

# Curso de Jogos Digitais

## Tecnologias Web

### Aula 01 - Introdução

Professor: André Flores dos Santos

Santa Maria – RS 2025

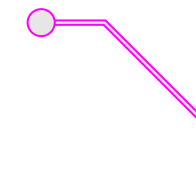
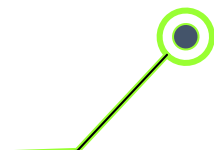
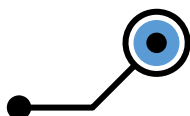


# Professor André:



- **Graduação em Ciência da Computação** (Bacharelado) pela Universidade Franciscana (UFN).
- **Graduação no Programa Especial de Graduação de Formação de Professores para a Educação Profissional (Licenciatura)** pela Universidade Federal de Santa Maria (UFSM).
- **Mestrado em Microeletrônica** pela Universidade Federal do Rio Grande do Sul (UFRGS).
- **Doutorado em Nanociências** pela Universidade Franciscana (UFN).

Trabalhos desenvolvidos ao longo da vida acadêmica em temas relacionados à tecnologia da informação, programação de computadores, software e hardware. Conhecimento e aprendizado em Física, Química, Biologia, Biotecnologia, Bioinformática e nanoestruturas, visando o desenvolvimento de pesquisas no Programa de Pós-Graduação em Nanociências (UFN).



# APRESENTAÇÃO DOS ALUNOS

- > Cada um falar o seu nome, cidade, se já tem experiência na área de computação, etc.
- > Quais os planos para o seu futuro, se pretende ser programador, desenvolvedor de jogos, etc.
- > E o que entende sobre **Tecnologias Web**, onde é aplicado e de que forma?

# SUMÁRIO

<b>01</b>	<b>02</b>	<b>03</b>
INTRODUÇÃO	HTML	CLIENTE-SERVIDOR
<b>04</b>	<b>05</b>	<b>06</b>
FLUXO	HOSPEDAGEM	TAGS

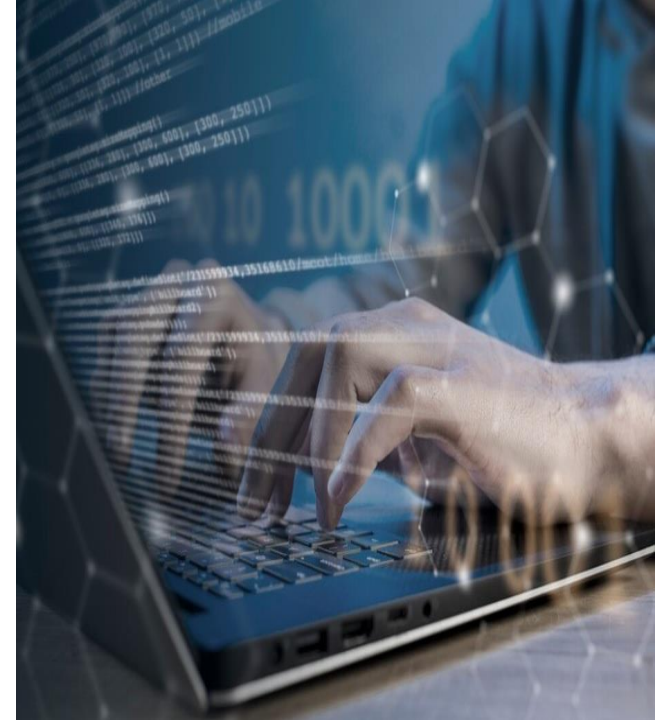
# Linguagem HTML

Ao falar de internet é impossível não mencionar a principal linguagem responsável por sua evolução, o *Hypertext Markup Language*, ou simplesmente HTML.

- Seu criador foi o matemático Tim Berners Lee, para suprir a necessidade de comunicação entre ele e um grupo de colegas pesquisadores.

# Linguagem HTML

- Com o crescimento da internet pública, HTML tornou a base para o desenvolvimento de aplicações e recursos para a transmissão de informações na *World Wide Web*.
- As primeiras especificações foram publicadas por Berners Lee no ano de 1993, tendo publicada a versão 2.0 no ano de 1995.



# Linguagem HTML

- A partir do ano de 1996 o desenvolvimento de especificações foi atribuído e mantido por um grupo de empresas fabricantes de software denominada *World Wide Web Consortium*, ou simplesmente W3C.



# Pilares estipulados pela W3C

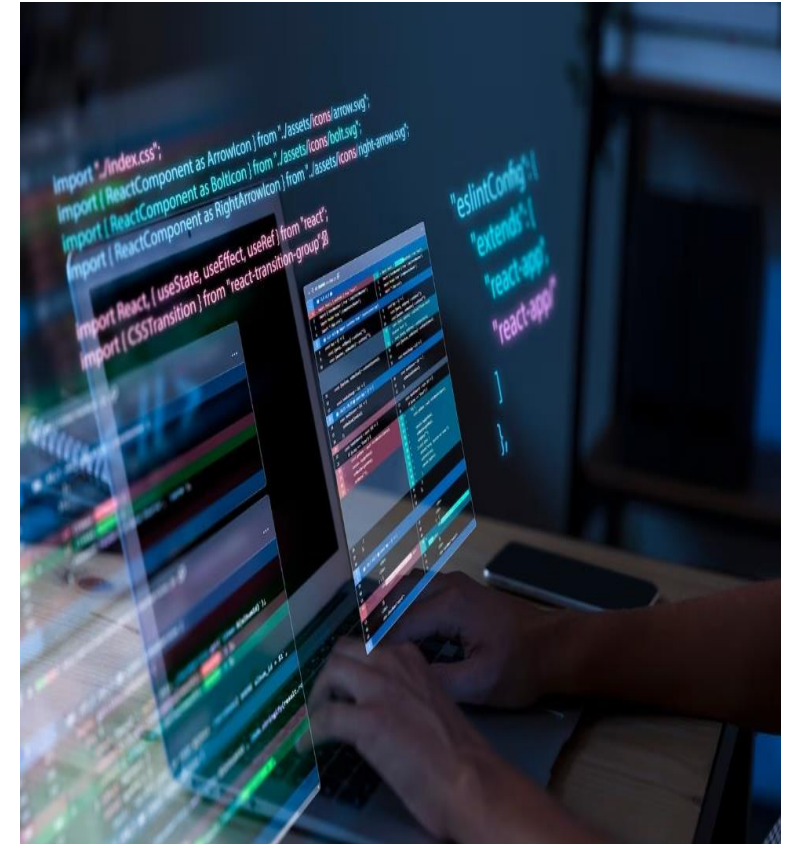
- Um esquema de nomes para localização de fontes de informação na Web, esse esquema chama-se URI.
- Um Protocolo de acesso para acessar estas fontes, hoje o **HTTP**.
- Uma linguagem de Hypertexto, para a fácil navegação entre as fontes de informação: o **HTML**.



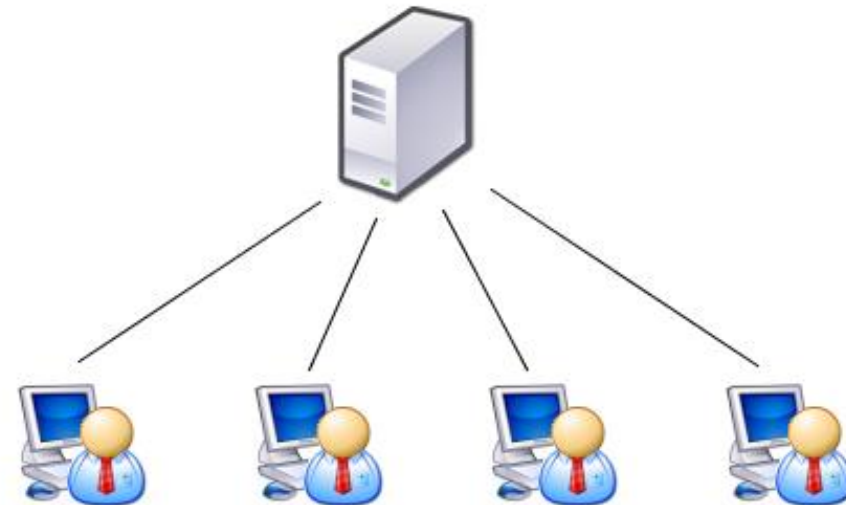
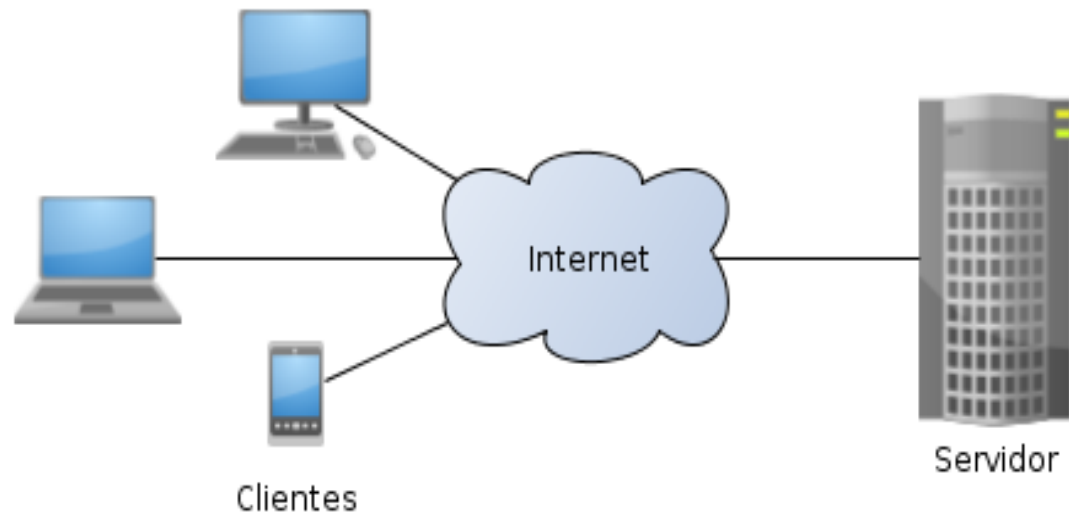


# HTML5

- A partir da versão 5 do HTML, começou a ser fornecido ferramentas para CSS e Javascript trabalhe do melhor modo possível, de forma que o website ou a aplicação continue leve e funcional.
- Também foi adicionado novas *tags*.
- E o mais importante: foi definido um padrão universal do HTML (*tags* e suas funcionalidades)

