

# WEB E DESENVOLVIMENTO FRONTEND

Prof. Fabricio Freire, MSc in Cybersecurity

# AGENDA

- O que é a Web?
- Arquitetura Cliente-Servidor
- Frontend vs Backend
- Linguagens do Frontend
- Ferramentas para Desenvolvimento Frontend
- Frameworks: Usar ou não?
- Dicas

# O QUE É A WEB?

- Internet ≠ Web
  - Internet: rede mundial de computadores (1960s)
  - Web (WWW): sistema de hipertexto sobre a internet (1989-1991)
- Web = Documentos interconectados via hyperlinks
  - Documentos HTML (hipertexto)
  - Links formando uma "teia" de informações
  - Base para serviços como Google, Facebook e outros

# ARQUITETURA CLIENTE- SERVIDOR

## Cliente (Frontend):

- Navegador web do usuário
- Interpreta e renderiza documentos HTML, CSS, JavaScript
- Interface com o usuário

## Servidor (Backend):

- Processo em execução aguardando requisições
- Processa solicitações e retorna respostas
- Pode interagir com bancos de dados

Fonte da imagem:



# COMUNICAÇÃO FRONTEND E BACKEND

- Comunicação via protocolo HTTP
  - Hypertext Transfer Protocol
- Sistema de endereçamento DNS
  - Traduz endereços como `www.google.com` para IPs numéricos
- APIs (Application Programming Interfaces)
  - Regras para comunicação entre sistemas
  - Blocos prontos de construção de código
  - Conjunto de endpoints para solicitar/receber dados

# FRONTEND: PROGRAMAÇÃO OU DESIGN?

- Não é apenas design
  - Não é só "desenhar telas"
  - Envolve muita programação
  - Transforma designs em interfaces dinâmicas e funcionais
- Principais responsabilidades:
  - Criar interfaces para interação do usuário
  - Possibilitar entrada e saída de dados
  - Tornar a experiência intuitiva e agradável



# LINGUAGENS DE FRONTEND

- HTML (Linguagem de Marcação de Hipertexto)
  - Estrutura do documento
  - Como paredes de uma casa
- CSS (Folhas de Estilo em Cascata)
  - Estilização e formatação
  - Como pintura e acabamentos da casa
- JavaScript
  - Programação e interatividade
  - Como sistemas elétricos e hidráulicos da casa

# HTML

- **HyperText Markup Language**
- Linguagem de marcação (não de programação)
- Subconjunto do XML
- Usa tags para criar documentos
- Elementos: textos, mídias e links
- Estrutura básica da página



# HTML

- **HyperText Markup Language**
- Linguagem de marcação (não de programação)
- Subconjunto do XML
- Usa tags para criar documentos
- Elementos: textos, mídias e links
- Estrutura básica da página

```
<h1>Título da Página</h1>  
<p>Este é um parágrafo com <a  
href="link.html">link</a>.</p>
```

# CSS

- **Cascading Style Sheets**
- Linguagem de estilização
- Define regras de formatação
- Controla aparência dos elementos HTML
- Cores, tamanhos, posicionamentos, etc.

```
h1 {  
    color: blue;  
    font-size: 24px;  
}  
p {  
    margin: 10px;  
    line-height: 1.5;  
}
```

# JAVASCRIPT

- Linguagem de programação interpretada pelo navegador
- Torna as páginas dinâmicas
- Manipula elementos HTML e estilos CSS
- Permite interatividade e lógica no cliente
- Sintaxe baseada na linguagem C

```
document.getElementById("botao").addEventListener("click", function() {  
    alert("Botão foi clicado!");  
    document.getElementById("resultado").innerHTML = "Processado!";  
});
```

# DOCUMENT OBJECT MODEL (DOM)

- Hierarquia de objetos JavaScript que casa com o HTML
- Permite acesso e manipulação dinâmica dos elementos
- Base para interatividade nas páginas

```
// Acessar e modificar elementos
const titulo = document.querySelector("h1");
titulo.textContent = "Novo título";
titulo.style.color = "red";
```

# FERRAMENTAS PARA DESENVOLVIMENTO FRONTEND

## 1. VS Code (Microsoft)

- Mais popular atualmente
- Código aberto, extensível
- Suporte ao Git e servidor web para testes

## 2. Sublime Text

- Leve e multiplataforma
- Muito utilizado por desenvolvedores

# FERRAMENTAS PARA DESENVOLVIMENTO FRONTEND

## 3. Atom

- Código aberto
- Desenvolvido com HTML, CSS e JavaScript
- Extensões em Node.js

## 4. Notepad++

- Simples e poderoso
- Código aberto e personalizável

# FERRAMENTAS DE INSPEÇÃO DO NAVEGADOR

DevTools (Chrome, Firefox, Edge, etc.)

