Põe os filhos lado a lado.

justify-content: space-between

Empurra os filhos pras pontas.

gap

Espaço entre os filhos do flex.

border-radius: 50%

Transforma quadrado em círculo.

rgba(...)

Cor com transparência (o quarto valor é opacidade).

transition

Faz a mudança ser suave em vez de instantânea.

Estilize a seção Sobre

HTML

INDEX.HTML · SOBRE

```
<section class="block" id="sobre">
  <div class="container">
    <div class="block-head">
      <h2>Uma frase forte sobre você.</h2>
      <span class="num">01 / 04</span>
    </div>
    <p class="lead">Um parágrafo de apoio em 2 a 4 linhas.</p>
  </div>
</section>
```

CSS

STYLE.CSS · BLOCK

```
.block {
  padding: 80px 0;
  border-bottom: 1px solid rgba(28, 26, 23, 0.12);
}

.block-head {
  display: flex;
  justify-content: space-between;
  align-items: baseline;
  gap: 24px;
  margin-bottom: 40px;
  flex-wrap: wrap;
}

.block-head h2 { font-size: 40px; max-width: 16ch; }

.lead {
  font-size: 17px;
  color: var(--ink-soft);
  max-width: 56ch;
  margin: 0 0 16px;
}
```

CONCEITO

max-width: 56ch limita a largura do parágrafo a aproximadamente 56 caracteres — bom pra leitura confortável.

CONFIRA

No material baixado: `etapas/v3-hero-e-sobre/`.

— ETAPA PRÁTICA

Etapa 04 / 07

04

Serviços, Trabalhos e Contato.

Montar as três seções de conteúdo com flexbox e grid.

TEMPO	FOCO	FICA
~35 minutos	Flex & Grid	Completo

Esta etapa cobre o resto do site:

- **Serviços** — lista vertical (com flexbox).
- **Trabalhos** — galeria em 3 colunas (com CSS Grid).
- **Contato** — grade 2×2 + redes sociais.

Serviços com flexbox

Cada serviço é um `` com flex horizontal: número à esquerda, conteúdo à direita.

HTML

INDEX.HTML · SERVIÇOS

```
<ul class="services">
  <li>
    <span class="svc-num"> 01</span>
    <div>
      <h3>Nome do Serviço</h3>
      <p>Descrição curta.</p>
    </div>
  </li>
</ul>
```

CSS

STYLE.CSS · SERVICES

```
.services { list-style: none; padding: 0; margin: 0; }

.services li {
  display: flex;
  gap: 24px;
  padding: 28px 0;
  border-top: 1px solid rgba(28, 26, 23, 0.15);
  align-items: flex-start;
}

.services li:last-child {
  border-bottom: 1px solid rgba(28, 26, 23, 0.15);
}

.svc-num {
  font-family: var(--font-mono);
  font-size: 12px;
  color: var(--muted);
  min-width: 50px;
}

.services h3 { font-size: 24px; margin-bottom: 8px; }
```

`list-style: none`

Tira as bolinhas da ``.

`:last-child`

“O último filho”, usado pra colocar borda no item final também.

CSS Grid em uma frase

`display: grid` cria uma grade. Você fala quantas colunas quer e o navegador organiza os filhos automaticamente.

CSS

GRID · 3 COLUNAS

```
.works {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 36px 28px;
}
```

“Faça uma grade com **3 colunas iguais**, com 36px de espaço vertical e 28px de espaço horizontal”.

HTML & CSS da galeria

HTML

INDEX.HTML · TRABALHOS

```
<div class="works">
  <div class="work">
    
    <h3>Trabalho · 01</h3>
    <p class="work-desc">Uma frase sobre o trabalho.</p>
    <div class="work-meta">
      <span>Ver detalhes →</span>
    </div>
  </div>
  <!-- repete pros outros -->
</div>
```

O CSS dos cards da galeria

CSS

STYLE.CSS · WORKS

```
.work img {  
  width: 100%;  
  aspect-ratio: 3 / 2;  
  object-fit: cover;  
  transition: transform 0.4s;  
}  
.work:hover img { transform: scale(1.02); }
```

aspect-ratio: 3 / 2

Mantém a proporção da imagem, independente do arquivo.

object-fit: cover

Corta a imagem pra preencher o espaço sem distorcer.

transform: scale(1.02)

Aumenta a imagem em 2% no hover.