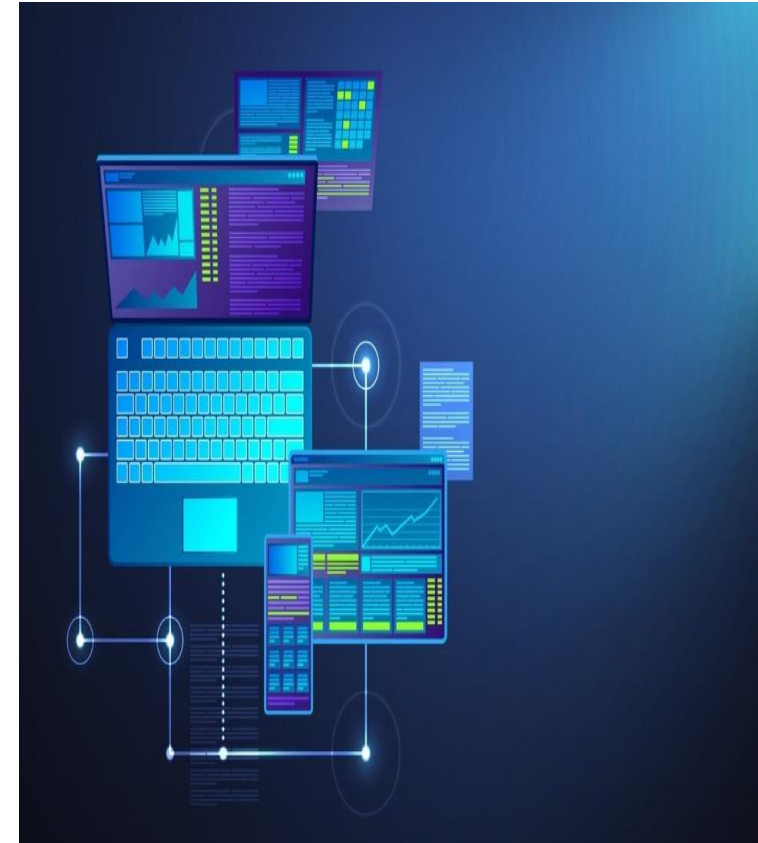


# Arquitetura Cliente-Servidor



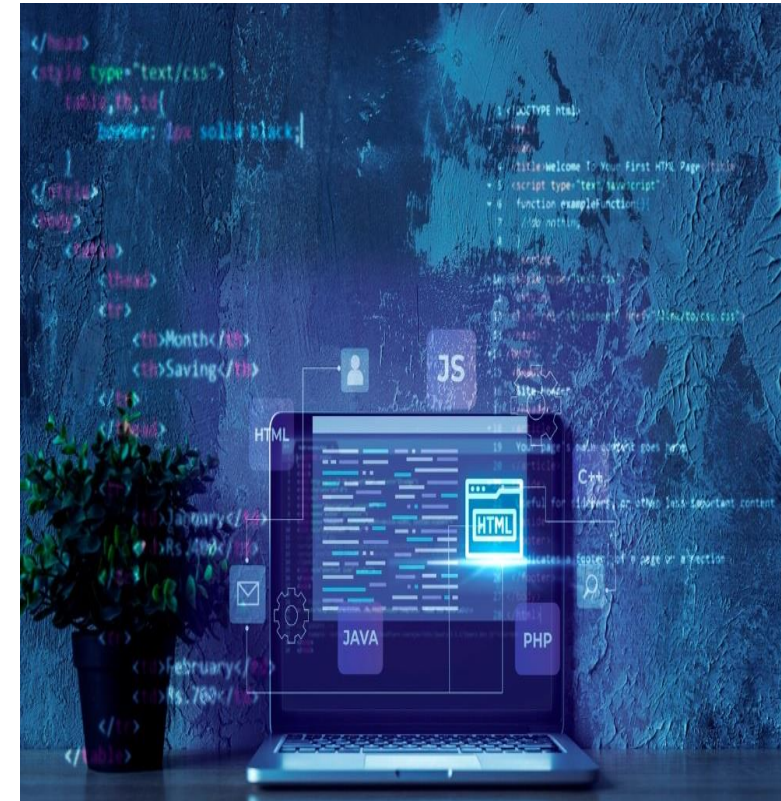
# Arquitetura Cliente-Servidor

- O termo Cliente/Servidor refere-se ao método de distribuição de aplicações computacionais através de muitas plataformas.
- Tipicamente essas aplicações estão divididas entre um provedor de acesso e uma central de dados e numerosos clientes contendo uma interface gráfica para usuários para acessar e manipular dados.



# Arquitetura Cliente-Servidor

- Cliente/Servidor geralmente refere-se a um modelo onde dois ou mais computadores interagem de modo que um oferece os serviços aos outros.
- Este modelo permite aos usuários acessarem informações e serviços de qualquer lugar.



# Arquitetura Cliente-Servidor

- **Cliente:**

- Pode ser denominado como **Front-End** e **WorkStation**;
- É um processo que interage com o usuário através de uma interface gráfica ou não, permitindo consultas ou comandos para recuperação de dados e análise e representando o meio pela qual os resultados são apresentados.





# Arquitetura Cliente-Servidor

- **Cliente:**

- É o processo ativo na relação Cliente/Servidor;
- Inicia e termina as conversações com os Servidores, solicitando serviços distribuídos;
- Não se comunica com outros Clientes;



# Arquitetura Cliente-Servidor

- Servidor:

- Pode ser denominado como **Back-End**;
- Fornece um determinado serviço que fica disponível para todo Cliente que o necessita;
- É o processo reativo na relação Cliente/Servidor;
- Recebe e responde às solicitações dos Clientes;



# Arquitetura Cliente-Servidor

- **Servidor:**

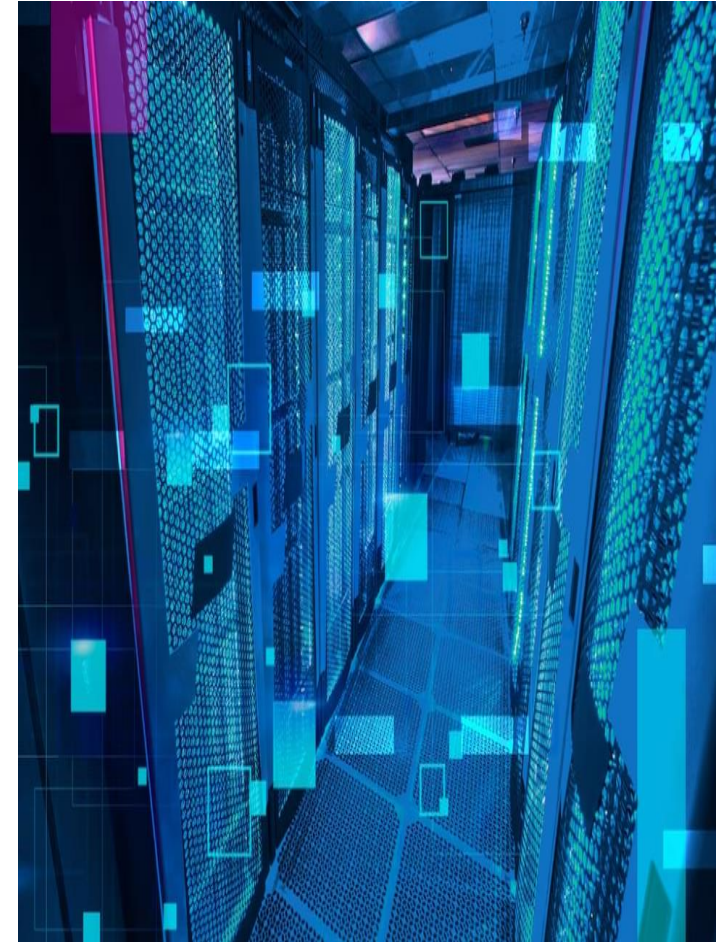
- Não se comunica com outros Servidores enquanto estiver fazendo o papel de Servidor;
- Atende a diversos Clientes simultaneamente;
- Possui execução contínua;



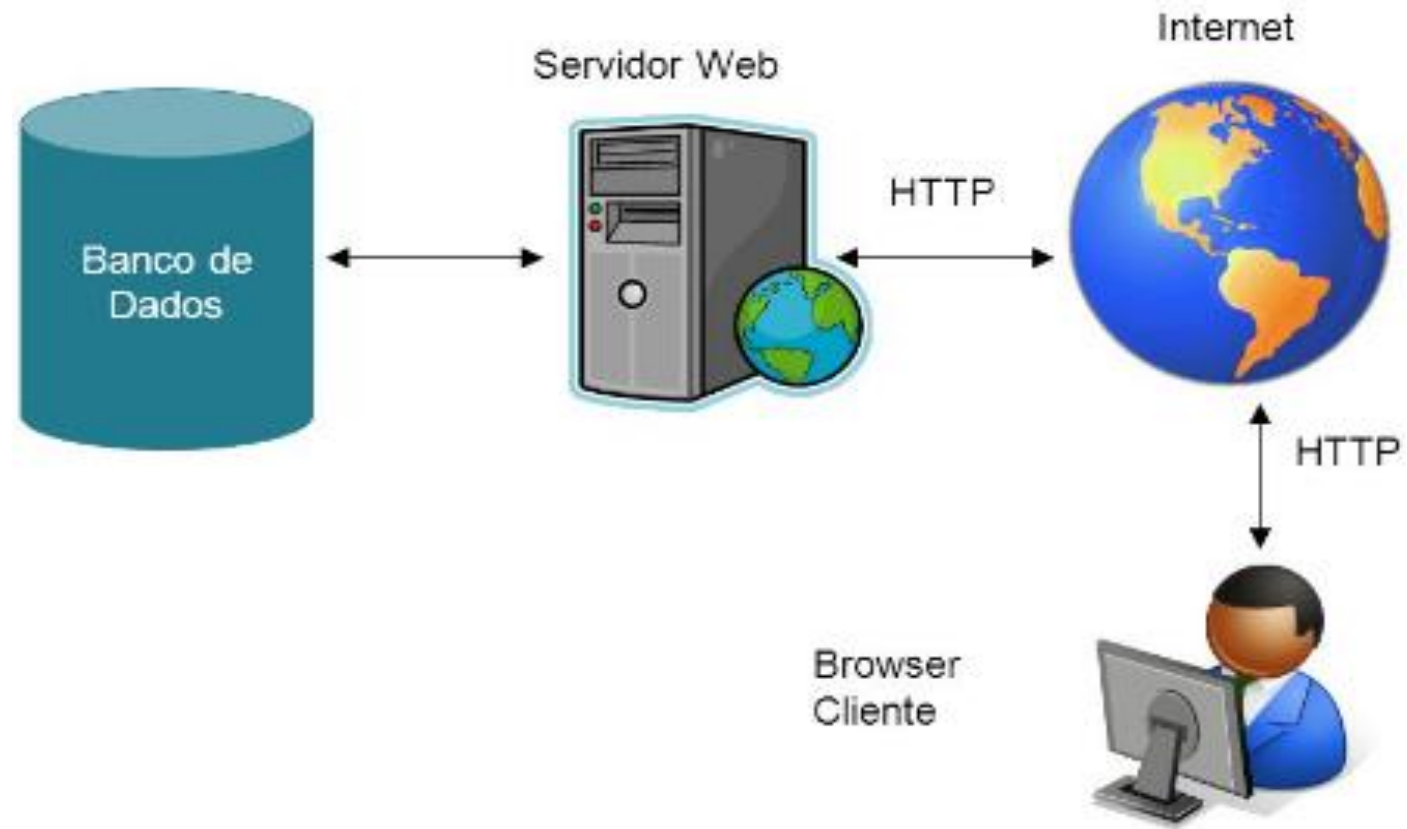


# Servidor

- Alguns tipos de serviços que um Servidor pode proporcionar são:
  - Servidor de Arquivos
  - Servidor de Impressora
  - Servidor de Banco de Dados
  - Servidor de Redes
  - Servidor de Fax
  - Servidor de Processamento e Imagens
  - Servidor de Comunicação
  - e etc.



