

Aula 1

O que é desenvolvimento web: Front-end vs Back-end:

O desenvolvimento web envolve a criação e manutenção de sites e aplicações acessíveis via navegadores. Ele se divide, principalmente, em duas áreas: **front-end** e **back-end**, que trabalham juntas para criar experiências completas na web.

Front-end (Desenvolvimento do Lado do Cliente)

O **front-end** é a parte visível de um site ou aplicativo — tudo com o que o usuário interage diretamente. É responsável pelo **design**, **layout**, e **comportamento** visual de uma página.

Principais tecnologias:

- **HTML (HyperText Markup Language)** – estrutura do conteúdo.
- **CSS (Cascading Style Sheets)** – estilização visual (cores, fontes, layout).
- **JavaScript** – interatividade (animações, menus dinâmicos, formulários interativos).
- **Frameworks/bibliotecas populares:** React, Vue.js, Angular, SASS.

Responsabilidades:

- Tornar a interface responsiva (funcionar bem em diferentes dispositivos).
 - Garantir boa usabilidade.
 - Implementar designs vindos do time de UI/UX.
-

Back-end (Desenvolvimento do Lado do Servidor)

O **back-end** é a parte “invisível” do sistema, que roda no servidor e lida com **lógica de negócios**, **banco de dados**, **autenticação**, e **comunicação com o front-end**.

Principais tecnologias:

- **Linguagens de programação:** Python, Java, PHP, Ruby, Node.js.
- **Frameworks populares:** Django (Python), Laravel (PHP), Express (Node.js), Spring (Java).
- **Banco de dados:** MySQL, PostgreSQL, MongoDB.
- **APIs REST e GraphQL** para comunicação com o front-end.

Responsabilidades:

- Processar e armazenar dados do usuário.
- Realizar autenticação e controle de acesso.
- Criar e gerenciar APIs.
- Garantir performance e segurança dos dados.

Diferenças principais

| Aspecto | Front-end | Back-end |
|------------------|-----------------------------------|-------------------------------------|
| Visibilidade | Visível ao usuário (interface) | Invisível ao usuário (servidor) |
| Linguagens | HTML, CSS, JavaScript | Python, Java, PHP, Node.js, etc. |
| Responsabilidade | Aparência e interação | Lógica, dados e segurança |

| | | |
|----------|-------------------------|-------------|
| Execução | No navegador do usuário | No servidor |
|----------|-------------------------|-------------|

Como Funciona a Web

A web (World Wide Web) é um sistema de comunicação baseado na arquitetura **cliente-servidor**, que utiliza o protocolo **HTTP** para transmitir informações.

Cliente (Navegador)

O **cliente** é o dispositivo (computador, celular, tablet) que o usuário utiliza para acessar a web, geralmente por meio de um **navegador** (Chrome, Firefox, Safari, etc.).

O que ele faz:

- Envia requisições (requests) para o **servidor**.
 - Recebe e exibe as respostas (respostas geralmente são páginas HTML, imagens, arquivos, etc).
-

Servidor

O **servidor** é um computador configurado para receber requisições da web e responder a elas. Ele armazena os sites e aplicações web.

O que ele faz:

- Recebe a requisição do cliente.
 - Processa a informação (pode consultar um banco de dados, por exemplo).
 - Envia a resposta adequada ao cliente.
-

HTTP (Hypertext Transfer Protocol)

O **HTTP** é o protocolo que define **como** as mensagens são formatadas e transmitidas na web, e **como os servidores e navegadores devem responder** a essas mensagens.

Características do HTTP:

- Baseado em **requisições e respostas**.
- Cada vez que você acessa uma página, o navegador faz uma ou mais requisições HTTP.
- É **sem estado (stateless)**: cada requisição é independente da anterior.

Tipos comuns de métodos HTTP:

- **GET**: Solicita dados (ex: acessar uma página).
- **POST**: Envia dados (ex: formulário de login).
- **PUT**: Atualiza dados.
- **DELETE**: Remove dados.

Exemplo do Ciclo Web

1. Você digita **www.exemplo.com** no navegador.
2. O navegador envia uma **requisição HTTP GET** para o servidor desse site.
3. O servidor processa a requisição e envia de volta uma **resposta HTTP** com o conteúdo da página.
4. O navegador interpreta esse conteúdo (HTML, CSS, JS) e exibe a página para você.

Componentes Adicionais Envolvidos

- **DNS (Domain Name System):** Converte o nome do site (www.google.com) para o endereço IP real do servidor.
- **HTTPS:** Versão segura do HTTP com criptografia (SSL/TLS).
- **Cookies e sessões:** Usados para manter informações entre requisições (ex: manter usuário logado).

