



# WEB BÁSICO AULA 1

# OBJETIVOS GLOBAIS DE APRENDIZAGEM

- Aplicar conceitos, métodos, técnicas através de ferramentas para o desenvolvimento de sistemas web utilizando linguagens de programação para demonstração conceitual e aplicada aos contextos de mercado com a utilização de frameworks em nível básico e intermediário. Projetar soluções usando tecnologia da informação nas organizações a fim de adequá-las às oportunidades de mudança. Desenvolver projetos de software com a utilização de frameworks, garantindo a resolução algorítmica de problemas.

| Aula    | Atividade prevista  |
|---------|---|
| Aula 01 | Apresentação da disciplina, orientações gerais e versionamento de código no GitHub.   |
| Aula 02 | Introdução ao desenvolvimento web, visão geral de TAGs HTML e criação de páginas HTML (html, head, meta, title, link, script, body).        |
| Aula 03 | Elementos de estruturação de páginas HTML (header, nav, main, article, section, aside, footer, div), pastas e navegação com hiperlinks (a). |
| Aula 04 | Estruturação de conteúdo (h1, h2, h3, p, textarea, br, ol, ul, li, span, strong, em, img, video, audio...).                                 |
| Aula 05 | Elementos de formulários (form, inputs, selects, buttons...).   |
| Aula 06 | Elementos de tabelas (table, caption, th, tr, td,thead, tfoot, tboby...).   |
| Aula 07 | Introdução aos seletores type, class e id, conceito do efeito cascata.  |
| Aula 08 | Formatação de conteúdo e box model (color, font-, text-, background, border, margin, padding, width, height...).                            |
| Aula 09 | Layouts e responsividade (display, float, position, z-index, overflow, column-, media).   |
| Aula 10 | Layouts e responsividade (flexbox e grid layout).   |
| Aula 11 | Introdução ao NodeJS e Javascript funcional (Visão geral de tipos, operadores e funções).   |
| Aula 12 | Criação, inserção, remoção e manipulação de componentes HTML no Document Object Model (DOM) usando Javascript.                              |
| Aula 13 | Criação, inserção, remoção e manipulação de componentes HTML no Document Object Model (DOM) usando Javascript.                              |
| Aula 14 | Consumo de APIs usando Fetch API com JSON Server (get, post, put e delete).   |

# O que é desenvolvimento web?

- Desenvolvimento web é o processo de criar sites e aplicações para a internet ou para uma network privada, conhecido como intranet. Desenvolvimento web não se trata do designer de um site; mas, ao invés disso, é tudo sobre codificação e programação que impulsionam a funcionalidade do site.
- Dos sites mais simples e estáticos para as plataformas de mídias sociais e aplicativos, de sites de e-commerce até sistemas de administração de conteúdo (CMS); todas as ferramentas que utilizamos diariamente na internet foram feitas por desenvolvedores web.
- O que faz um desenvolvedor web se envolve nestas três camadas: codificação client-side (frontend), codificação server-side (backend) e tecnologia de banco de dados.

# Client-side

- Script client-side, ou desenvolvimento frontend, refere-se a tudo que o usuário final experimenta diretamente. Programação client-side se executa no browser da internet e está relacionado diretamente com o que as pessoas vêem quando visitam um site. Coisas como layout, fontes, cores, menus e formulários de contato são todos feitos por frontend.

# Server-side

- Server-side scripting, ou desenvolvimento backend, se trata de tudo que acontece nos bastidores. O backend é essencialmente a parte de um website que o usuário não vê. É responsável por armazenar e organizar dados, e garantir que tudo no server-side funcione bem. Ele faz isso comunicando com o frontend. Quando algo acontece no server-side — digamos que um usuário preencha um formulário—o navegador envia um pedido para o server-side. O server-side “responde” com informação relevante em forma de programação frontend no qual o browser pode interpretar e mostrar.