# Arquitetura Cliente-Servidor



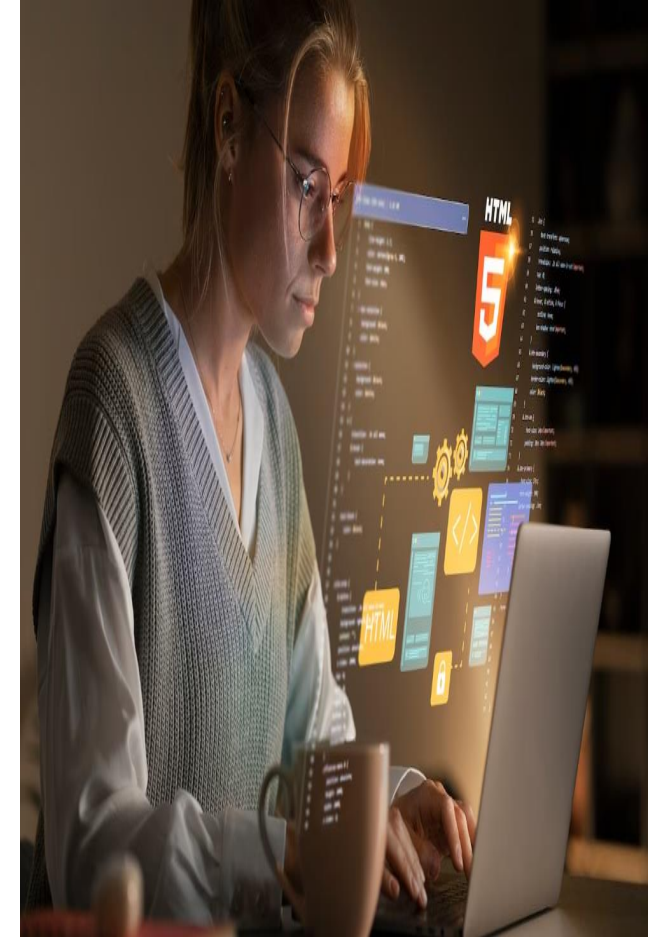
# Front-end

- Responsável pela coleta de informações do cliente e em alguns casos essas informações podem ser processadas.
- Exemplos:
  - HTML
  - CSS
  - Javascript
  - Ajax
  - jQuery



# Back-end

- Responsável por tudo que é executado no servidor, que envolve o plano de negócio do sistema.
- Envolve análise dos dados provenientes do Front-End, acesso ao banco de dados, armazenamento e manutenção de arquivos.
- Exemplos:
  - C#
  - Java
  - Perl
  - PHP
  - Python



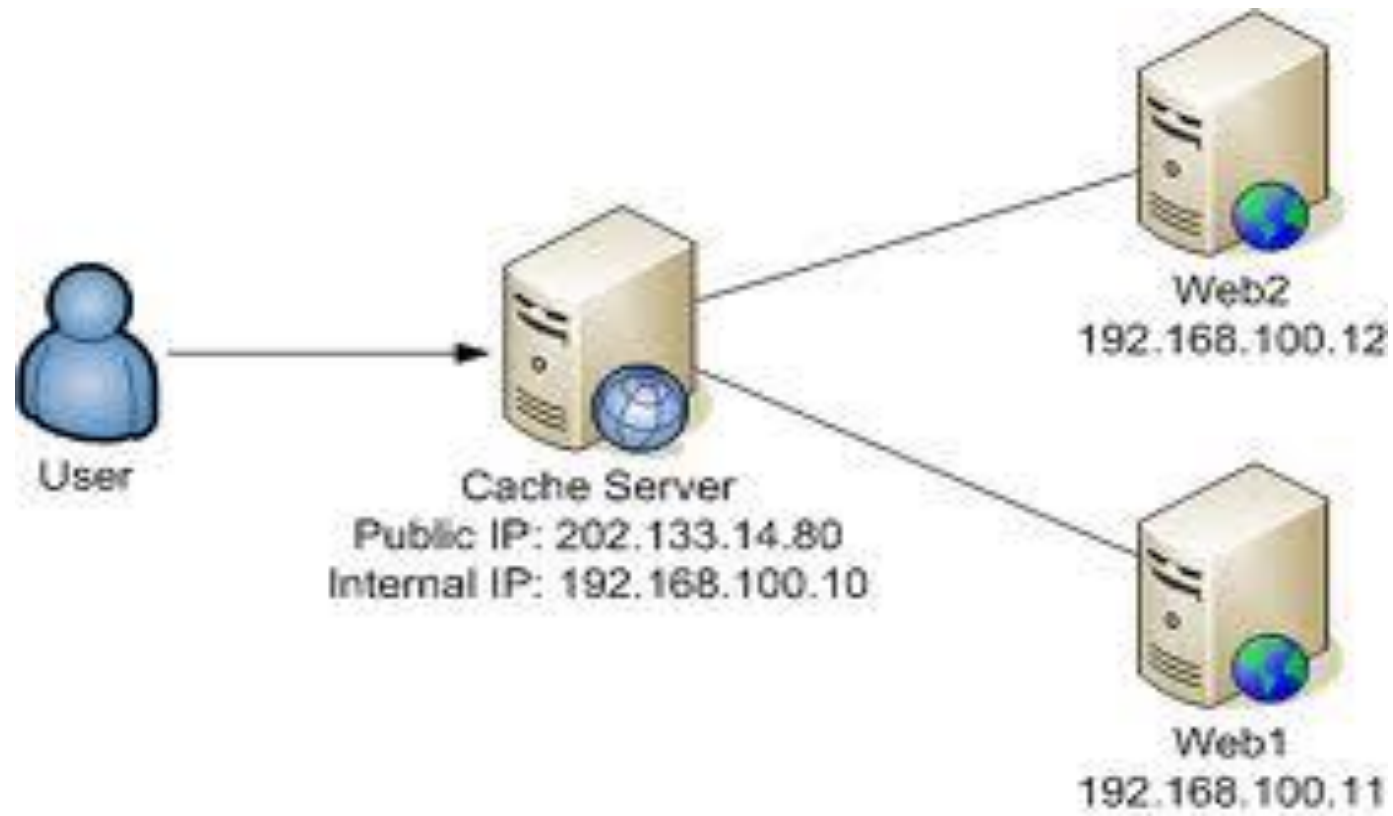
# Fluxo

- Cliente acesso um site pelo endereço, exemplo: [www.globo.com.br](http://www.globo.com.br)
- Um servidor DNS recebe a requisição e encaminha para o endereço IP onde o site está hospedado, no caso: 186.192.90.5
- O servidor do site (IP) processa e encaminha o Front-End ao Browser(navegador do cliente)
- Cliente recebe a página web (html, css, js ...)









# Fluxo





# Fluxo


 MENU


 


 BUSCAR


 Entrar com Conta Globo


  
Siga o canal da  
TV Globo


  
Fale com a Glô  
no Zap


  
Filmes

  
Novelas


  
Séries


  
Criança  
Esperança


  
TV Digital

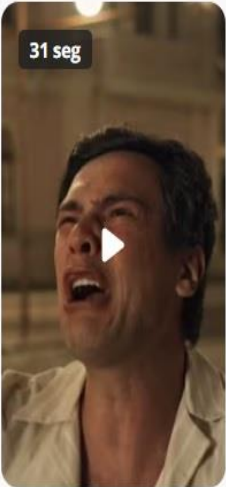



Em Alta na TV Globo


  
41 seg

  
1 min

  
1 min


  
31 seg

  
40 seg


  
1 mi

Programação


11:45  
Jornal do Almoço



13:00  
Globo Esporte RS



13:25  
Jornal Hoje

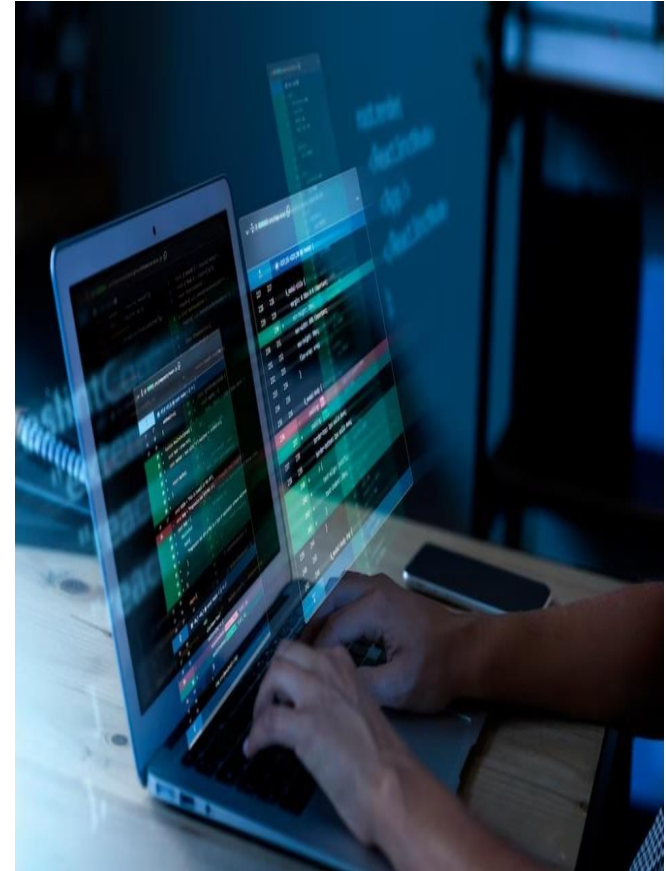


Programação completa →

horário de Brasília

# Fluxo

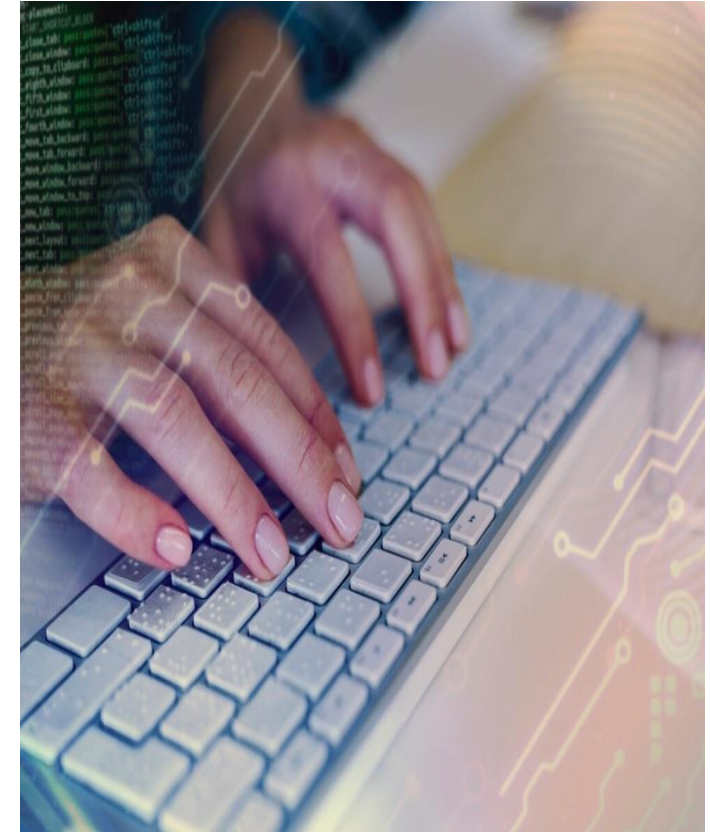
- No site, o cliente navega entre as páginas, cada interação do usuário, o processo se repete, ele envia uma requisição ao servidor e o servidor responde a essa requisição alterando a página do cliente.
- Efetuando assim, uma série de “conversas” entre cliente e servidor.





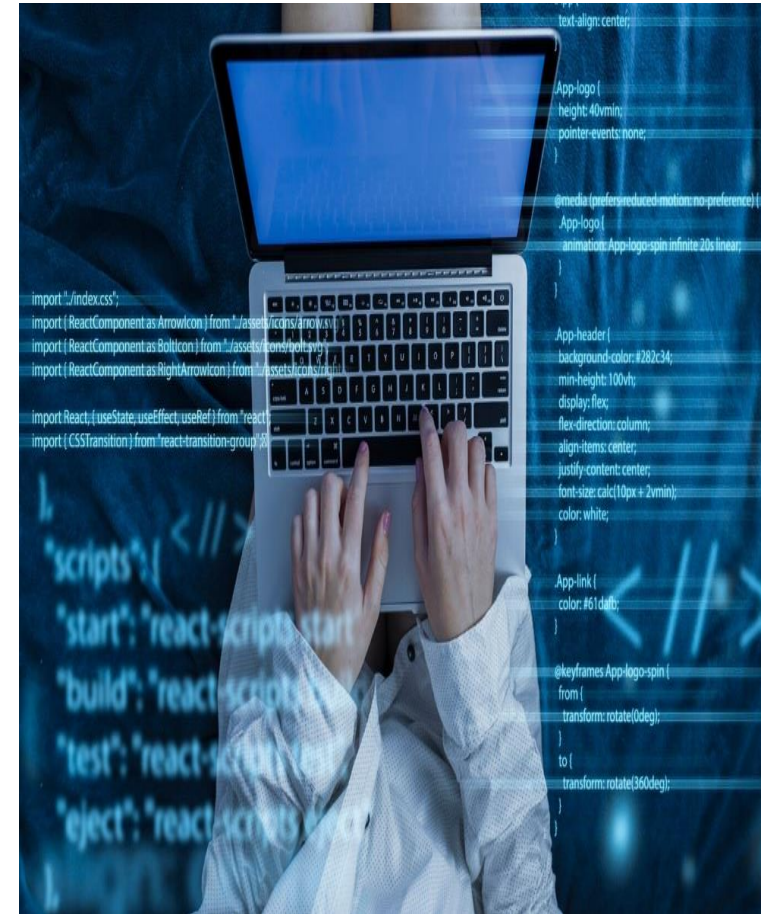
# Fluxo

- Em uma das páginas, existe um formulário de cadastro, o usuário preenche todos os campos, mas a data de nascimento foi inserida com um valor errado, exemplo: 22/13/1992. Só que, antes de submeter a requisição ao servidor, o programador elaborou uma validação dos campos via **JavaScript** no qual foi identificada a data inválida e o usuário foi alertado. Esse fluxo não “saiu” do Front-End.
- Após o usuário efetuar a correção e submeter o formulário novamente, a requisição será enviada ao servidor.



