

O Guia descomplicado Front e Back-end Do início ao código

```
127 </div>
128 </header>
129 <!-- Header -->
130
131 <!-- About -->
132
133 <section class="about" id="about">
134   <div class="container text-center">
135
136     <div class="row stats-row">
137       <div class="stats-col text-center col-md-3 col-sm-6">
138         <div class="circle">
139           <span class="stats-up" data-toggle="counter-up">1234</span> Customers
140         </div>
141       </div>
142
143       <div class="stats-col text-center col-md-3 col-sm-6">
144         <div class="circle">
145           <span class="stats-up" data-toggle="counter-up">1234</span> Projects
146         </div>
147       </div>
148
149       <div class="stats-col text-center col-md-3 col-sm-6">
150         <div class="circle">
151           <span class="stats-up" data-toggle="counter-up">1234</span> Support
152         </div>
153       </div>
154
155       <div class="stats-col text-center col-md-3 col-sm-6">
156         <div class="circle">
157           <span class="stats-up" data-toggle="counter-up">1234</span> Hard Workers
158         </div>
159       </div>
160     </div>
161
162     </div>
163   </section>
164   <!-- About -->
165   <!-- Parallax -->
166
167   <div class="block bg-primary block-pt-lg block-bg-overlay text-center" data-bg-img="img/parallax-bg.jpg" data-
168     data-toggle="parallax-bg">
```

por Felipe Alvim



O autor

Felipe Alvim é head de marketing e vendas com uma competência fundamental em Programação e desenvolvimento web, o que o diferencia no mercado.

Ele é um estrategista de crescimento que entende o código por trás da conversão, garantindo que as plataformas e sites não só pareçam bons, mas estejam tecnicamente otimizados.

Unindo sua base técnica à sua experiência em vendas e marketing

Olá,

**FUTURO
DESENVOLVEDOR!**

Introdução

Introdução

Ei, que bom ter você aqui! Eu sou seu guia nesta jornada. Se você sente curiosidade sobre "como a internet funciona" ou busca uma nova carreira, saiba: programar não é mágica, é lógica, e qualquer um pode aprender!

Não precisa ser um gênio ou ter um supercomputador. Apenas curiosidade e paciência. Eu sei que termos como HTML, JavaScript e Backend soam estranhos, mas meu papel é simplificá-los. Programar é dar instruções a uma máquina. Quando seu celular reage a um toque, um programador escreveu essa instrução.

Simples assim.

Antes de codar, vamos entender o "campo de batalha": a Web. Ela é uma rede gigante de computadores. Temos o Cliente (seu navegador, celular) e o Servidor (um computador distante que guarda os arquivos de um site).

Programar é solucionar a dor de um cliente. Antes de programar, converse!

A cada acesso, seu Navegador (Cliente) envia um pedido ao Servidor, que processa e devolve os códigos da página. Seu Navegador os transforma naquilo que você vê.

Básico!

Cliente pede, Servidor responde. Programação é escrever as regras dessa conversa.

VAMOS JUNTOS?



Capítulo 1



Frontend vs. Backend

Frontend vs. Backend:

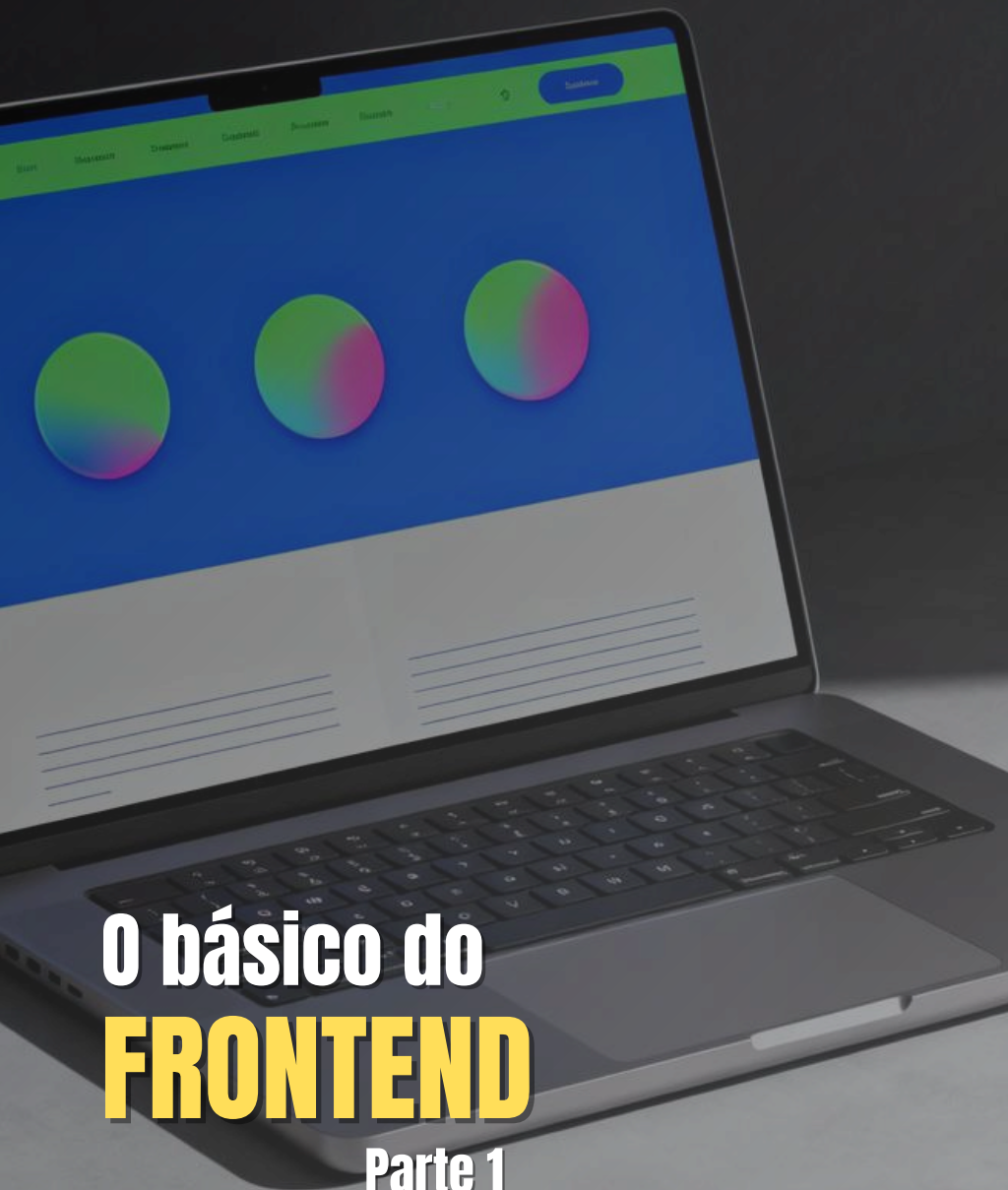
O desenvolvimento web tem dois times principais. Para entender, imagine um restaurante:

Frontend (O salão e o prato): É a parte que o cliente vê. A decoração, o garçom, a apresentação da comida. No site, é tudo que você vê e interage: botões, textos, imagens, layout. Ele roda no seu computador (lado do cliente) e foca na Experiência do Usuário (UX).

Backend (A cozinha e a lógica): É onde a mágica "real" acontece, mas longe dos olhos do cliente. É o cozinheiro, o estoque, a lógica para o prato ser preparado. No site, é tudo que acontece no Servidor: processar logins, guardar dados, fazer cálculos complexos. Roda num computador distante (lado do servidor) e foca na lógica de negócios e na segurança dos dados.

Um desenvolvedor pode ser Full Stack (dominar os dois lados), mas para começar, é bom escolher um foco. Entender essa divisão é crucial para sua jornada.

Capítulo 2



O básico do **FRONTEND**

Parte 1

HTML: O esqueleto da sua página

Dando Estrutura com Tags e Elementos

Se seu site fosse um corpo, o HTML (HyperText Markup Language) seria o esqueleto. Ele dá a estrutura e o sentido (semântica) de tudo.