# Tecnologia de database

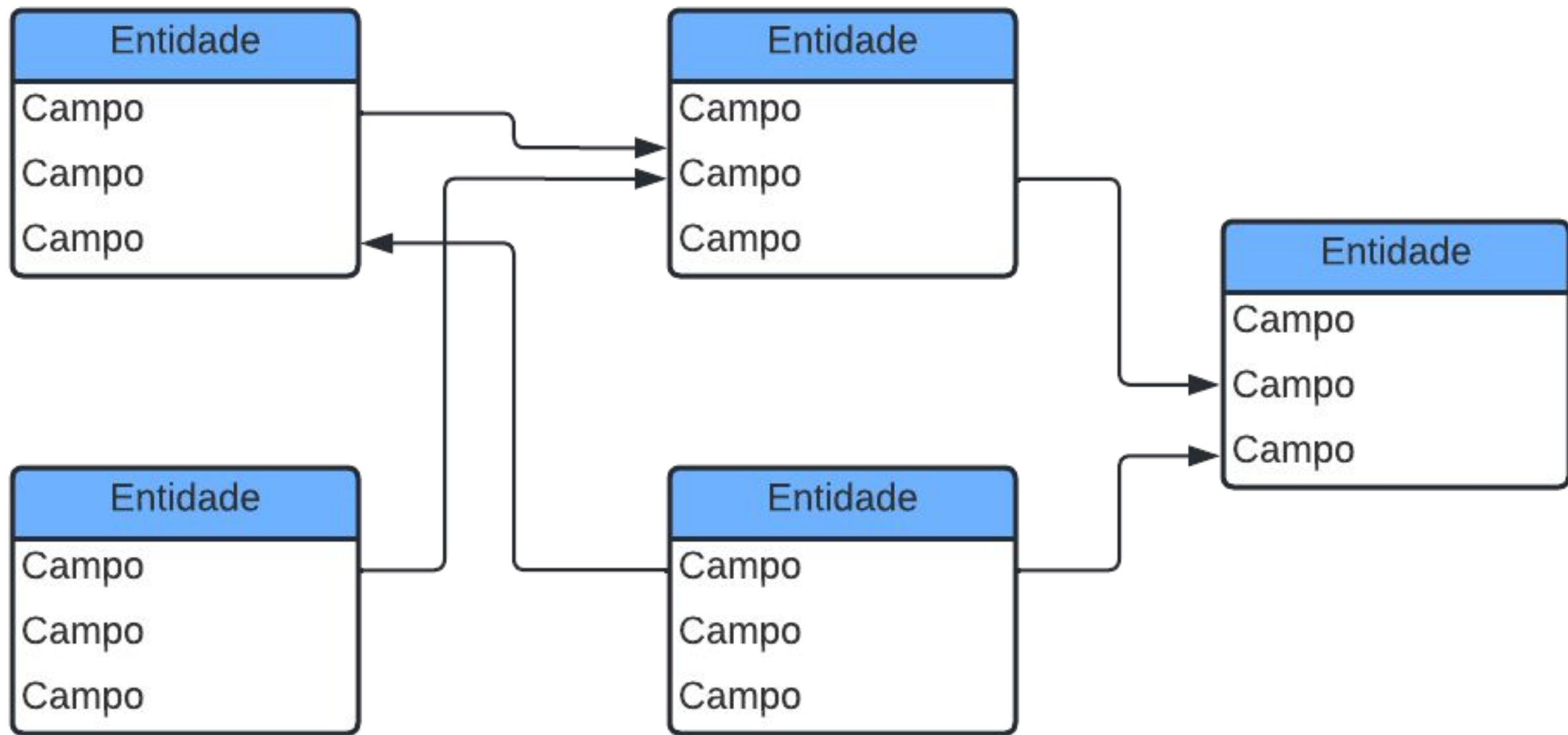
- Aprender sobre o que faz um desenvolvedor web também envolve aprender sobre banco de dados, pois sites também contam com a tecnologia de banco de dados. Os bancos de dados contêm todos os dados e conteúdos que são necessários para o funcionamento de um site, armazenando-os de tal maneira que seja fácil para recuperar, organizar, editar e salvar. O banco de dados roda em um servidor, e muitos sites usam alguma forma de sistema de banco de dados.



# Banco de dados

- Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um **sistema de gerenciamento de banco de dados (DBMS)**. Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados.