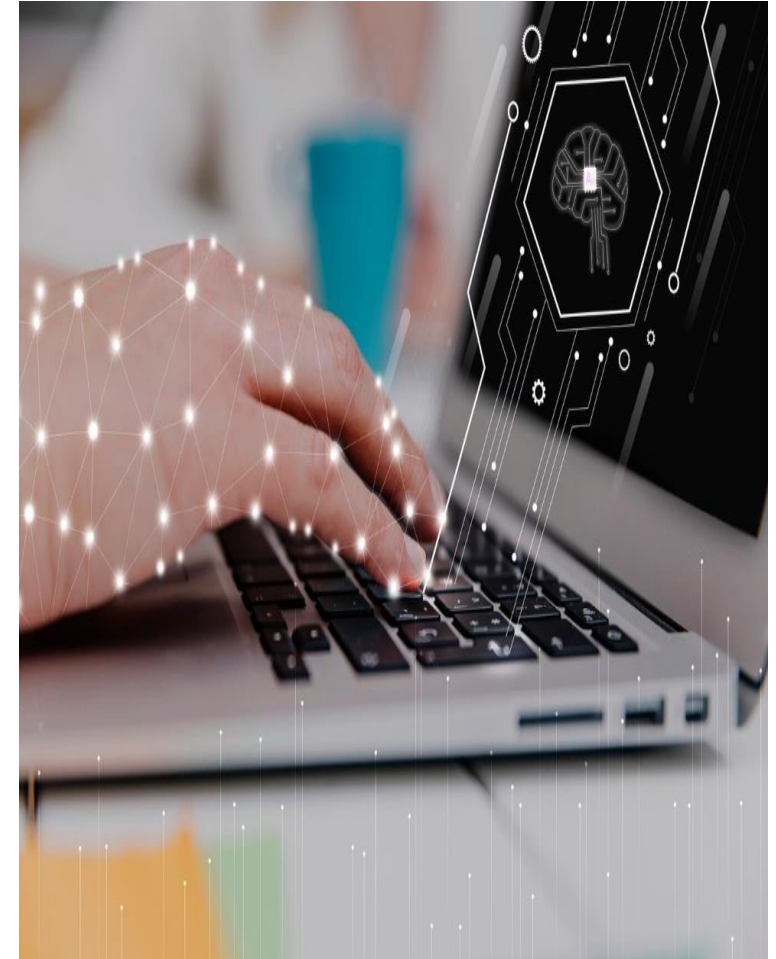
# Fluxo

- O servidor pode encaminhar páginas HTML com arquivos CSS e JS inteiros, tais como são ao cliente, e seus navegadores (browsers) que efetuam a interpretação destes códigos para o cliente.
- Estas páginas são denominadas ESTÁTICAS.

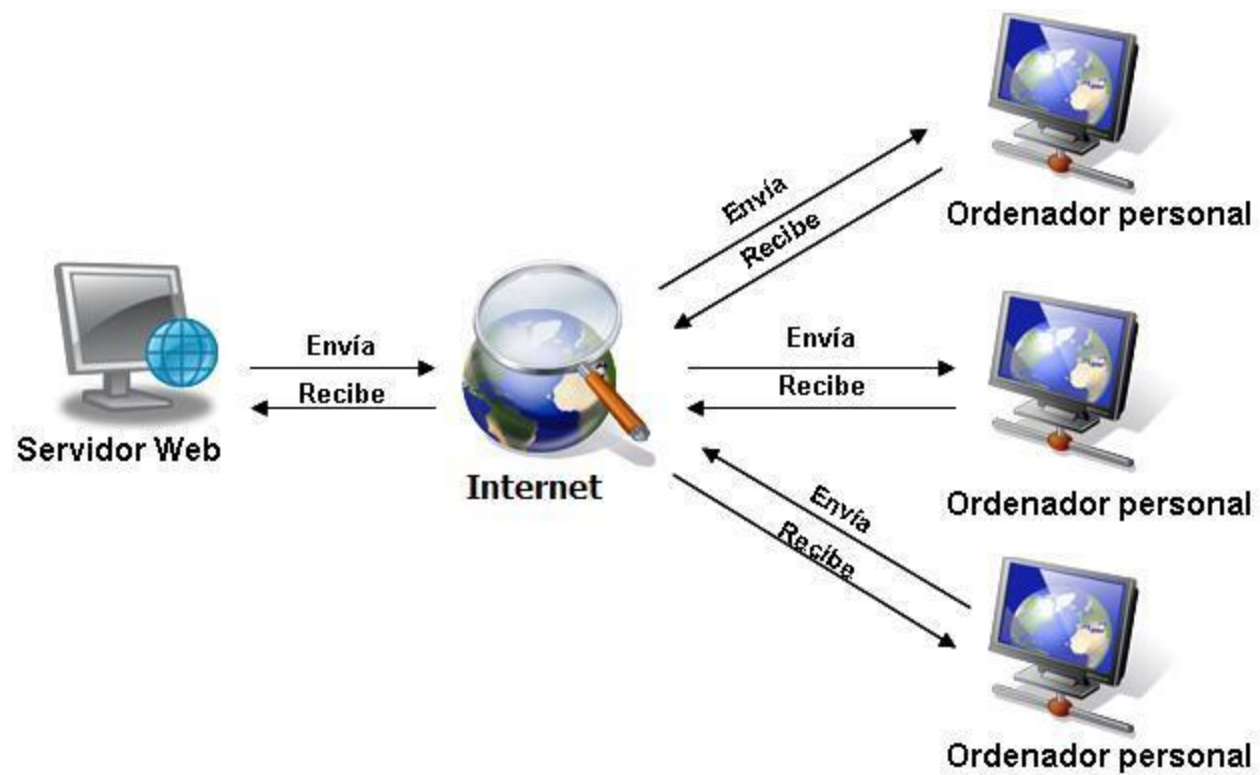


# Fluxo

- Mas, muitas vezes, toda a página ou pequenas partes delas só ficam “prontas” após o servidor processar, como por exemplo, buscar informações do banco de dados ou de arquivos.
- Estas páginas são denominadas DINÂMICAS.



# Flujo



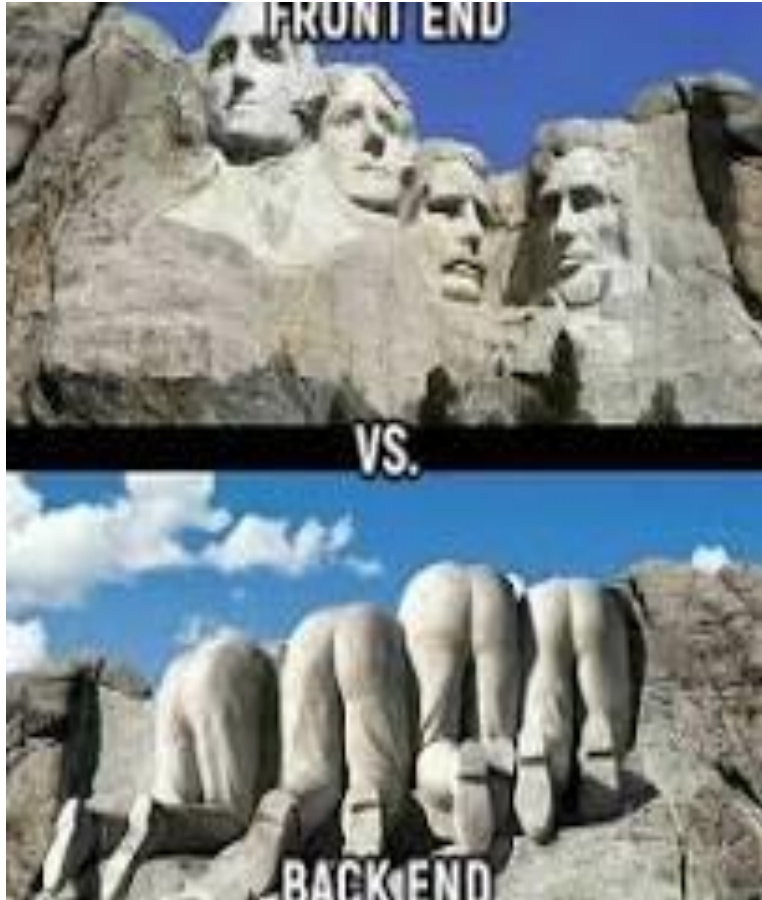


# BACK-END X FRONT-END



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

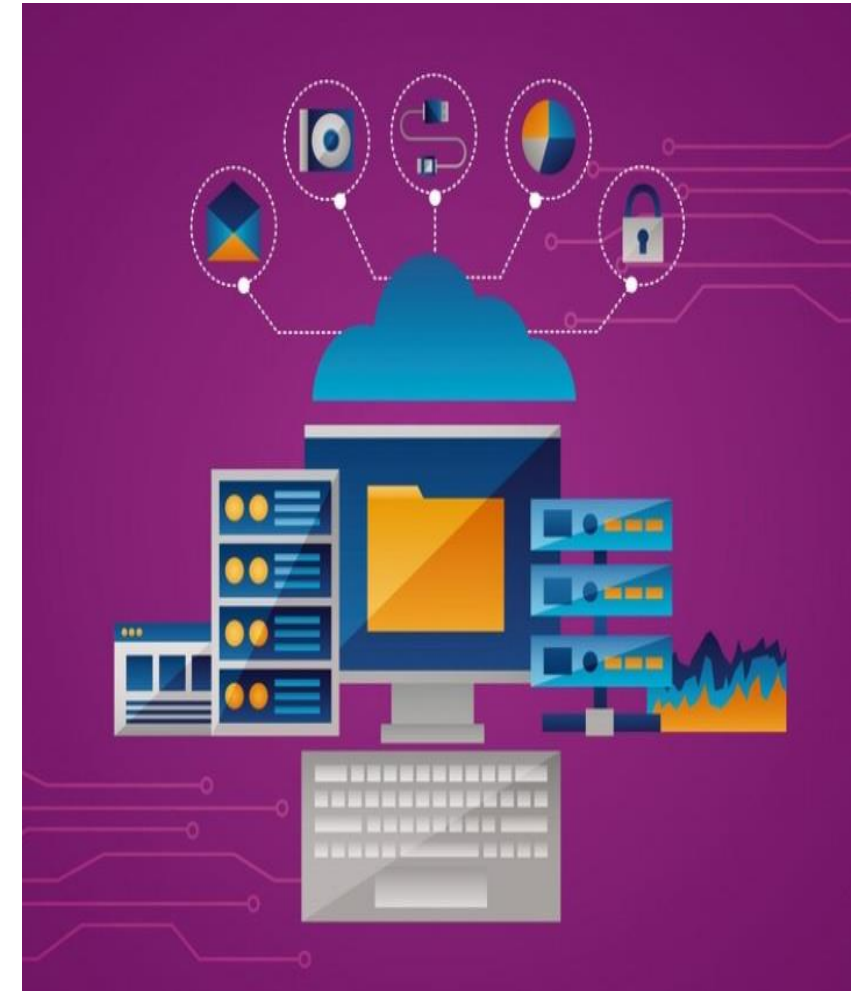
# BACK-END X FRONT-END



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

# Hospedagem

- Ao publicar um site, devemos pensar qual plataforma iremos utilizar para hospedar nosso site.
- Ex: Locaweb, Hostgator, Hostinger, Infinity free, entre outros.
- Devemos levar em conta, a linguagem utilizada no nosso Back-End.