Bloco de contato

HTML

INDEX.HTML · CONTATO

```
<section class="block" id="contato">
  <div class="container">
    <div class="block-head">
      <h2>Contato.</h2>
      <span class="num">04 / 04</span>
    </div>
    <div class="contact-grid">
      <div class="contact-cell">
        <span class="label">Telefone</span>
        <strong>+55 62 9 9999-9999</strong>
      </div>
    </div>
    <div class="socials">
      <a href="#">Instagram</a>
      <a href="#">LinkedIn</a>
    </div>
  </div>
</section>
```

CSS

STYLE.CSS · CONTACT

```
.contact-grid {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 24px;
}

.socials a {
  font-family: var(--font-mono);
  text-transform: uppercase;
  letter-spacing: 0.1em;
  border-bottom: 1px solid transparent;
  transition: border-color 0.2s;
}

.socials a:hover {
  color: var(--accent);
  border-color: var(--accent);
}
```

CONFIRA

No material baixado: [etapas/v4-galeria-e-contato/](#).

— ETAPA PRÁTICA

Páginas de detalhe.

Etapa 05 / 07

05

Criar uma 2ª página e ligar uma página na outra com links.

TEMPO	CONCEITO
~15 minutos	Links entre páginas

FICA
Várias páginas

Um site pode ter várias páginas

Cada página é um arquivo `.html` separado, na mesma pasta. Pra ir de uma pra outra, o link aponta pro **nome do arquivo** — igual o `#contato`, mas agora pra outro arquivo.

Transforme o card num link

No `index.html`, envolva o card do trabalho com um `<a>` apontando pra página de detalhe:

HTML

INDEX.HTML • CARD DO TRABALHO

```
<!-- o card inteiro vira um link -->
<a class="work" href="trabalho-1.html">
  
  <h3>Trabalho · 01</h3>
  <p class="work-desc">Uma frase sobre o trabalho.</p>
  <div class="work-meta"><span>Ver detalhes →</span></div>
</a>
```

Crie a página **trabalho-1.html**

Um arquivo novo, na mesma pasta, com a mesma estrutura: imagem de um lado, texto do outro, e um “← Voltar” que leva pro index.

```
HTML TRABALHO-1.HTML  
  
<main class="block">  
  <div class="container">  
    <a class="back" href="index.html#trabalhos">← Voltar</a>  
    <div class="detail">  
        
      <div class="detail-info">  
        <h1>Trabalho · 01</h1>  
        <p class="lead">Descrição do trabalho.</p>  
      </div>  
    </div>  
  </div>  
</main>
```

REPITA

Copie o arquivo pra `trabalho-2.html`, troque a imagem e o texto, e ligue o próximo card. Pronto: seu site tem várias páginas.

CONFIRA

No material que você baixou, `trabalho-1.html` a `trabalho-6.html` já estão prontas.

— ETAPA PRÁTICA

Etapa 06 / 07

06

Polimento e celular.

Fazer o site funcionar bem no celular.

TEMPO	CONCEITO	FICA
~15 minutos	Media query	Responsivo

Por que o site quebra no celular

O layout que a gente fez tem 3 colunas de galeria, 2 colunas de contato e nome em fonte de 64 pixels. No celular não cabe. A solução: **media query**.

Media query

Media query é um jeito de dizer pro CSS: *“estas regras só valem se a tela tiver largura menor que tantos pixels”*.

CSS

SINTAXE

```
@media (max-width: 720px) {  
  /* regras que só valem em tela menor que 720px */  
}
```

Adicione no final do **style.css**

CSS

STYLE.CSS • RESPONSIVO

```
@media (max-width: 720px) {
  .hero { padding: 60px 0 80px; }
  .hero h1 { font-size: 48px; }
  .hero-meta { gap: 20px; }

  .block { padding: 60px 0; }
  .block-head h2 { font-size: 30px; }

  /* Galeria vira coluna única no celular. */
  .works {
    grid-template-columns: 1fr;
    gap: 32px;
  }

  /* Contato também. */
  .contact-grid { grid-template-columns: 1fr; }

  .services li { gap: 16px; }
  .services h3 { font-size: 20px; }
}
```

1fr

“Uma fração” — uma coluna ocupando tudo.

sobrescreve

O resto dos estilos do desktop ainda valem; a media query só sobrescreve o que precisa.

Teste no celular

Aperte **F12** (ou **Cmd+Option+I** no Mac) pra abrir as ferramentas do navegador.