# O que um desenvolvedor web faz?

## DESENVOLVEDOR FRONTEND

- Codifica a área externa de um website, que são as partes que os usuários enxergam e interagem
- Traz vida ao WebDesign usando HTML, JavaScript e CSS
- Garante um design responsivo

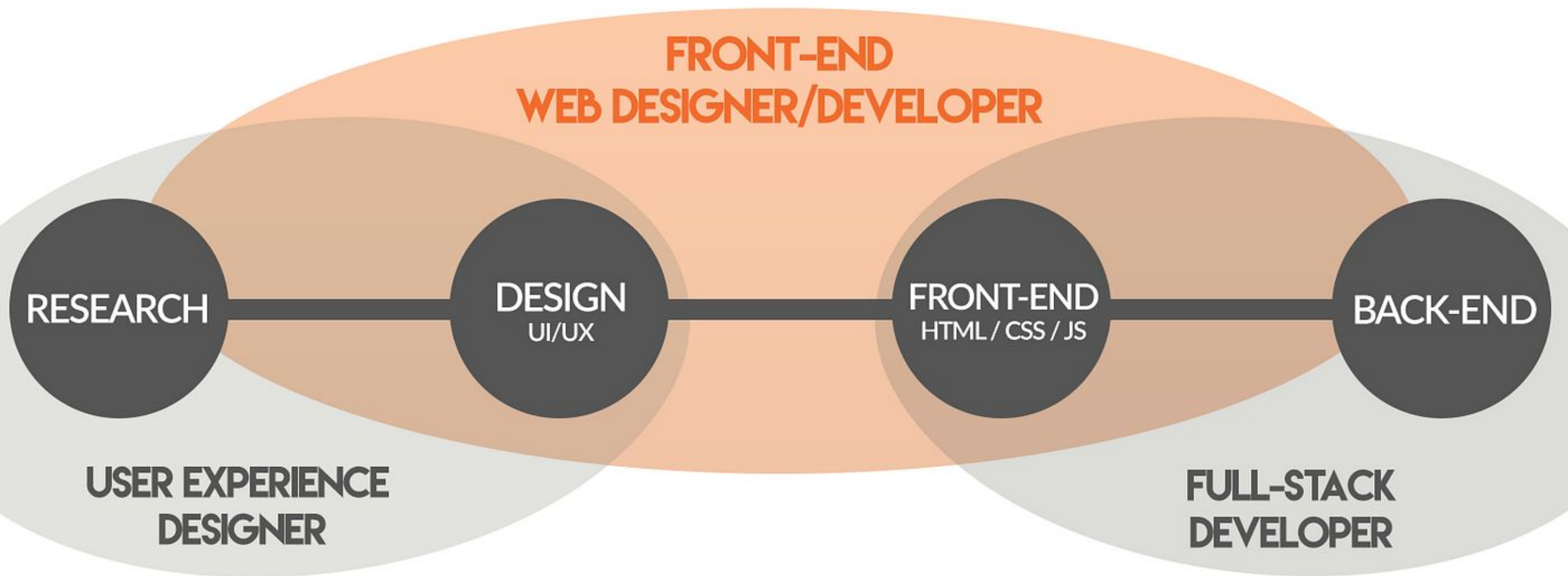
## DESENVOLVEDOR BACKEND

- Trabalha por de trás das páginas, construindo e mantendo a tecnologia necessária para alimentar o Frontend
- Certifica-se de que o Frontend está funcional
- Cria e gerencia a base de dados

## DESENVOLVEDOR FULL STACK

- Especialista em Front-end e Back-end
- Guia as estratégias e boas práticas
- Antenado com lógicas de negócio e experiência do usuário





# Construção de interfaces

- Em programação, **front end diz respeito a tudo que é visível pelo usuário**, como os menus, interfaces e outros elementos gráficos. O Desenvolvedor Front End é aquele que se dedica ao que está “na superfície”, no layout de um site ou no design de um aplicativo.
- A primeira atribuição desse profissional é a **construção da interface do software** sendo desenvolvido. Não na parte artística, mas principalmente quanto à técnica, no desenvolvimento do código que dá origem a todos esses elementos visuais.
- Bons desenvolvedores utilizam essas ferramentas já levando em conta boas práticas de interface, além de **tornar o código mais fácil de ler e editar no futuro**. Dessa forma, qualquer atualização de design fica mais fácil de aplicar.

# Implementação do design

- É importante esclarecer que o trabalho desse desenvolvedor não é o mesmo de um designer. Enquanto o designer elabora o esquema de cores e pensa na experiência do usuário, o **Desenvolvedor Front End fará o trabalho de implementação desse design**
- Você também pode estudar elementos de **UI e UX** para se aprimorar nessa carreira, mas o foco está em como o código se traduz em todos os elementos visuais.