O HTML é feito de tags (como `<h1>` para título, `<p>` para parágrafo). Elas são "etiquetas" que dizem ao navegador o que cada pedaço de conteúdo é. Um elemento é a tag de abertura, o conteúdo e a tag de fechamento. `<h1> </h1>`

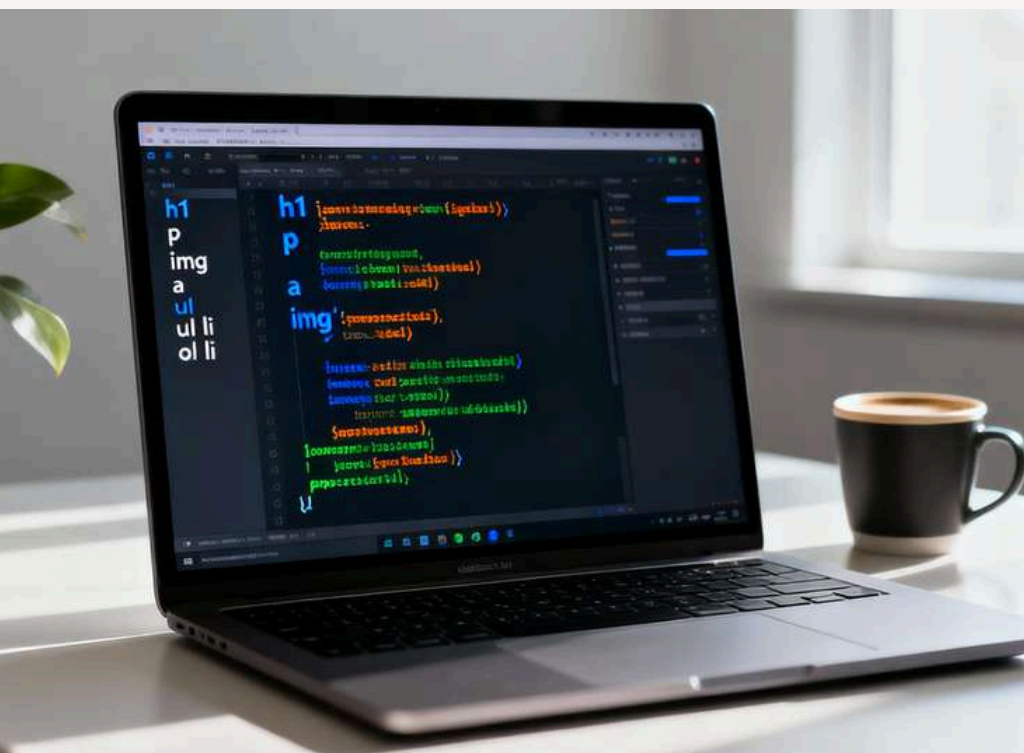
As tags básicas são:

- `<h1>` até `<h6>`: Títulos de diferentes níveis.
- `<p>`: Parágrafos de texto.
- ``: Para inserir imagens.
- `<a>`: Para criar links que te levam para outras páginas.
- `` / ``: Para listas não ordenadas.
- `` / ``: Para listas ordenadas

A Regra de ouro: O HTML tem que ser semântico. Isso significa usar a tag certa para o conteúdo certo.

Se algo é um título, use `<h1>` e não tente fazer o texto parecer um título com outra coisa. O HTML é sobre significado. A aparência?

Isso é trabalho para o próximo capítulo.



Capítulo 3

O básico do **FRONTEND**

Parte 2

CSS:

A roupa e a maquiagem da sua página

Pegou o esqueleto HTML?

Agora o CSS (Cascading Style Sheets) entra em cena para dar cor, fonte, posicionar elementos, definir a roupa e a maquiagem do seu site. O HTML diz o que é; o CSS diz como deve parecer.

Como funciona?