# Hospedagem – Escolha do servidor

- Linguagens .Net (C#, VB...)
  - IIS (Serviços de Informações da Internet)
  - Ambiente Windows
- PHP
  - Apache
  - Windows e Linux
- Java
  - TomCat
  - Windows e Linux
- Python
  - Apache
  - Windows e Linux



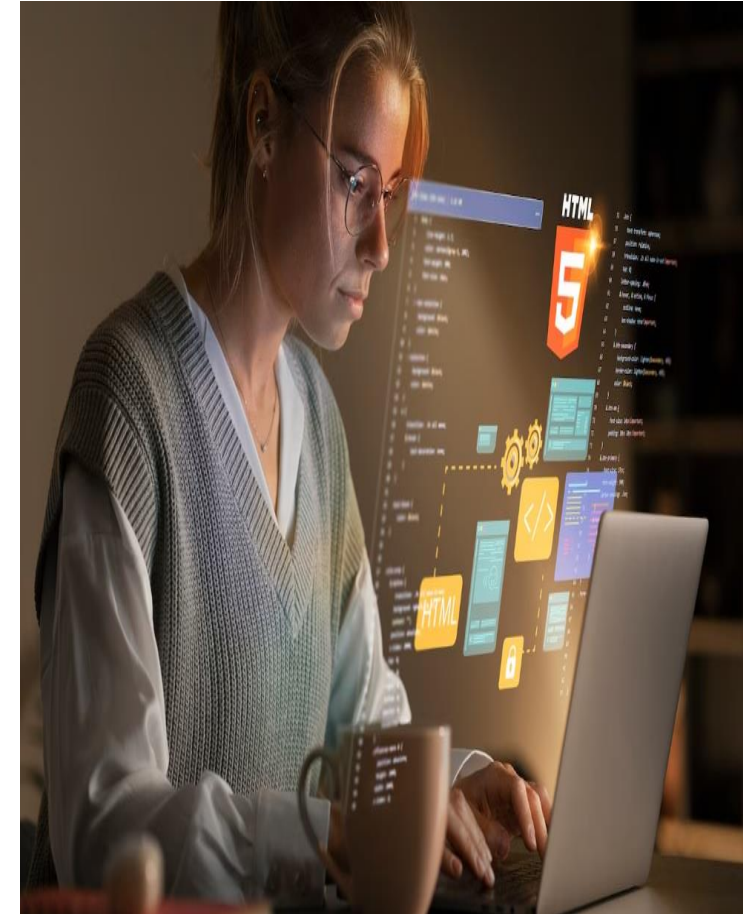


# HTML



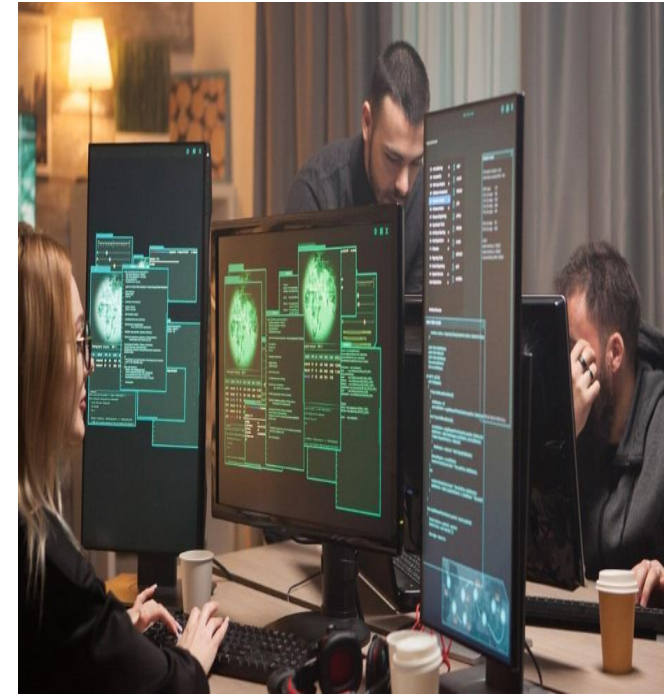
# Tags básicas

- **<html>**: define o **início** de um documento HTML e indica ao navegador que **todo conteúdo posterior deve ser tratado como uma série de códigos HTML**;
- **<head>**: define o **cabeçalho** de um documento HTML, que traz informações sobre o documento que está sendo aberto;
- **<body>**: define o conteúdo principal, o **corpo** do documento. Esta é a **parte do documento HTML que é exibida no navegador**. No corpo podem-se definir atributos comuns a toda a página, como cor de fundo, margens, e outras formatações.



## <body>

- **bgcolor**: cor do plano de fundo. Deve ser usado o código hexadecimal ou o nome da cor (conforme tabela vista mais adiante), assim como em todos os atributos de cores.
- **background**: URL de uma imagem para ser usada como plano de fundo.
- **link**: cor natural dos links, ou seja, enquanto eles ainda não foram clicados (o padrão é azul).
- **alink**: cor dos links quando são clicados (o padrão é vermelho).
- **vlink**: cor dos links após serem visitados (o padrão é roxo).
- **text**: cor do texto da página.





# Exemplo

```
<html>  
  <head>  
    ...  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```



# Doctype

- Para usar os novos elementos de HTML5, devemos informar o *Doctype* corretamente, como segue abaixo:

```
<!DOCTYPE html>
```

```
<html lang="pt-br">
```

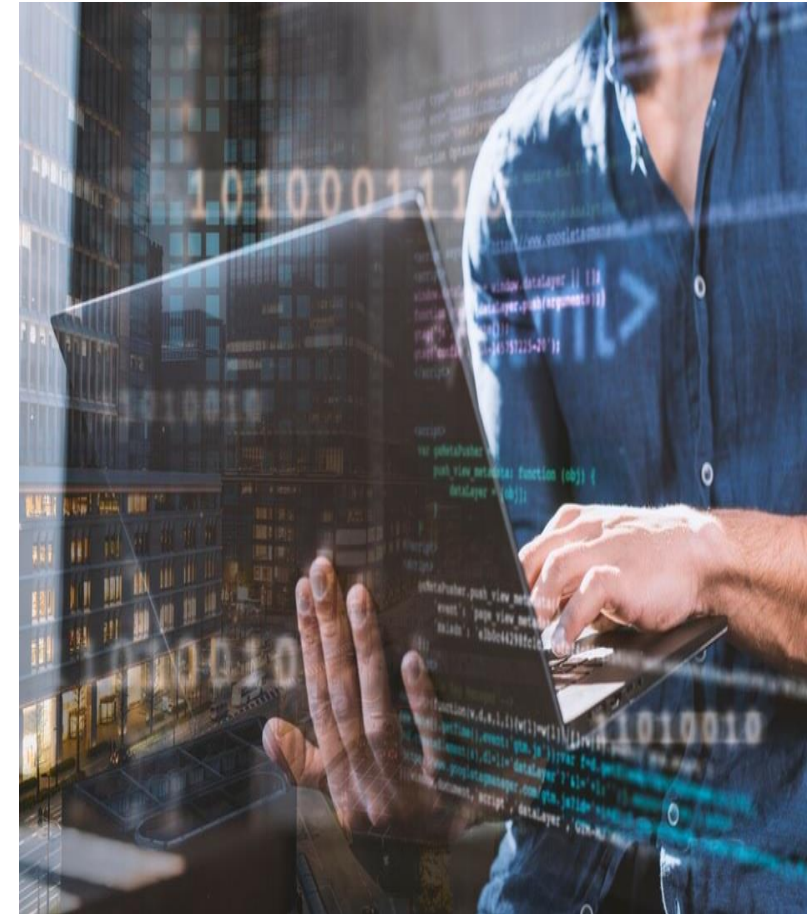
•  
•  
•



# Cabeçalhos

- Podem incluir pauta de conteúdos desde o logotipo da caixa de busca. No blog colocaremos apenas o título.
  - Uma página pode conter mais de um cabeçalho, em seção ou artigos, por exemplo. Por isso é interessante usar o atributo ID (falaremos mais sobre ele em breve).

```
<body>  
  <header id="cabecalho">  
    <h1>Nosso Blog!</h1>  
  </header>  
</body>
```



# Rodapés

- O elemento *footer* define informação de rodapé para um documento ou uma seção adjacente. Por isso também é interessante usar ID para diferenciar uns dos outros.

```
<footer id="rodape">
```

```
    <p>© 2018 Nosso Blog.</p>
```

```
</footer>
```





# Navegação

- A navegação é essencial, por isso foi criada a nova tag para trata-la adequadamente no HTML5. Exemplo:

```
<header id="cabecalho">
```

```
<h1>Nosso Blog!</h1>
```

```
<nav>
```

```
<ul>
```

```
<li><a href="/">Últimas Postagens</a></li>
```

```
<li><a href="https:\\www.google.com.br">Google</a></li>
```

```
<li><a href="#Posts">Posts</a></li>
```

```
<li><a href="contact">Contato</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```





# Seções e Artigos

- Seções são as regiões lógicas de uma página
- O elemento *section* veio para substituir o uso excessivo de *divs*.
- No entanto, não devemos nos deixar levar pelas seções, devemos usá-las para agrupamento lógico do conteúdo.
- No caso de um blog, será criada uma seção que conterá todas as postagens.
- Todavia, cada postagem não deve estar em uma nova seção, pois tem um tag mais apropriado para isso!



# Seções e Artigos

```
<section id="posts">  
  CONTEUDO DA POSTAGEM  
</section>
```



# Artigos

- O tag *article* é o elemento perfeito para descrever o real conteúdo de uma página WEB.
  - Cada artigo pode conter cabeçalho, conteúdo e rodapé



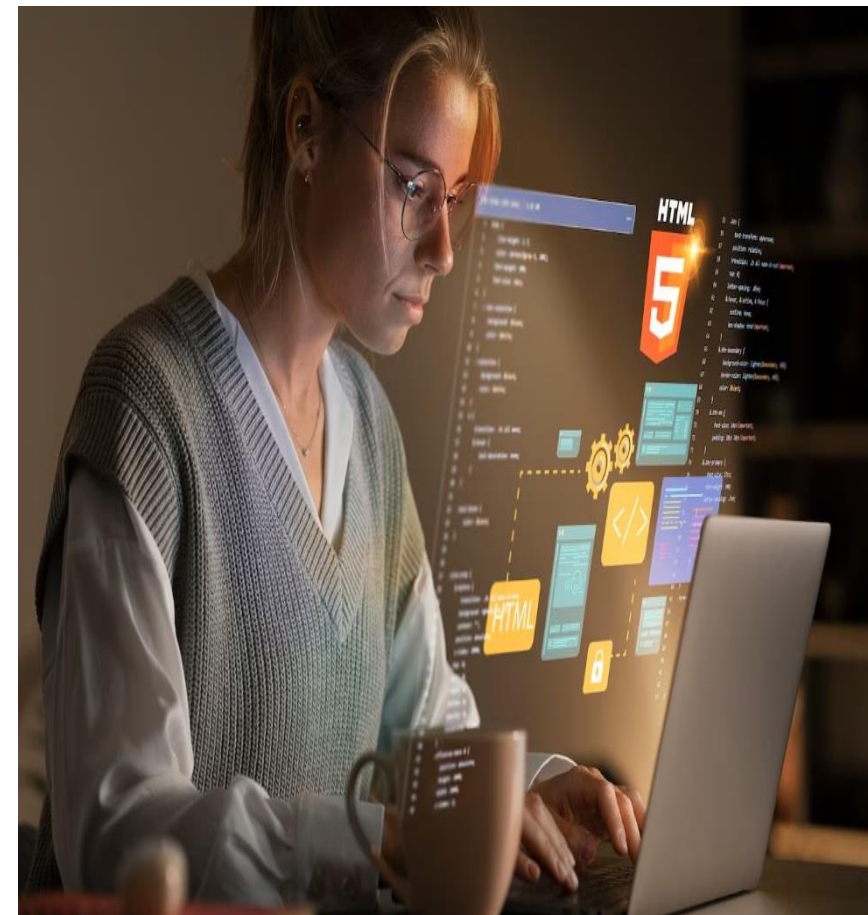


# Artigos exemplo

...

```
<section id="posts">
  <article class="post">
    <header>
      <h2>O que é um Blog?</h2>
      <p>Postado por Wikipedia
        <time datetime="2013-12-13T14:39"> em 13 de fevereiro, 2024, às 14h
          39min</time>
      </p>
    </header>
    <p>Muitos blogs fornecem comentários ou notícias sobre um assunto em particular;
    outros funcionam mais como diários online. Um blog típico combina texto, imagens e
    links para outros blogs, páginas da Web e mídias relacionadas a seu tema. A capacidade
    de leitores deixarem comentários de forma a interagir com o autor e outros leitores é
    uma parte importante de muitos blogs. </p>
    <footer>
      <p><a href="comments"><i>25 Comentários</i></a> ...</p>
    </footer>
  </article>
</section>
```

...