# Correção de erros visuais

- Boa parte dos **problemas visuais** encontrados em softwares e sites são resultado de algum erro no código, como um valor incorreto de uma variável ou um erro de digitação. Coisas assim podem levar a imagens que não carregam, menos fora de enquadramento ou ícones que não funcionam.
- O Desenvolvedor Front End também tem a tarefa de **identificar e corrigir esses erros** ao longo do desenvolvimento. Muitos deles podem ocorrer após o desenvolvimento, devido a atualizações em sistemas operacionais, navegadores ou outras mudanças no próprio código usado.

- **Monitorar a experiência do usuário**

- Para saber se o **design utilizado no site ou software foi efetivo**, é fundamental monitorar as atividades do usuário final.
- Coletar dados sobre as principais partes da **UI** que dificultam a navegação ajuda a identificar quais são as melhorias necessárias em futuras versões.

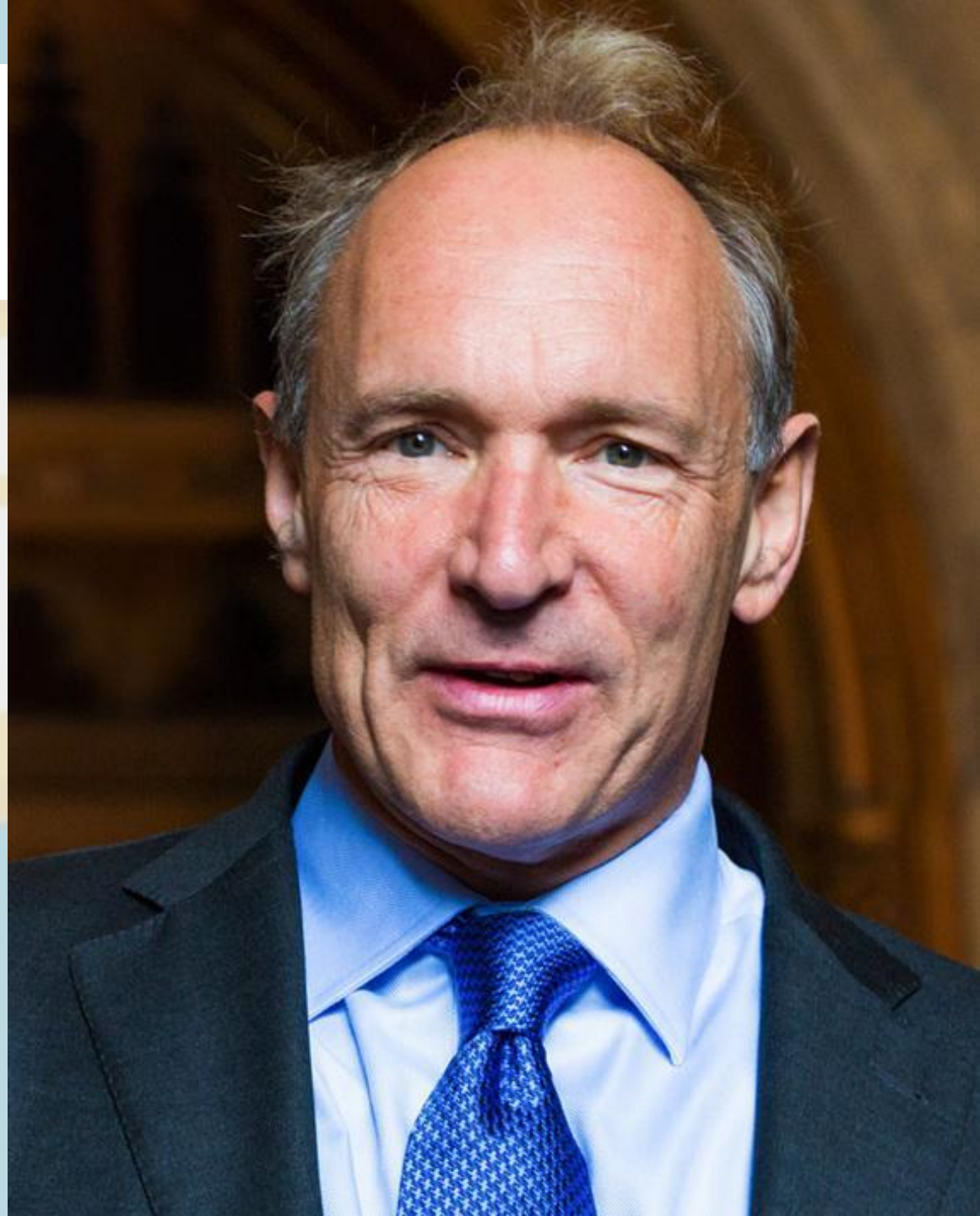
- **Captação e implementação de feedback**

- Mesmo que o produto final tenha sido entregue, ainda deve haver bastante espaço para modificações e melhorias.
- Para isso, o **Desenvolvedor Front End deve buscar feedback dos usuários**, tentando entender mais claramente como estão as experiências a fim de propor novas etapas no desenvolvimento, se for o caso.