Procure o ícone de celular no topo. Clica nele e escolhe um modelo. Atualiza a página e veja.

CONFIRA

No material baixado: `etapas/v6-polish-e-responsivo/`.

— ETAPA PRÁTICA

Etapa 07 / 07

07

Publicando seu site.

Colocar seu site no ar com um endereço público — via Netlify Drop.

TEMPO	SERVIÇO	CUSTO
~15 minutos	Netlify Drop	R\$ 0,00

O que é o Netlify Drop

Netlify é uma empresa que hospeda sites estáticos de graça. O serviço “Drop” deixa você publicar arrastando uma pasta na tela, sem precisar criar conta nem instalar nada.

Passo a passo

- 01 Abra o navegador em `https://app.netlify.com/drop`.
- 02 Vá no seu computador e abra a pasta onde está o seu `meu-site`.
- 03 Arraste a pasta `meu-site` inteira pra cima da área cinza do Netlify.
- 04 Espere de 5 a 30 segundos.
- 05 Pronto. O Netlify te dá um endereço tipo `random-name.netlify.app`.

Esse endereço é público. Manda no zap da sua mãe.

Quero mudar o endereço pra algo melhor

Por padrão o endereço é aleatório. Pra mudar:

01 No painel do Netlify (depois que você publicou), procure **Site settings**
→ **Change site name**.

02 Coloque o nome que você quer (ex: `ana-lima.netlify.app`).

DOMÍNIO PRÓPRIO

Se quiser um domínio próprio (tipo `analima.com.br`), aí precisa comprar o domínio (Registro.br custa R\$ 40/ano pra `.com.br`) e configurar no Netlify. Mas isso é assunto pra depois.

Quero atualizar o site

Toda vez que você mudar algo, é só arrastar a pasta de novo no Netlify Drop. Ele substitui a versão antiga.

*Site no ar, endereço próprio, atualizado por arrastar e soltar.
Você acabou de fazer sozinho, do zero, o que muita gente contrata pronto.*

— CAPÍTULO 11 • PRÓXIMOS PASSOS

Se você curtiu e quer *ir além*.

01 • APRENDER MAIS

HTML & CSS

MDN Web Docs (pt-BR) — documentação mais confiável da web.

Origamid — cursos gratuitos no YouTube, muito bons.

freeCodeCamp — currículo completo e gratuito.

02 • EXPANDIR

Seu site

Mais páginas: crie outros `.html` e ligue com `<a href>`.

Formulário: Netlify Forms (gratuito).

Animações: estude `@keyframes` e `animation`.

03 • PRÓXIMA LINGUAGEM

JavaScript

Se quiser que seu site tenha botões que fazem coisa (abrir menu, mudar tema, validar formulário).

Comece pelo básico: variáveis, funções, e como mexer no HTML pelo JS.

— CAPÍTULO 12 · REFERÊNCIA

Glossário

rápido.

TERMO	O QUE É
HTML	Linguagem das tags, define a estrutura.
CSS	Linguagem que define o visual.
Tag	Pedaço entre <code>< ></code> que envolve conteúdo.
Classe	Nome que você dá a um elemento pra estilizar.
Seletor	Pedaço do CSS que escolhe o que estilizar.
Propriedade	O que você está estilizando (cor, tamanho...).
Valor	Como você quer estilizar (vermelho, 16px...).
Variável CSS	Valor que você guarda pra reutilizar.
Flexbox	Sistema de alinhar elementos em linha ou coluna.
Grid	Sistema de alinhar elementos em grade.
Aspect-ratio	Proporção entre largura e altura.
Media query	Regra CSS que só vale em certo tamanho de tela.
Hover	Quando o mouse passa em cima de algo.
Transition	Mudança suave de uma propriedade ao longo do tempo.
CTA	Call to Action — botão de ação principal.
Responsivo	Site que se adapta a diferentes telas.

— FIM

Até a próxima.

Apostila escrita por **Leo AD** (theleoad.com). Pode copiar, modificar e usar à vontade.

SOBRE

Mini-curso de HTML e CSS, 2h30, gratuito. Pra quem nunca abriu um editor de código na vida.

TIPOGRAFIA

Plus Jakarta Sans
Inter
JetBrains Mono
Fraunces (itálicos)

ÚLTIMA REVISÃO

Maio de 2026
Edição 01