1. Seletores: Você "seleciona" qual elemento HTML quer estilizar.

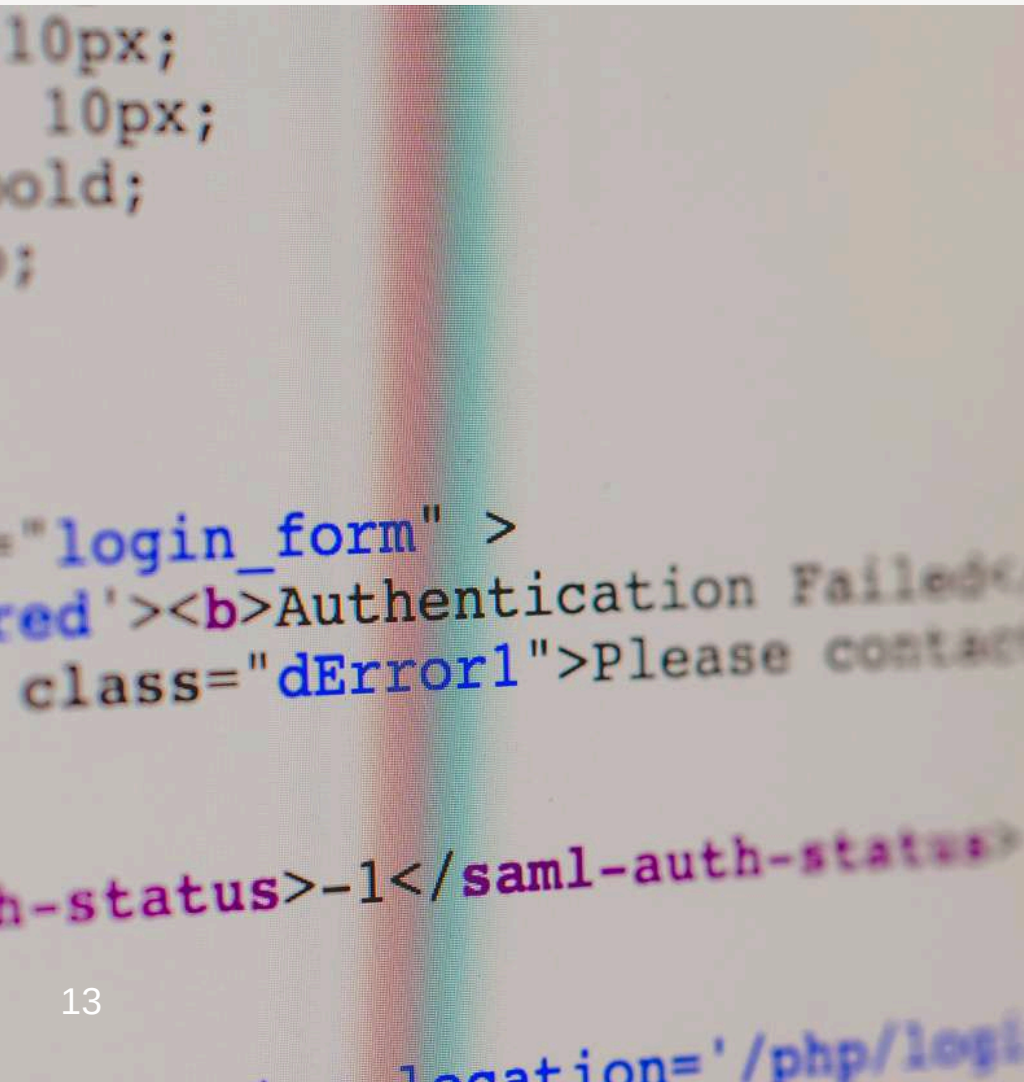
2. Propriedade e valor: Você escolhe o que quer mudar (a propriedade) e qual o novo estilo (o valor). Ex: color: blue; (cor azul); font-size: 16px; (tamanho da fonte).

3. O box model: essencial! No CSS, tudo é uma "caixa" invisível com:

- Conteúdo: O texto/imagem.
- Padding: Espaço interno da caixa.
- Border: A linha da borda.

O nome "**Cascading**" (cascata) se refere a como o CSS decide qual estilo aplicar se houver conflitos (ex: você define uma cor azul e depois vermelha para o mesmo elemento, qual vence?).