# Cabeçalhos de Texto

<h1>Título de nível 1</h1>

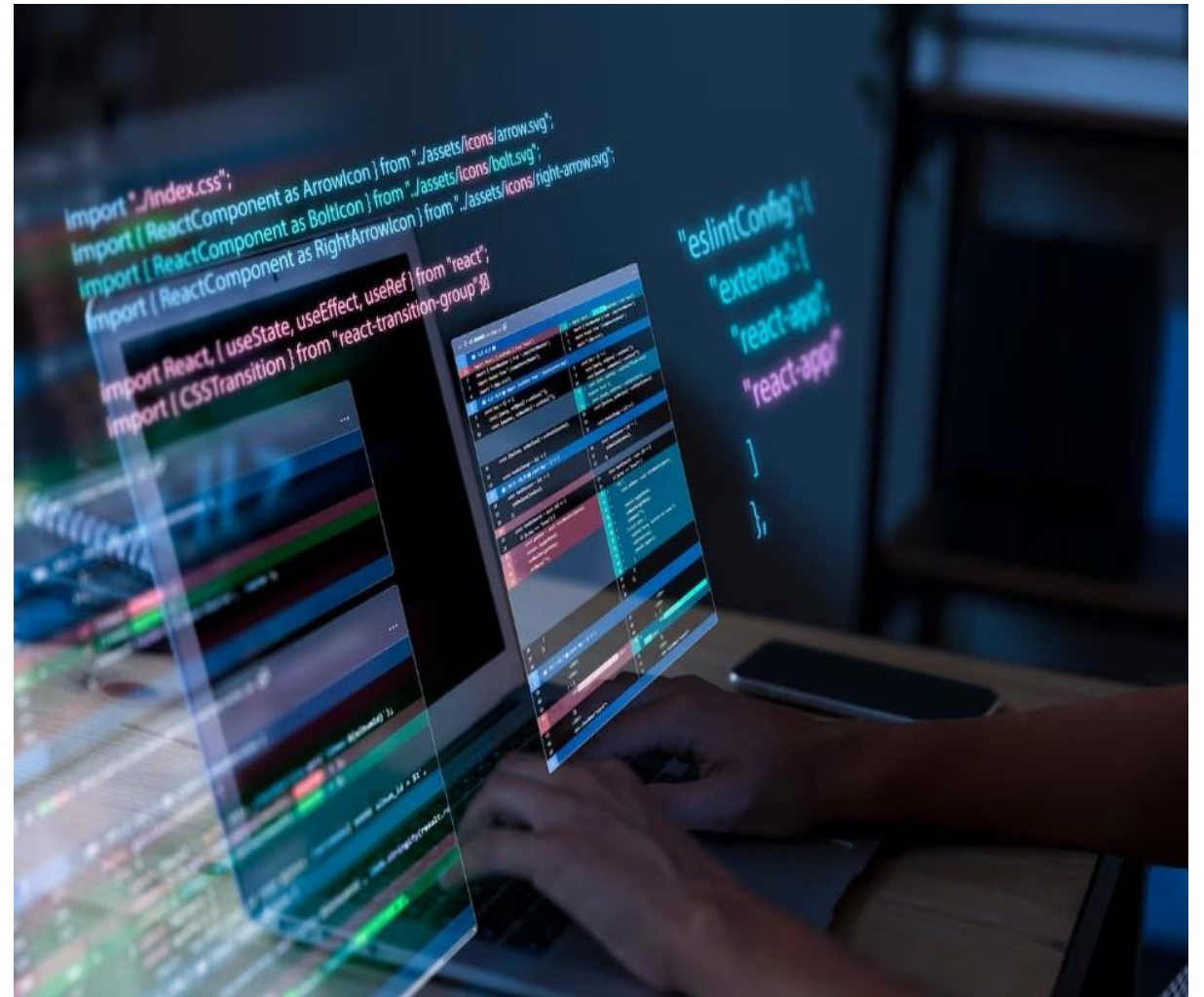
<h2>Título de nível 2</h2>

<h3>Título de nível 3</h3>

<h4>Título de nível 4</h4>

<h5>Título de nível 5</h5>

<h6>Título de nível 6</h6>





# Alinhando os Cabeçalhos

- É possível alterar o alinhamento dos títulos definindo seu **atributo** “align” com um dos seguintes valores: *center*, *left* e *right*;

<h1 align='center'>Título de nível 1 - centralizado</h1>

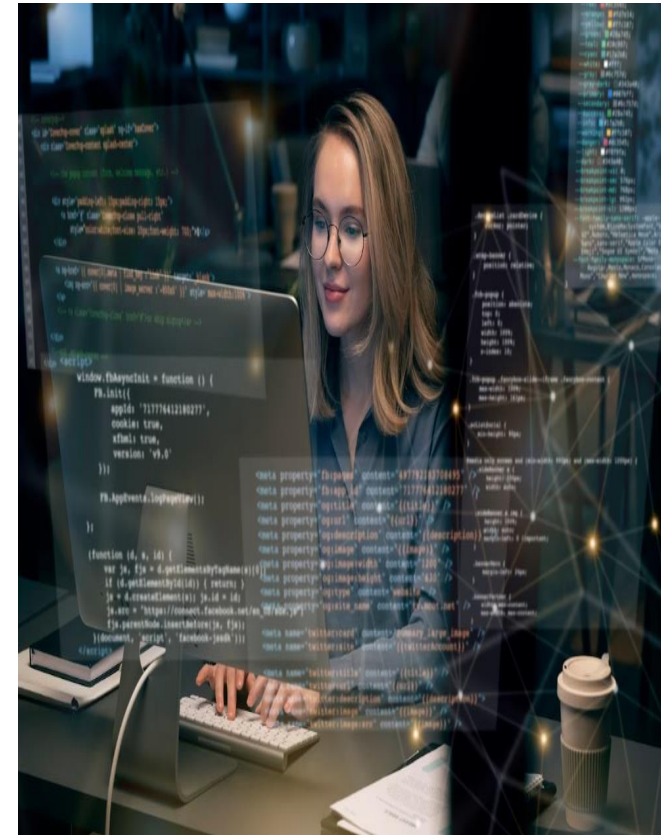
<h2 align='left'>Título de nível 2 - esquerda</h2>

<h3 align='right'>Título de nível 3 - direita</h3>



# Tag <font> - tamanho e cor do texto

- Usando a *tag* <font> é possível alterar algumas das características primordiais do texto como o tipo da fonte (Arial, Times New Roman, etc), a cor e o tamanho.
- Essas propriedades são alteradas com a definição dos atributos face, color e size, respectivamente, da tag font.



# Exemplos com a tag <font>

<font face="Arial" size="3" color="blue">Arial 3 Azul</font>

<font face="Times" size="4" color="green">Times 4 Verde</font>

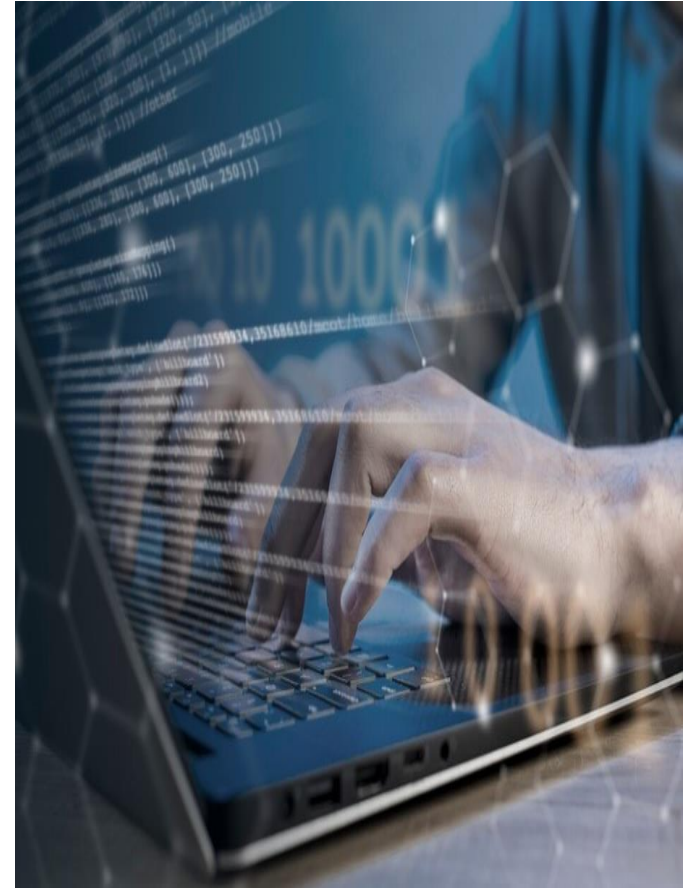
<font face="Courier" size="5" color="red">Courier 5 Vermelho</font>



# Formatação adicional do texto

- Existem algumas tags bastante úteis que permitem aplicar uma formatação a um trecho do texto, apenas adicionando as tags de abertura e fechamento antes e depois do texto a ser formatado, assim como foi visto para a tag <font>.
- <b></b>: marca o texto como em negrito (bold).
- <i></i>: marca o texto como em itálico (italics).
- \*<u></u>: marca um texto como sublinhado (underlined).
- \*<s></s> ou <strike></strike>: marca um texto como riscado.

