



Curso: Ciência da Computação

Disciplina: Front-End Frameworks

Prof. Romulo Cezar

Tecnologias utilizadas no front-end

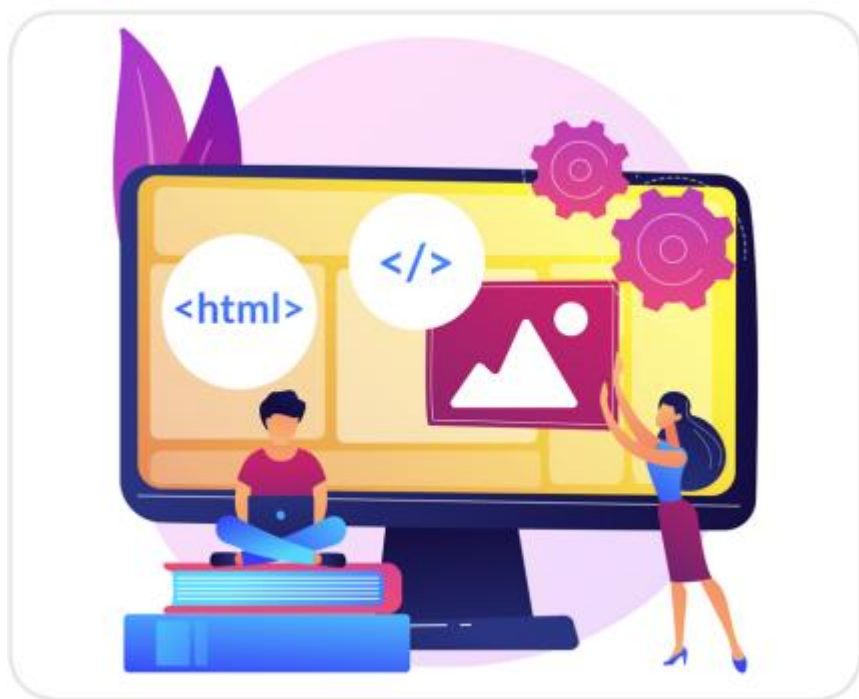
Para compreender o front-end, é essencial conhecer as principais tecnologias utilizadas nessa área. O desenvolvimento front-end envolve uma variedade de ferramentas e linguagens que permitem criar interfaces de usuário interativas e visualmente atraentes.

Vamos ver algumas das tecnologias mais relevantes e suas aplicações.

O HTML é a base do front-end, pois ele define a estrutura e o conteúdo de uma página web. Com HTML, é possível criar elementos como títulos, parágrafos, imagens, links e formulários.

Por exemplo, ao utilizar a tag ``, podemos inserir uma imagem em uma página, especificando o caminho do arquivo correspondente.

Figura 1 - Com o HTML, podemos criar elementos como cabeçalhos, parágrafos, imagens, links e formulários



O CSS é responsável pela aparência visual das páginas web.

Com o CSS, podemos definir estilos, como cores, fontes, tamanhos, margens e animações.

Por exemplo, podemos usar o seletor de classe no CSS para estilizar todos os elementos com uma determinada classe, alterando sua cor de fundo ou o tamanho da fonte.

O JavaScript, por sua vez, é uma linguagem de programação que permite adicionar interatividade e dinamismo a uma página web.

Com o JavaScript, podemos criar eventos, manipular elementos da página, fazer requisições assíncronas e criar animações. Por exemplo, podemos usar o JavaScript para exibir uma mensagem de boas-vindas em uma janela de pop-up quando um botão é clicado.

Além dessas tecnologias, existem diversos frameworks e bibliotecas que facilitam o desenvolvimento front-end. Um exemplo popular é o React, que é um framework JavaScript mantido pelo Facebook.

E o que são Frameworks ?

Frameworks são conjuntos de ferramentas, bibliotecas e convenções preestabelecidas que facilitam o desenvolvimento de aplicações web. Eles oferecem estruturas e padrões para agilizar o processo de criação, fornecendo componentes reutilizáveis, gerenciamento de estados e interação com o usuário, permitindo aos desenvolvedores focar mais na lógica de negócio do que nas tarefas repetitivas de codificação.

E o que é o React?