O CSS tem regras claras para isso.



Capítulo 4

O básico do **FRONTEND**

Parte 3

JavaScript:

Dando vida e ação ao seu site



Se o HTML é o esqueleto e o CSS é a maquiagem, o JavaScript (JS) é o sistema nervoso e o cérebro. É a única das três que é uma linguagem de programação de verdade, com lógica, variáveis e funções. É o JS que transforma uma página estática em uma aplicação interativa.

O que o JS faz:

Interatividade: Quando você clica em um botão e um menu se abre sem recarregar a página, isso é JS.

Validação: Se um formulário te avisa que você esqueceu de preencher um campo, isso é JS.

Modificação de Conteúdo: O JS pode mudar o HTML e o CSS "ao vivo" enquanto a pessoa usa o site.

Conceitos essenciais (programação básica):

1. Variáveis: São "caixinhas" na memória que guardam informações (textos, números, listas). Exemplo: `let nome = "Maria";`

2. Funções: São blocos de código que executam uma tarefa específica. Elas nos permitem reutilizar código. Exemplo: Uma função para calcular o valor do frete.

3. Condicionais (Lógica): O famoso `if/else` (Se/Senão). Permite que o código tome decisões. Se o usuário está logado, mostre o perfil, senão, mostre o botão de login.

4. Eventos: O JS fica "escutando" o que acontece na tela (um clique, o mouse passando por cima, o teclado digitando) e executa uma função quando um desses eventos acontece.

Dica: O JS é a sua porta de entrada para a lógica de programação. Domine ele no frontend e você terá uma base sólida para qualquer outra linguagem!

Capítulo 5

React



Vue



Angu

Turbinando o Frontend:
**BIBLIOTECAS E
FRAMEWORKS**

Acelerando com React, Vue e Angular

Você já aprendeu HTML, CSS e JavaScript puro (o "Vanilla JS"). Mas os grandes aplicativos da web (como redes sociais) usam algo mais:

Bibliotecas e Frameworks. Eles são como "kits de ferramentas" superpoderosos.

Bibliotecas (Ex: React): São coleções de funções prontas que facilitam sua vida. Você as chama quando precisa. O React, do Facebook, é super popular para criar componentes (peças reutilizáveis, como um botão de like).

Frameworks (Ex: Angular): São como um "esqueleto de casa" pronto. Eles definem as regras e a estrutura do seu projeto. Você se adapta a ele.

Por que usar?

Imagine construir uma casa. Usar só o "Vanilla JS" é como cortar a madeira e fazer os pregos do zero. Usar uma biblioteca/framework é como receber janelas prontas, vigas cortadas e uma betoneira.

Eles aceleram o desenvolvimento, organizam o código e facilitam a manutenção de apps grandes. Para conseguir um emprego Frontend hoje, dominar uma dessas ferramentas (especialmente React) é quase obrigatório.

```
state={
  products: storeProducts
}
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="cont
          <Title name="our
            <div className=
              <ProductCons
                {(value)
                  con
                }}
              </ProductCon
            </div>
          </div>
        </div>
      </div>
    </React.Fragment>
```

Capítulo 6

Backend: O Cérebro da Operação

Infraestrutura de Suporte



Backend

O CÉREBRO DA
OPERAÇÃO

O que o servidor faz e as linguagens da "cozinha"

Agora, saímos da tela do usuário e entramos na "cozinha" do nosso restaurante: o Servidor. O backend é o cérebro que processa as informações do frontend. Quando você clica em "Login", seu e-mail e senha vão para o Servidor.

O que o backend faz?

Roteamento: É o mapa que diz para onde cada pedido deve ir (ex: /perfil vai para o código que busca seus dados).

Lógica de negócios: São as regras do aplicativo. Se a senha estiver errada, recusa o login. Se o saldo é insuficiente, não transfere. Ele garante que as regras do seu negócio sejam cumpridas.

Segurança e autenticação: Verifica sua identidade (login) e autoriza acessos.