# O Que é HTML

- O HTML foi inventado por **Tim Berners-Lee**, um físico do centro de pesquisas CERN, na Suíça. Ele surgiu com a ideia de um sistema de hipertexto na internet.



# O Que é HTML

- A Linguagem de Marcação de Hipertexto (HTML) é uma linguagem de computador que compõe a maior parte das páginas da internet e dos aplicativos online. Um hipertexto é um texto usado para fazer referência a outros textos, enquanto uma linguagem de marcação é composta por uma série de marcações que dizem para os servidores da web qual é o estilo e a estrutura de um documento.
- O HTML não é considerado uma linguagem de programação, já que ele não pode criar funcionalidades dinâmicas. Ao invés disso, com o HTML, os usuários podem criar e estruturar seções, parágrafos e links usando elementos, tags e atributos.

# Elementos Bloco

- Os elementos bloco ocupam a largura completa de uma página. Eles sempre iniciam uma nova linha no documento. Por exemplo, um elemento cabeçalho estará numa linha diferente de um elemento parágrafo.
- Toda página HTML usa essas três tags:
- A tag `<html>` é o elemento raiz que define todo o documento HTML.
- A tag `<head>` contém metadados sobre o título e a codificação de caracteres da página.
- A tag `<body>` inclui todo o conteúdo que aparece na página.



```
<html>
```

```
<head>
```

```
<!-- META INFORMAÇÕES -->
```

```
</head>
```

```
<body>
```

```
<!-- CONTEÚDO DA PÁGINA -->
```

```
</body>
```

```
</html>
```

- <!DOCTYPE html>
- <head>
- </head>
- <body>
- </body>
- </html>

- O HTML funciona através do uso de várias **tags** para definir as estruturas dos textos. Elas incluem cabeçalhos, tabelas e parágrafos. Cada tag é definida através do uso da fórmula **<A>** e **</A>**. Elas são chamadas de tags de “abertura” e “fechamento”, respectivamente.