É uma biblioteca que facilita a criação de interfaces de usuário dinâmicas e baseadas em componentes, o que torna o código mais reutilizável e permite atualizações eficientes da interface conforme o estado da aplicação muda. **Por exemplo**, é possível desenvolver um formulário interativo com React, onde qualquer alteração nos campos é refletida imediatamente na tela.

Outro framework bastante utilizado é o **Angular**, desenvolvido pelo Google. Ele oferece uma estrutura completa para desenvolvimento de aplicações web, incluindo funcionalidades como manipulação de dados, navegação entre páginas (roteamento), validação de formulários etc.

Com o Angular, **por exemplo**, podemos criar uma aplicação de lista de tarefas que permite adicionar, editar e remover itens em tempo real.

E o que são Interfaces de Usuário? Interfaces de usuário reativas e componentizadas são abordagens de desenvolvimento front-end que priorizam a reatividade e a modularidade.

Elas permitem que os elementos da interface respondam dinamicamente a eventos e alterações de estado, proporcionando uma experiência interativa e fluida para o usuário.

O **Vue.js** é mais um framework JavaScript que tem se destacado nos últimos anos. É reconhecido por sua simplicidade e facilidade com que pode ser integrado a projetos já existentes. Entre seus principais recursos estão a componentização e o gerenciamento de estado.

Um exemplo de aplicação com Vue.js seria a criação de uma galeria de imagens com carregamento assíncrono e transições suaves.

Essas são apenas algumas das tecnologias mais utilizadas no desenvolvimento front-end.

Como essa área está em constante evolução, é fundamental estar atento às tendências e novidades. Dominar essas ferramentas permite criar interfaces modernas, responsivas e de alta qualidade, proporcionando uma experiência agradável e eficiente para o usuário.

Figura 2 - O domínio dessas tecnologias permite criar interfaces interativas, responsivas e de alta qualidade



De modo geral, entender as principais tecnologias usadas no front-end é fundamental para se tornar um desenvolvedor web completo.

O HTML define a estrutura, o CSS define a aparência, e o JavaScript adiciona interatividade. Além disso, frameworks como React, Angular e Vue.js oferecem recursos avançados para o desenvolvimento front-end.

Sintetizando

Vimos até aqui os conceitos básicos de front-end, diferenciando-o do back-end. Falamos das principais tecnologias utilizadas no front-end, como HTML, CSS e JavaScript.

Com relação ao HTML, vimos que ele é a linguagem fundamental do frontend e permite criar a estrutura e o conteúdo de uma página web. Com o uso de tags e elementos, podemos criar cabeçalhos, parágrafos, imagens e links.

Já o CSS é responsável pela aparência visual das páginas web.

Ele nos permite definir estilos, como cores, fontes, tamanhos e margens.

Com ele, podemos estilizar elementos de forma precisa e criar animações para tornar a experiência do usuário mais atraente. Outro ponto importante foi o JavaScript, uma linguagem de programação que adiciona interatividade e dinamismo a uma página web.

Com ele, podemos criar eventos, manipular elementos da página, fazer requisições assíncronas e criar animações.

Essa tecnologia nos permite criar experiências mais ricas e interativas para os usuários. Além disso, abordamos o uso de frameworks e bibliotecas Front-End, como React, Angular e Vue.js, que facilitam o desenvolvimento de aplicações web eficientes e modernas.

Essas ferramentas nos permitem criar interfaces de usuário reativas, componentizadas e escaláveis, aumentando a produtividade e a qualidade do código. Compreender esses conceitos e dominar essas tecnologias é essencial para se tornar um desenvolvedor front-end.

HTML e CSS

Veremos agora a estrutura básica do HTML, as tags e os elementos essenciais na construção de uma webpage, bem como utilizar técnicas de CSS para criar layouts responsivos e design fluido, integrando HTML, CSS e princípios de design responsivo.

Isso é fundamental para o desenvolvedor front-end, pois é possível criar páginas web com uma estrutura organizada, estilos personalizados e layouts que se adaptam a diferentes dispositivos (desktops, smartphones e tablets).

A habilidade de aplicar corretamente a estrutura do HTML e usar técnicas avançadas de estilização com CSS permite criar páginas web que atendam aos padrões modernos e sigam as melhores práticas de desenvolvimento front-end.

Isso inclui a capacidade de utilizar corretamente as tags e os elementos do HTML para estruturar o conteúdo, aplicar estilos personalizados com CSS para tornar a página visualmente atraente e usar técnicas de design responsivo para garantir que a página se adapte de forma fluida a diferentes tamanhos de tela.