Gerenciamento de Dados: Conversa com o banco de dados (próximo capítulo) para salvar ou buscar informações.

O backend não se importa com a beleza do site, mas com a segurança, a lógica e o processamento correto dos dados.

Para construir o backend, temos várias "linguagens da cozinha":

PHP: Tradicional e muito usada (o WordPress é em PHP).

Python: Limpa e legível, ótima para iniciantes, também usada em IA.

Node.js (JavaScript): Permite usar JavaScript também no servidor, o que é ótimo para Full Stack.

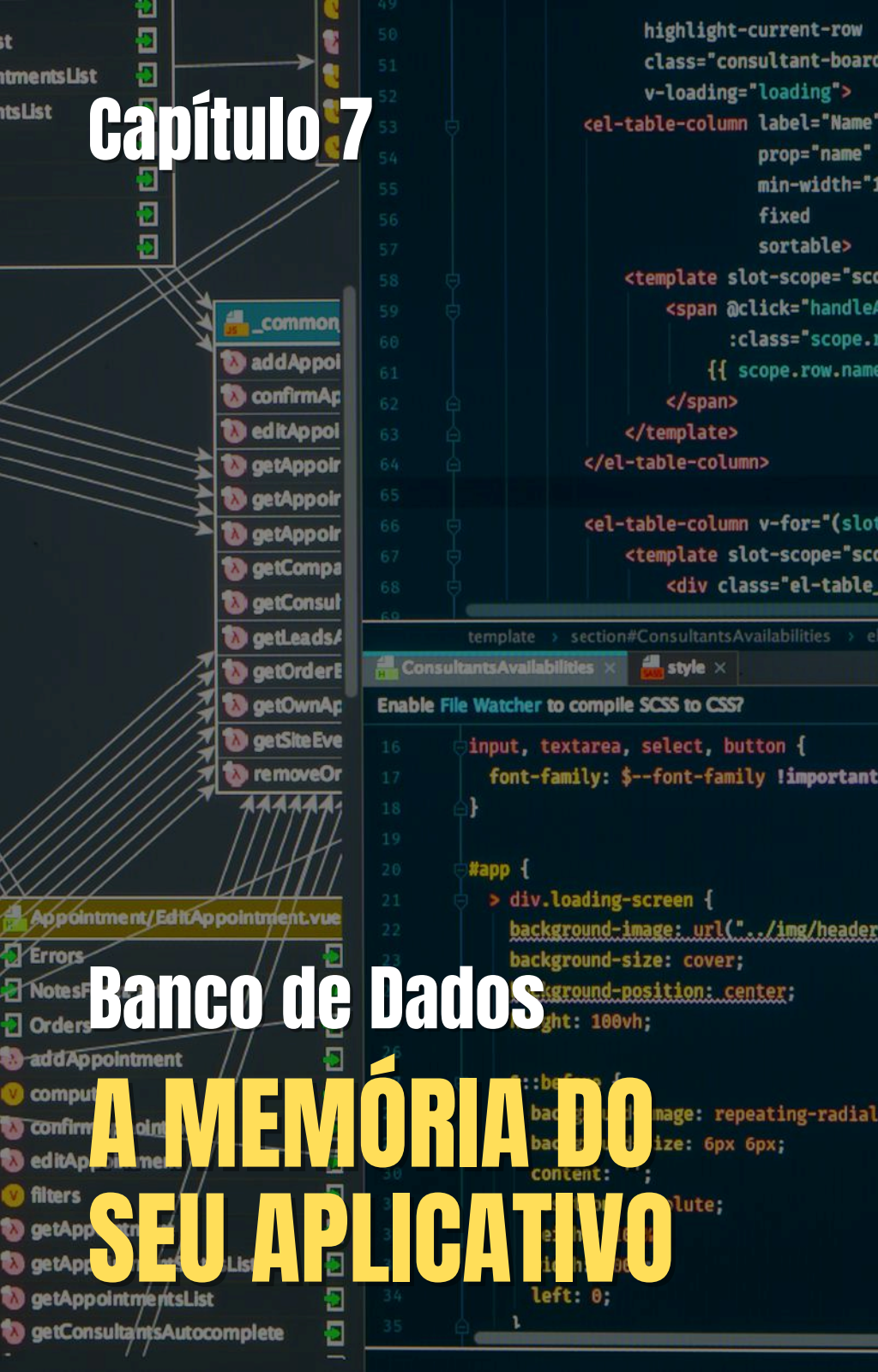
Java / C#: Robustas, usadas em grandes sistemas.