*Observação: as tags marcadas com asterisco (\*) foram descontinuadas na versão 5 da HTML, ou seja, seu uso **não é mais indicado**.*

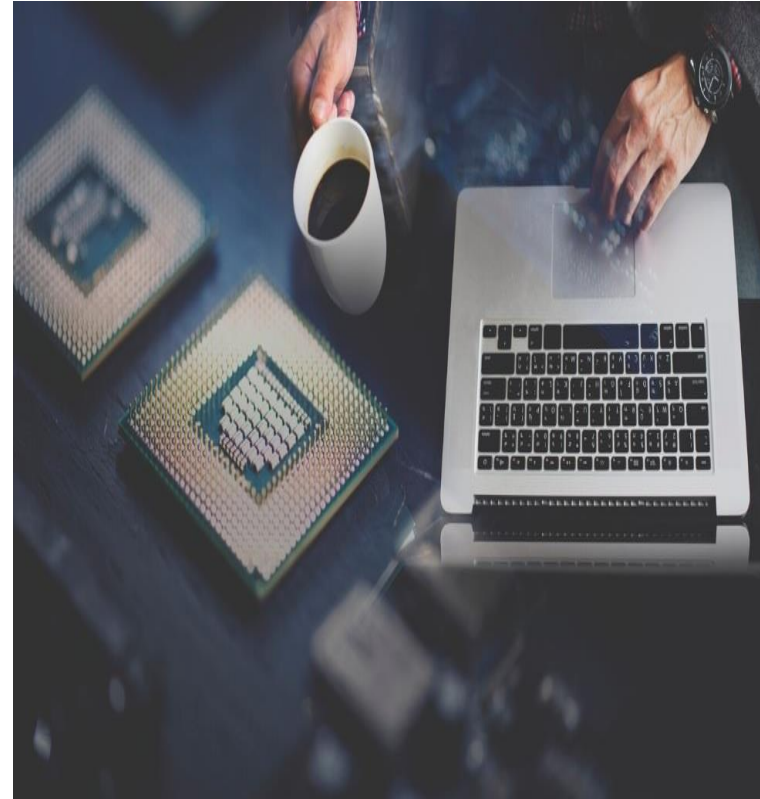




# Formatação adicional do texto – cont.

- `<sub></sub>`: marca um texto como subscrito.
- `<sup></sup>`: marca o texto como sobrescrito.
- `*<center></center>`: centraliza o texto.
- `<sub>subscrito</sub>`
- `<sup>sobrescrito</sup>`

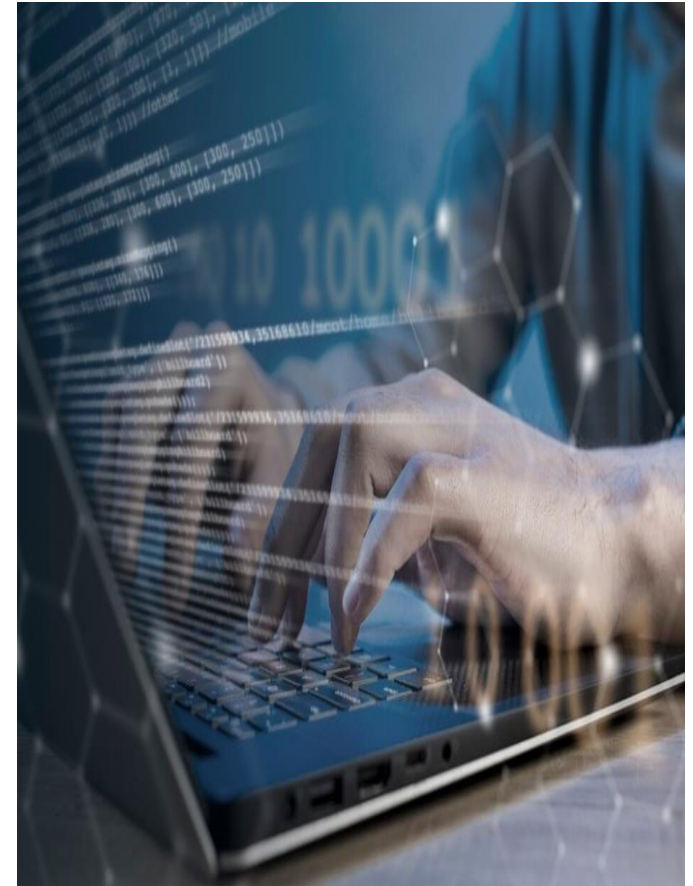
*Observação: as tags marcadas com asterisco (\*) foram descontinuadas na versão 5 da HTML, ou seja, seu uso não é mais indicado.*





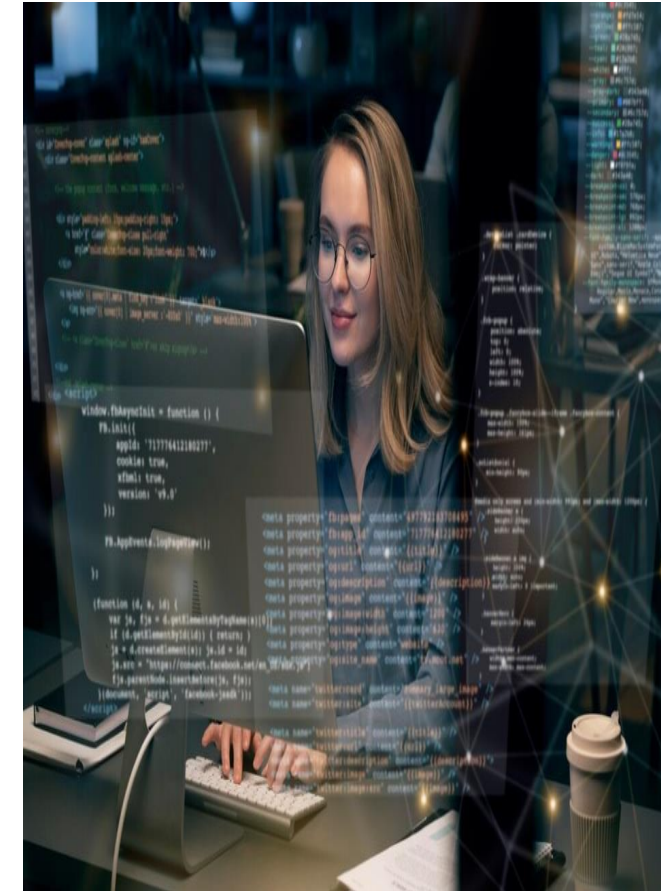
# Parágrafos e quebras de linha

- `<br/>` - quebra de linha;
- `<p></p>` - sinalização de um parágrafo;



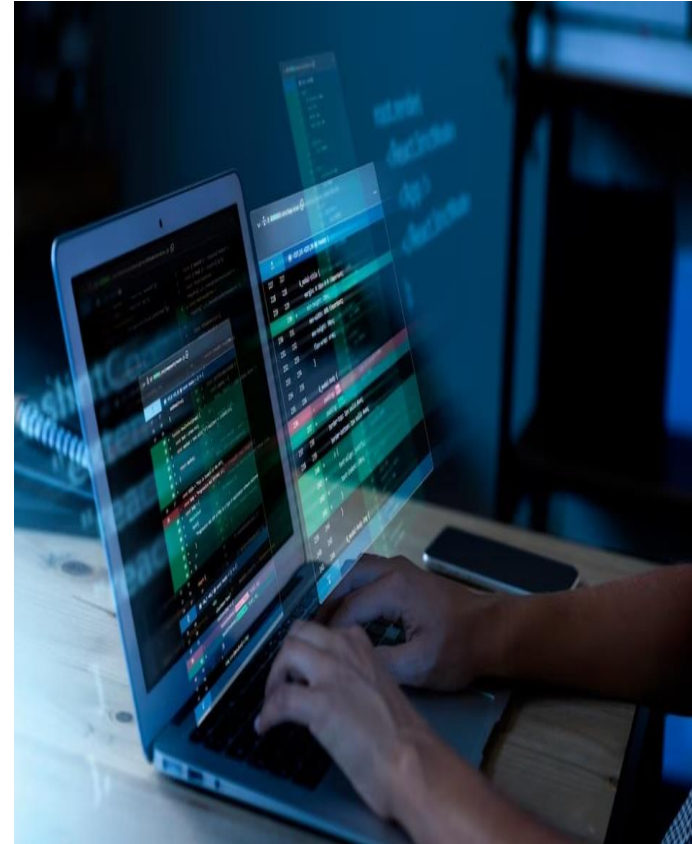
# Alinhamento de Parágrafos

- Podemos ainda alterar seu alinhamento para atender às diversas situações que surgem no dia-a-dia. Isso pode ser feito com a definição do atributo “align” da tag <p>, com um dos seguintes valores: *left* (alinhado à esquerda), *right* (alinhado à direita), *center* (centralizado) e *justify* (justificado).
- <p align=“left/right/center/justify”></p>



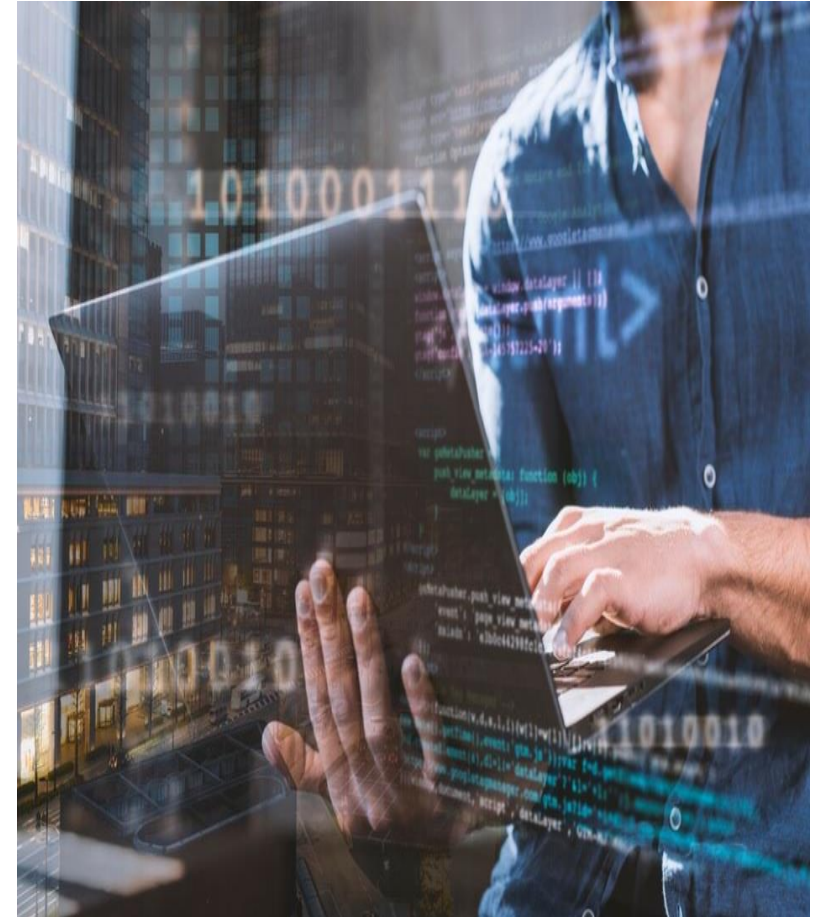
# Imagens

- Inserir **imagens** na página é uma das necessidades mais comuns e, por esse motivo, também é consideravelmente simples de ser feito. Para esse fim existe a tag **<img>**, cujos atributos são mostrados a seguir:
- **src**: caminho completo do arquivo de imagem (incluindo a extensão do arquivo).
- **alt**: texto alternativo para a imagem, geralmente uma descrição da mesma.
- **width**: largura da imagem em pixels.
- **height**: altura da imagem em pixels.