Estrutura básica do HTML, tags e elementos essenciais na construção de uma webpage

Ao aplicar a estrutura básica do HTML, as tags e os elementos essenciais na construção de uma página da web, estamos dando os primeiros passos para a criação de um site funcional e bem-organizado.

O HTML é a linguagem de marcação utilizada para estruturar e organizar o conteúdo de uma página na web, enquanto as tags e os elementos fornecem a estrutura e o significado semântico para os diferentes elementos presentes na página. A estrutura básica do HTML é composta por algumas tags fundamentais que devem estar presentes em todas as páginas.

Um exemplo dessa estrutura básica é:

```
```html
<!DOCTYPE html>
<html>
 <head>
 <title>Título da Página</title>
 </head>
 <body>
 <!-- Conteúdo da Página -->
 </body>
</html>
```
```

Nesse exemplo, temos a tag `<!DOCTYPE html>`, que define o tipo de documento como HTML5.

Em seguida, temos a tag `<html>`, que delimita o início e o fim do documento HTML. Dentro do elemento `<html>`, temos a tag `<head>`, na qual são incluídas informações sobre a página, como o título exibido na barra do navegador.

Por fim, temos o elemento `<body>`, em que é inserido o conteúdo visível da página.

Além da estrutura básica, existem várias tags e elementos essenciais que podemos utilizar na construção de uma página da web.

Alguns exemplos são:

- *Tags* de títulos: as *tags* ``<h1>``, ``<h2>``, ``<h3>``, ``<h4>``, ``<h5>`` e ``<h6>`` são utilizadas para definir títulos e subtítulos na página, sendo que ``<h1>`` é o título principal e ``<h6>`` é o menor subtítulo.
- Parágrafos: a *tag* ``<p>`` é utilizada para criar parágrafos de texto na página.
- Listas: as *tags* ```` (lista não ordenada) e ```` (lista ordenada) são utilizadas para criar listas de itens. Os itens de uma lista são definidos pela *tag* ````.
- *Links*: a *tag* ``<a>`` é utilizada para criar *links* para outras páginas ou recursos. O atributo ``href`` define o endereço do *link*.

- Imagens: a tag `` é utilizada para exibir imagens na página. O atributo `src` define o caminho da imagem.
- Tabelas: a tag `<table>` é utilizada para criar tabelas na página. As células da tabela são definidas pelas tags `<tr>` (linha da tabela) e `<td>` (célula da tabela).

Esses são apenas alguns exemplos das tags e elementos essenciais que podemos usar na construção de uma página da web.

A partir desses elementos, é possível criar uma estrutura bem-organizada e **semântica**, permitindo que os navegadores e mecanismos de busca entendam a hierarquia e o significado do conteúdo da página.

E o que é Semântica?

Semântica refere-se ao uso apropriado de elementos HTML para atribuir significado e estrutura correta ao conteúdo de uma página web. Ao aplicar semântica, os elementos HTML são escolhidos com base em seu significado e propósito, tornando o código mais legível para humanos e máquinas, além de melhorar a acessibilidade e a indexação pelos mecanismos de busca.

Ao aplicar corretamente a estrutura básica do HTML, as tags e os elementos essenciais, estamos criando uma base sólida para a construção de uma página da web.

Com o uso adequado do HTML, podemos estruturar o conteúdo, tornando-o acessível e bem-organizado. Isso facilita a navegação do usuário, melhora a experiência do usuário e contribui para um melhor posicionamento nos resultados de busca.

Próxima aula

-Técnicas de estilização com CSS

- Alguns projetos HTML

Projeto 1: Página Inicial Simples

Neste projeto em HTML, vamos criar uma página inicial simples com um cabeçalho, um parágrafo e um link.

Explicação do Código do Projeto 1

Vamos começar com o código HTML básico:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Minha Página Inicial</title>
5 </head>
6 <body>
7     <header>
8         <h1>Bem-vindo à Minha Página</h1>
9     </header>
10    <p>Esta é uma página inicial simples.</p>
11    <a href="#">Saiba mais</a>
12 </body>
13 </html>
```

Neste código, usamos as tags HTML para criar o cabeçalho, parágrafo e link. A tag <header> define o cabeçalho da página, e a tag <a> cria um link.