Escolha uma, aprenda a lógica, e você poderá migrar para outras!

Capítulo 7

Banco de Dados

A MEMÓRIA DO
SEU APLICATIVO



Onde os dados são guardados e como se conversa com eles

Seu aplicativo precisa de uma memória de longo prazo para guardar tudo: nomes de usuários, histórico de compras, seus posts. Essa é a função do banco de dados (DB). O backend é quem gerencia o acesso a essa memória. Pense no DB como um armário de arquivos superorganizado.

Tipos de bancos de dados:

1. SQL (Relacionais): O mais comum (Ex: MySQL, PostgreSQL). Organiza os dados em tabelas (como planilhas, com linhas e colunas) que se relacionam entre si.

A linguagem para "conversar" com ele é o SQL (Structured Query Language).

2. NoSQL (Não relacionais): Mais flexíveis (Ex: MongoDB). Guardam dados de formas variadas (documentos, grafos), ótimos para dados que mudam muito.

Capítulo 8



Sua Jornada a Partir de
AGORA

Dicas práticas e onde codar de verdade

Parabéns!

Você já tem um mapa mental completo da programação web. Mas conhecimento sem prática é só teoria.

Seus Próximos passos práticos:

1. Escolha um Lado (por enquanto):

Quer ver resultados rápidos? Vá para o Frontend (HTML, CSS, JavaScript). Gosta de lógica e resolver problemas? Foco na Lógica de Programação (variáveis, funções, etc.) com JavaScript ou Python.

2. Ferramenta essencial: VS Code: Baixe o Visual Studio Code, um editor de código gratuito e o mais usado. É onde a mágica acontece.

3. Crie um projeto (O segredo da prática): Não só tutoriais! Crie algo seu, por mais simples que seja (um site de currículo, uma calculadora).

Conclusão



O código está lhe
CHAMANDO

Uau!

Chegamos ao fim da nossa conversa formal, mas não do nosso papo sobre programação. Se você chegou até aqui, já merece um baita parabéns!

Pense bem: há poucas páginas, termos como HTML, CSS, JavaScript, Backend e Banco de Dados podiam soar como outro idioma. Agora, você não só entende o que são, mas também como eles se encaixam para construir tudo que vemos (e não vemos) na internet.

Lembre-se do que eu te disse no início: Programar não é mágica, é lógica. E você acaba de desvendar grande parte dessa lógica. Não é pouca coisa!

Este e-book foi o seu ponto de partida, o seu mapa inicial. Ele te deu a visão geral, os nomes dos lugares e as principais rotas.

Mas o verdadeiro aprendizado, a experiência que vai te transformar em um(a) desenvolvedor(a) de verdade, acontece colocando a mão na massa.

Experimente: Não tenha medo de quebrar as coisas. É assim que a gente aprende.

Crie: Comece com projetos pequenos, simples, mas que sejam seus.

Pesquise: O Google será seu melhor amigo. As respostas estão lá fora, e saber encontrá-las é uma habilidade de ouro.

Conecte-se: A comunidade de desenvolvedores é incrível, cheia de gente disposta a ajudar (e a aprender com você também!).

Seja para uma transição de carreira ou para a sua primeira grande paixão profissional, o desenvolvimento web é um universo em constante expansão, com oportunidades de sobra para quem tem curiosidade e vontade de aprender.

O seu futuro desenvolvedor(a) está te esperando. Vá em frente e comece a construir! Mal posso esperar para ver o que você vai criar. Com carinho e muitos códigos,

Felipe Alvim



(85) 9.9826.1414



nexusinnova.com.br



contato@nexusinnova.com.br