**<p> CONTEÚDO**  
**</p>**

**Tag de abertura**

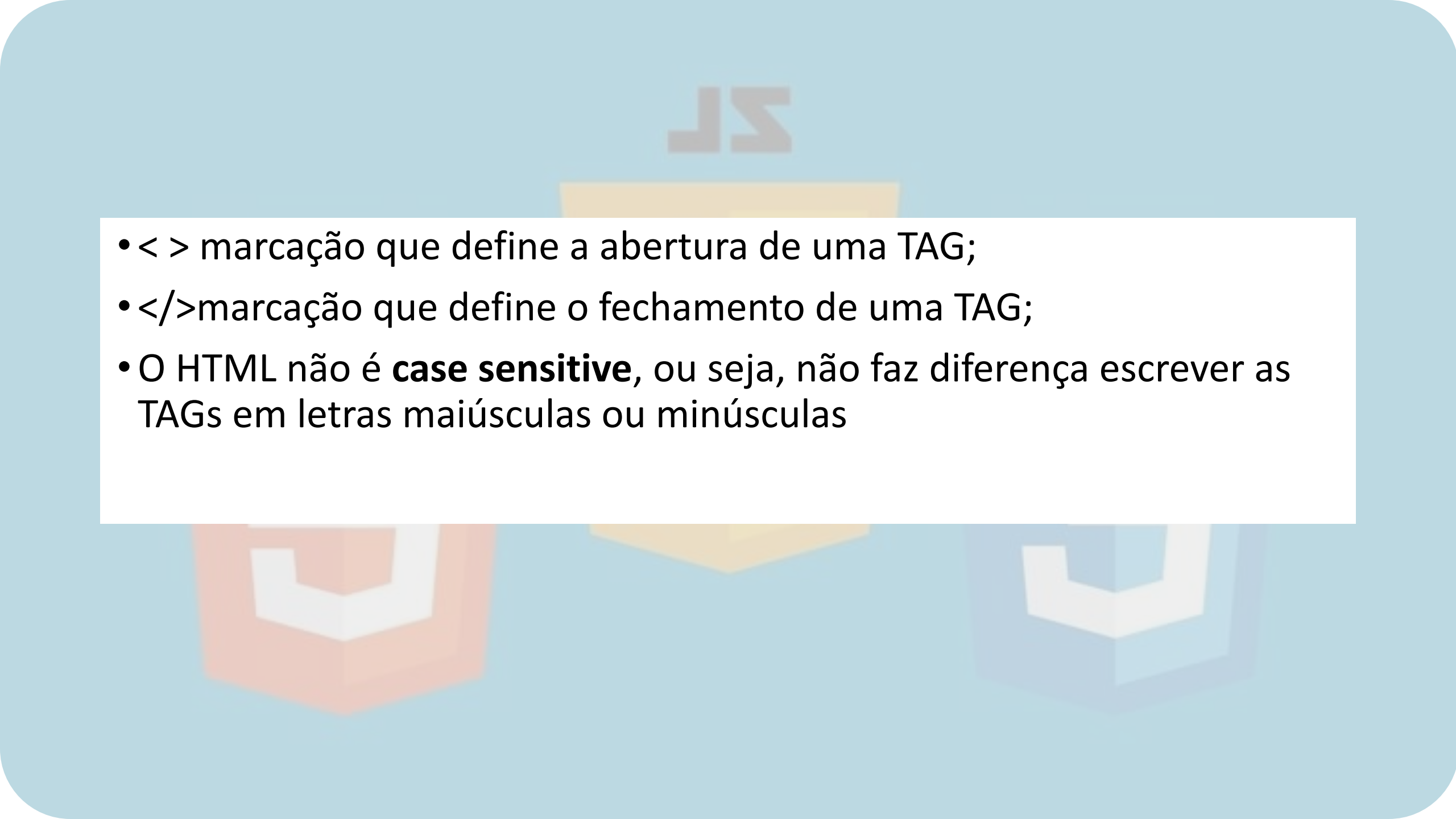
**Tag de fechamento**

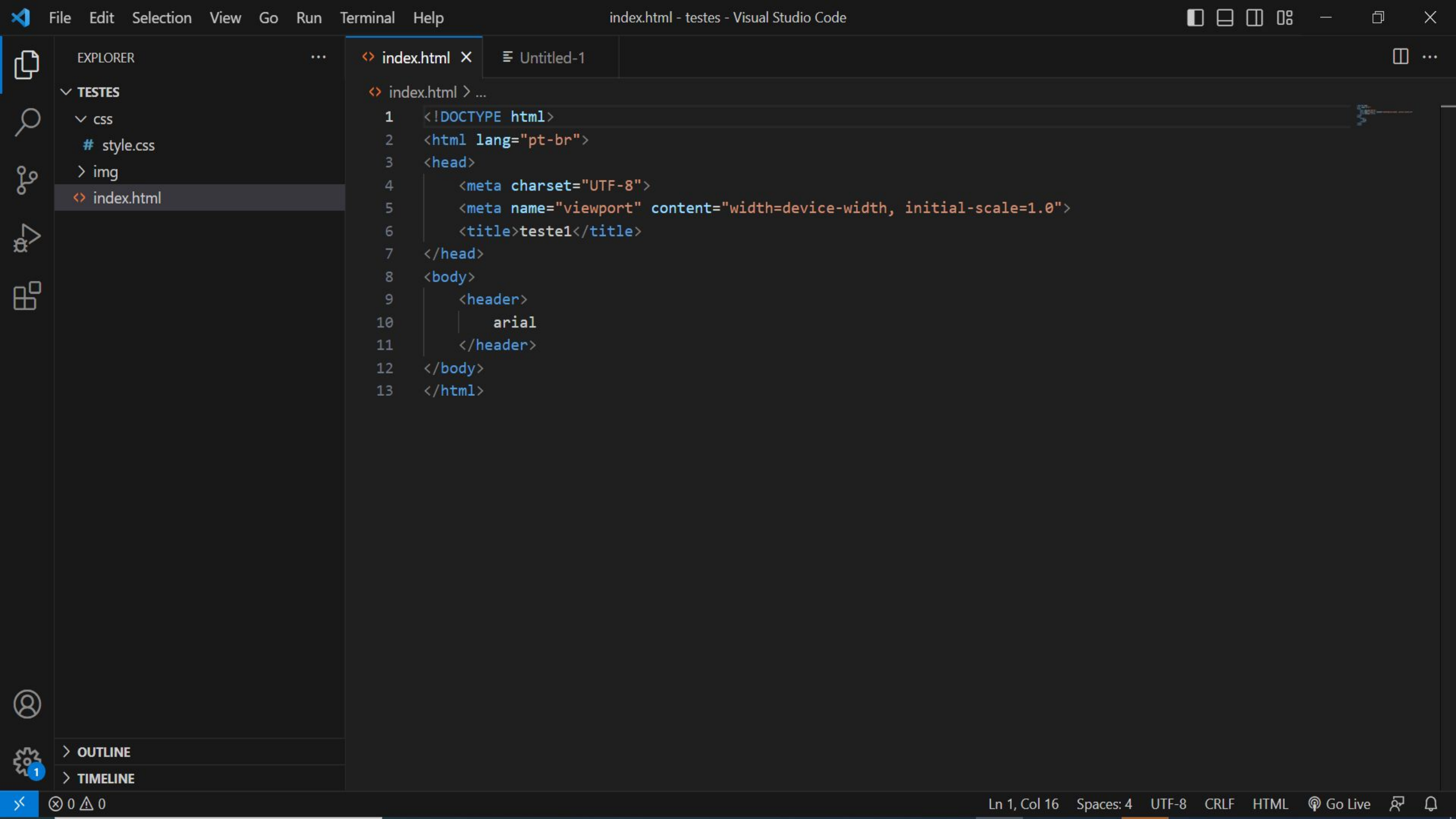
**<p class="maneiro">Olá mundo!</p>**

**Um atributo e seu valor**

**Conteúdo que será exibido**



- 
- < > marcação que define a abertura de uma TAG;
  - </> marcação que define o fechamento de uma TAG;
  - O HTML não é **case sensitive**, ou seja, não faz diferença escrever as TAGs em letras maiúsculas ou minúsculas



# Qual a Relação Entre HTML, CSS e JavaScript?

- Mesmo que o HTML seja uma linguagem poderosa, não é o suficiente para criar um site profissional e responsivo. Ele pode ser utilizado apenas para adicionar elementos de texto e criação de estruturas de conteúdo.
- Porém o HTML funciona muito bem com outras duas linguagens de front-end: **CSS (Cascading Style Sheets)**, e **JavaScript**. Juntos eles proporcionam a implementação de funcionalidades avançadas e uma ótima experiência ao usuário.
- O CSS é responsável pelo estilo como background, cores, layouts, espaçamentos, e animações.
- O JavaScript permite a adição de funcionalidades dinâmicas como sliders, pop-ups, e galerias de fotos.



**JavaScript**

**Behavioral**

**CSS**

**Presentational**

**HTML**

**Structural**

- Pense neles como as linguagens de codificação fundamentais da Internet. Assim como no mundo existem vários idiomas, as páginas da web também têm idiomas. No caso dos sites, suas linguagens são **HTML**, **CSS** e **Javascript**. Claro que você já deve ter ouvido falar deles, mas você realmente sabe como eles funcionam?
- Desenvolvimento Web é o que você vê quando acessa um site. Eu os chamo de “Os três amigos”.
- HTML é Hypertext Markup Language, ou “**O Construtor**”.
- CSS é Cascading Style Sheets, que considero “**O Artista**”.
- E JavaScript “**O Feiticeiro**”.



# HTML4

div id = "header"

div id =  
"menu"

div class = "post"

div class = "post"

div id = "footer"