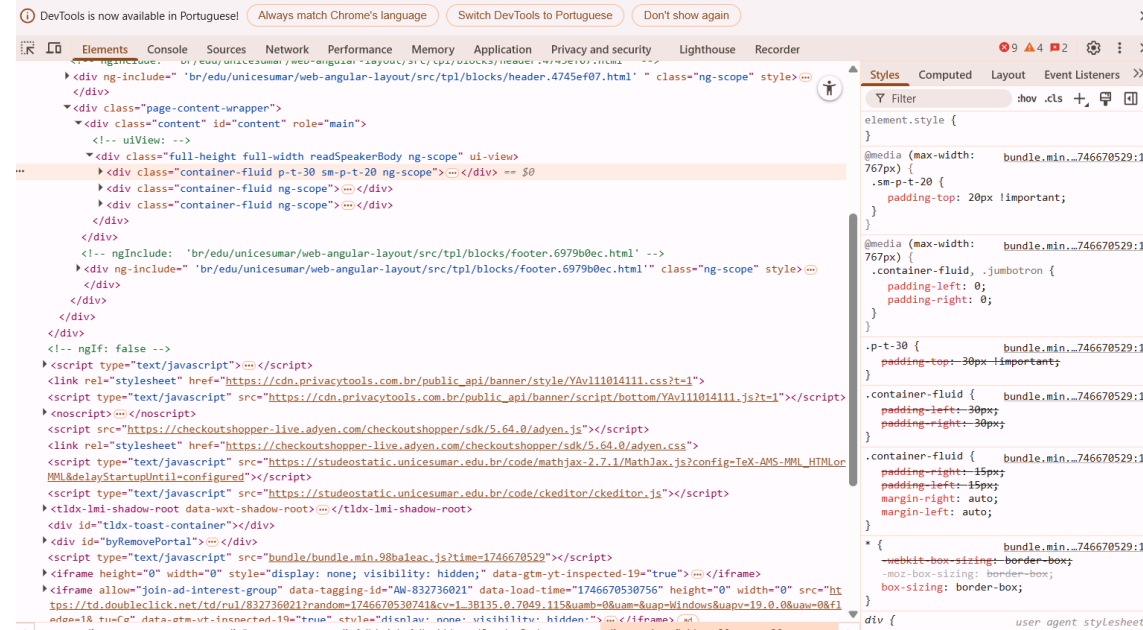
- Inspetor de elementos
- Console JavaScript
- DepuradorAnálise de rede
- Estudo de performance



# FERRAMENTAS DE INSPEÇÃO DO NAVEGADOR

DevTools (Chrome, Firefox, Edge, etc.)

- Inspetor de elementos
- Console JavaScript
- Depurador
- Análise de rede
- Estudo de performance



# FRAMEWORKS: USAR OU NÃO?

## Vantagens dos frameworks:

- Agilidade e produtividade
- Reutilização de código
- Padrões de projeto simplificados
- Maior qualidade no produto final

## Por que estudar sem frameworks primeiro:

- Entender os fundamentos é essencial
- Conhecimento na linguagem "pura" dá domínio sobre o código
- Frameworks mudam constantemente
- Base sólida facilita a adaptação a novas ferramentas

# FRAMEWORKS E BIBLIOTECAS POPULARES

## CSS:

- Bootstrap - Biblioteca popular para UI
- Tailwind - Framework utilitário
- Sass/Less - Pré-processadores de CSS

## JavaScript:

- React.js - Biblioteca para interfaces
- Vue.js - Framework progressivo
- Angular - Framework completo
- Next.js - Framework React com SSR

# DICAS

- A programação demanda prática constante
- Ver código pronto é diferente de escrever sozinho
- Consultar referências é normal e necessário
- Mesmo desenvolvedores experientes consultam documentação
- "Abrace a documentação e seja feliz!"

# CONCLUSÃO

- Entender tanto frontend quanto backend é valioso
- A base da linguagem mantém-se mais estável que frameworks
- Conhecimento profundo facilita a adaptação a novas tecnologias
- Prática e consulta a documentação são essenciais

# RECURSOS ADICIONAIS

- Documentação oficial: MDN Web Docs
- Comunidades: Stack Overflow, GitHub
- Canais no YouTube sobre desenvolvimento web

**BONS ESTUDOS**