Ferramentas Básicas para Desenvolvimento Web

Navegador (Browser)

O navegador é essencial para testar e depurar páginas web. Ele interpreta HTML, CSS e JavaScript e mostra como sua aplicação aparece para o usuário final.

☀️ Recomendado:

- **Google Chrome**
 - ☒ Amplo suporte a ferramentas de desenvolvedor.
 - ☒ Atualizações constantes.
- **Mozilla Firefox Developer Edition**
 - ☒ Ferramentas avançadas de inspeção CSS/JS.
- **Microsoft Edge**
 - ☒ Baseado no mesmo motor do Chrome (Chromium), com boas dev tools.

Ferramentas internas (DevTools):

- Inspecionar elementos da página (HTML/CSS)
- Console para testes de JavaScript

- Rede (Network) para analisar carregamento de arquivos
 - Armazenamento (Cookies, LocalStorage)
 - Depurador (debug) de código JS
-

Editor de Código: Visual Studio Code (VS Code)

O **VS Code** é um dos editores mais populares entre desenvolvedores web por ser leve, personalizável e cheio de recursos úteis.

Extensões úteis:

- **Live Server**
→ Cria um servidor local que atualiza a página automaticamente quando você salva o arquivo.
 - **Prettier**
→ Formata automaticamente seu código (HTML, CSS, JS, etc.) de forma padronizada.
 - **ESLint**
→ Detecta e corrige problemas no código JavaScript conforme padrões.
 - **HTML CSS Support**
→ Melhora a auto-completação de classes CSS dentro de arquivos HTML.
 - **Path Intellisense**
→ Sugere caminhos automaticamente para arquivos e imagens.
 - **Rainbow Brackets**
→ Colore pares de colchetes para facilitar leitura de código.
 - **Indent Rainbow**
→ Colore frente de divs para facilitar leitura de código.
-

Outras ferramentas que você pode usar junto:

- **Git e GitHub** – Para controle de versão.
- **Terminal integrado no VS Code** – Executa comandos sem sair do editor.

Estrutura Básica de um Documento HTML

Todo arquivo HTML segue uma estrutura padrão. Veja abaixo:

```
html                                                                    Copiar  Editar

<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Título da Página</title>
  </head>
  <body>
    <h1>Olá, mundo!</h1>
    <p>Esse é meu primeiro site em HTML.</p>
  </body>
</html>
```

| | |
|------------------------------------|---|
| <code><head></code> | Contém metadados , título, links de CSS, scripts, etc. |
| <code><meta charset></code> | Define a codificação de caracteres (UTF-8 = suporta acentuação). |
| <code><meta viewport></code> | Torna a página responsiva em celulares/tablets. |
| <code><title></code> | Define o título da aba do navegador. |
| <code><body></code> | Onde vai o conteúdo visível da página (texto, imagens, etc). |

Atalhos Úteis no VS Code para HTML

📌 Emmet (já vem ativado no VS Code):

Você pode digitar **!** e pressionar **Tab** → Gera a estrutura básica do HTML automaticamente.

Outros exemplos:

- `ul>li*3` → Gera:

```
html

<ul>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

- `div.container>h1{Título}+p{Texto}` → Gera:

```
html

<div class="container">
  <h1>Título</h1>
  <p>Texto</p>
</div>
```

Referências

Mozilla Developer Network (MDN Web Docs)

- <https://developer.mozilla.org>

W3Schools – Tutoriais e documentação básica de HTML, CSS, JS e linguagens back-end

- <https://www.w3schools.com>

GeeksForGeeks

- <https://www.geeksforgeeks.org/frontend-vs-backend/>

Mozilla Developer Network (MDN Web Docs)

- <https://developer.mozilla.org/en-US/docs/Web/HTTP>

How the Web Works – MDN

- https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works

FreeCodeCamp – How the Internet Works

- <https://www.freecodecamp.org/news/how-does-the-internet-work/>

Visual Studio Code – Site oficial

- <https://code.visualstudio.com/>

VS Code Extensions Marketplace

- <https://marketplace.visualstudio.com/vscode>

Chrome DevTools Documentation – MDN

- <https://firefox-source-docs.mozilla.org/devtools-user/index.html>

MDN Web Docs (HTML básico)

- <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

W3Schools HTML Tutorial

- <https://www.w3schools.com/html/>

Emmet Docs (atalhos)

- <https://www.w3schools.com/html/>