**VS**

# HTML5

header

nav

article

article

footer

# HTML



# SEMÂNTICO

`<header>`

`<nav>`

`<section>`

`<aside>`

`<article>`

`<footer>`



**<header>**

**<nav>**

**<section>**

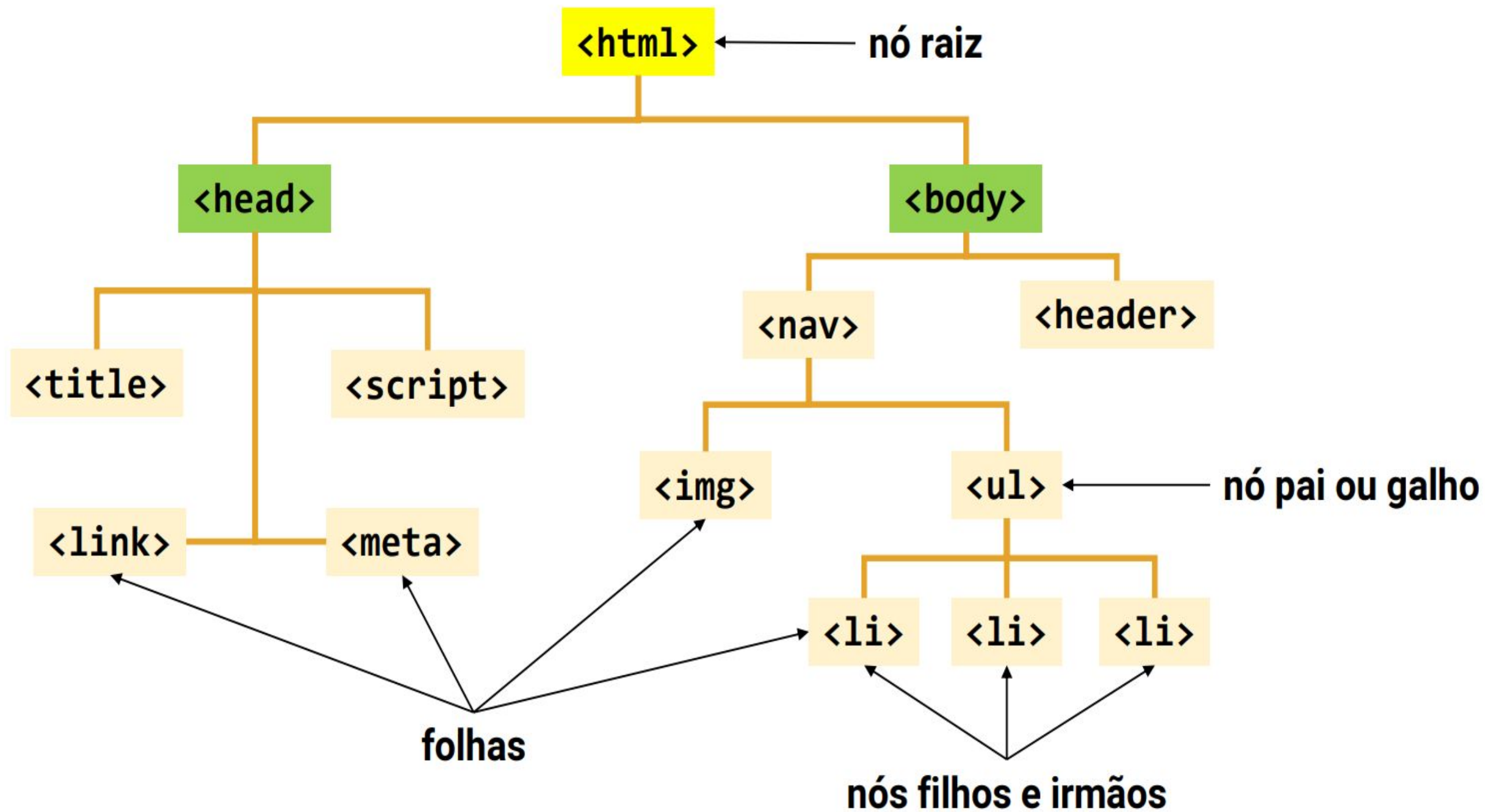
**<aside>**

**<footer>**

- **<header>** cabeçalho da página ou seção;
- **<nav>** links de navegação;
- **<section>** seções da página;
- **<article>** conteúdo da página;
- **<aside>** conteúdo na barra lateral;
- **<footer>** rodapé da página ou seção.

# SEMÂNTICA

<article>, <h1>, <h2>, <h3>, <p>, <b>, <strong>, <i>, <em>, <sup>, <sub>, <cite>, <blockquote>, <img>, <small>, <mark>, <br>, <pre>, <ul>, <ol>, <li>, <time>, <code>.



# PARA SABER MAIS

- <https://www.impacta.com.br/blog/20-topicos-bom-desenvolvedor-front-end-deveria-saber-em-2023/>
- [https://www.w3schools.com/html/html\\_css.asp](https://www.w3schools.com/html/html_css.asp)
- <https://www.w3c.br/>