# Imagens

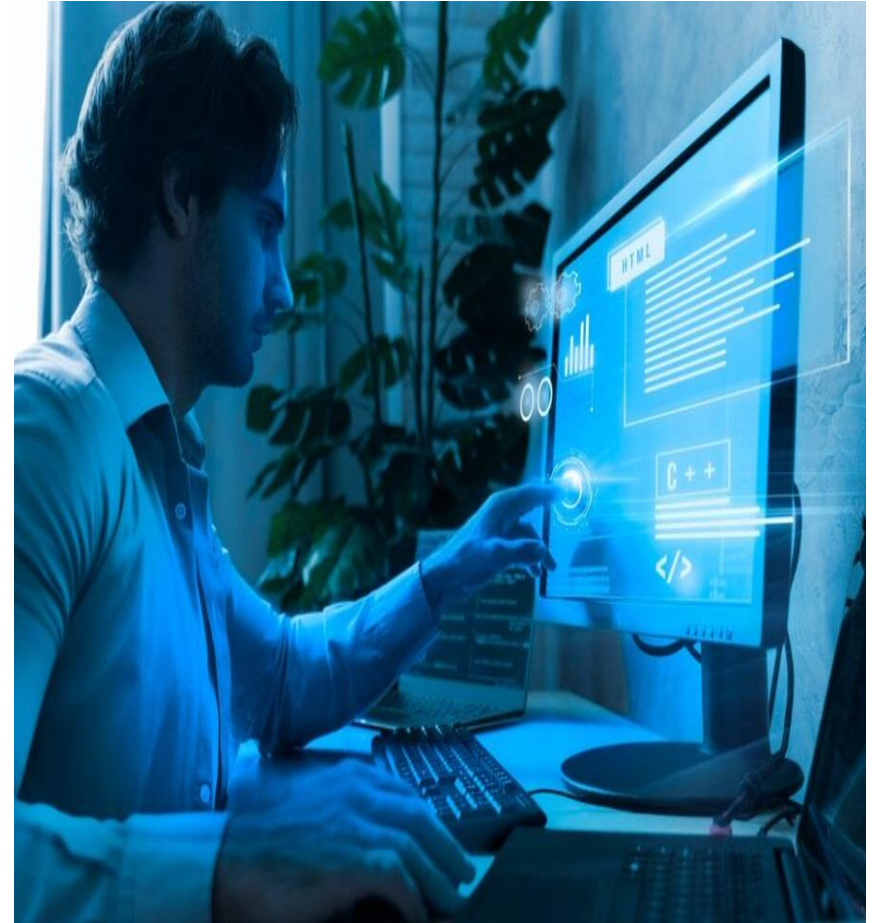
- **Observação:** os atributos **src** e **alt** são obrigatórios para a tag `<img>`. Além disso, a tag deve ser fechada nela mesma, ou seja, a sintaxe mínima é ``





# Inserindo Imagens

- ``
- ``
- ``





# Observações sobre as imagens

- Caso **não sejam definidas** a largura e/ou a altura, a imagem será adicionada à página em seu **tamanho original**.
- Ao alterar apenas uma das dimensões, a outra é redefinida proporcionalmente. Caso se deseje alterar tanto a largura quanto a altura com valores específicos, os dois atributos devem ser definidos.

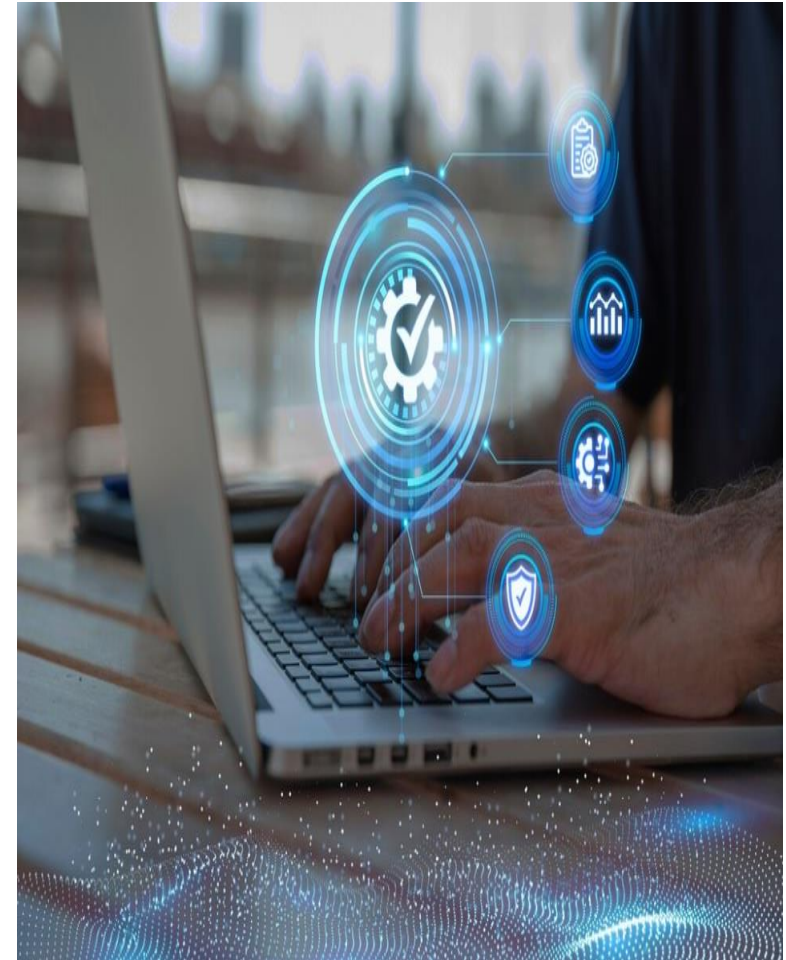


# Criando Links

```
<a href='https://www.google.com'>Meu Link</a>
```

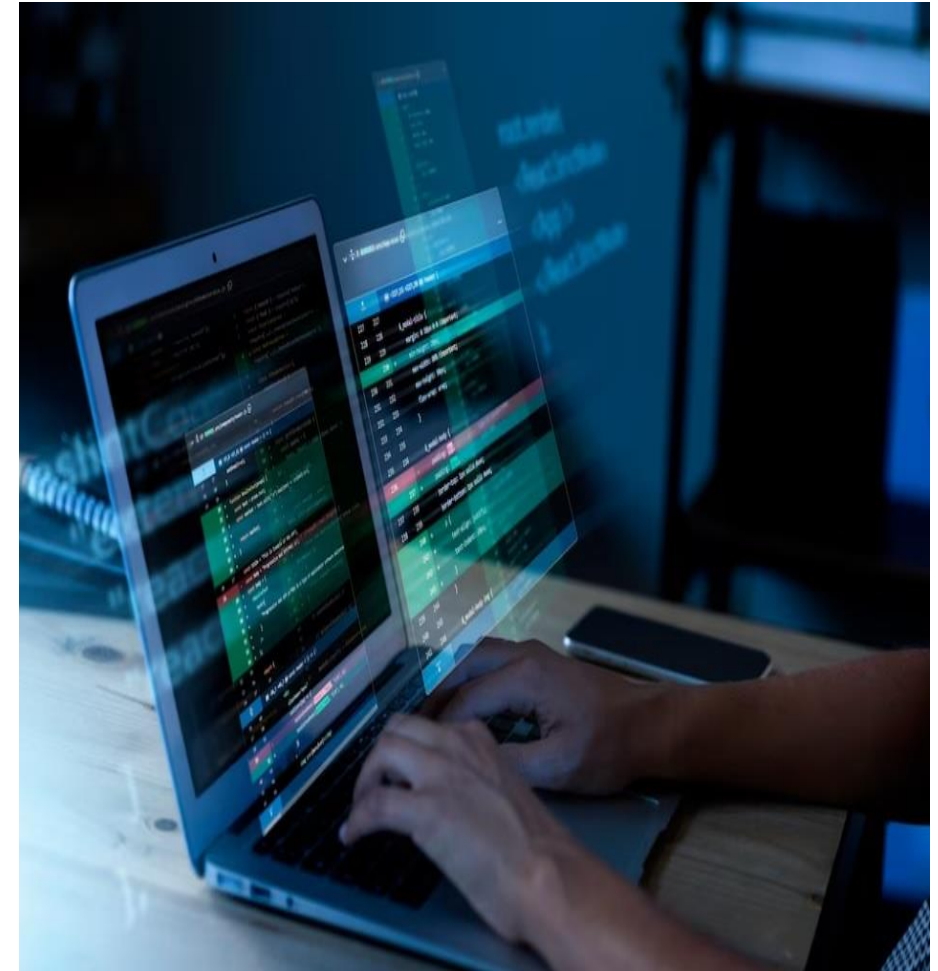
```
<a href='http://www.ufn.edu.br/'>  
    
</a>
```

```
<a href='www.terra.com.br'>  
  <h1>Titulo como link</h1>  
</a>
```



# Listas

- Outra estrutura bastante comum de ser utilizada em páginas web é a **lista**, que serve pra **organizar um conjunto de itens, sequencialmente ou não**. As listas podem ser ordenadas ou não, ou seja, cada item podem ou não ter um número/letra/símbolo que o identifique sequencialmente.
- As tags para listas ordenadas e não ordenadas são, respectivamente, `<ol></ol>` e `<ul></ul>`. Entre essas marcações devem ser inseridos os itens, que levam a tag `<li></li>`



# Usando listas

- Listas ordenadas:

`<ol>`

`<li>Primeiro item</li>`

`<li>Segundo item</li>`

`<li>Terceiro item</li>`

`</ol>`

- 1.Primeiro item
- 2.Segundo item
- 3.Terceiro item

- Listas não ordenadas:

`<ul>`

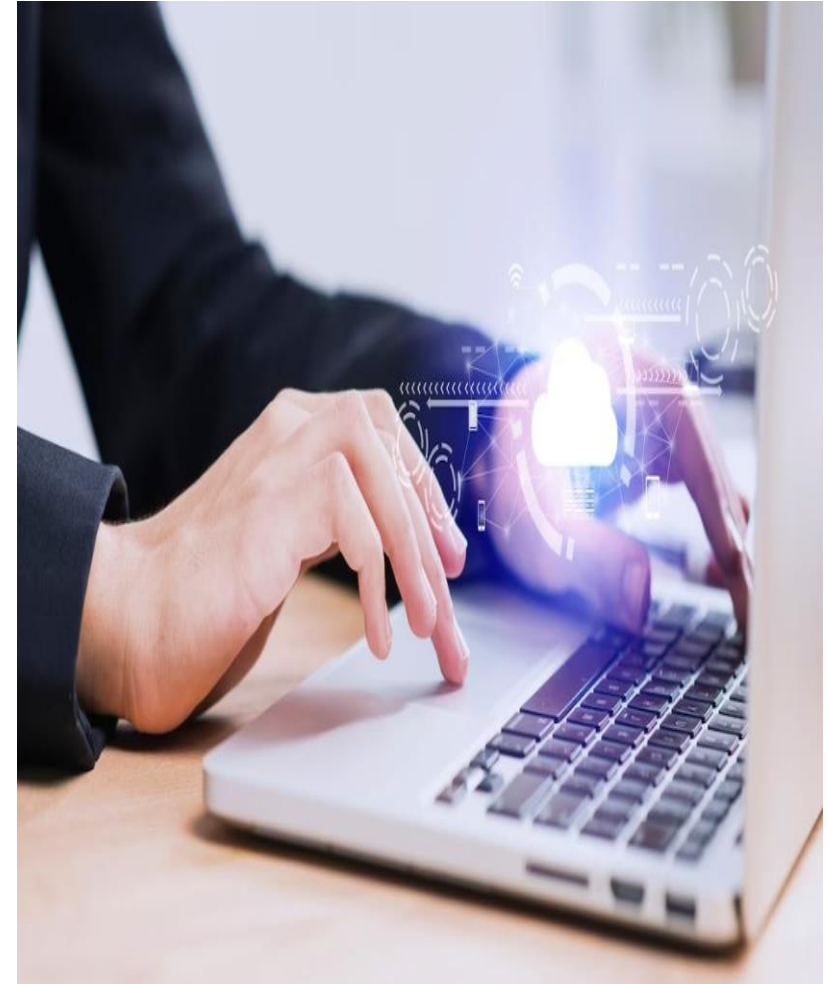
`<li>Primeiro item</li>`

`<li>Segundo item</li>`

`<li>Terceiro item</li>`

`</ul>`

- Primeiro item
- Segundo item
- Terceiro item





# Lista de Definição

- É usada para apresentar itens com um título seguido de sua definição. A tag principal desse tipo de lista é a `<dl></dl>`, enquanto os itens são compostos por duas tags: `<dt></dt>` para o título e `<dd></dd>` para a definição.

`<dl>`

`<dt>Item 1</dt>`

`<dd>Definicao do item 1</dd>`

`<dt>Item 2</dt>`

`<dd>Definicao do item 2</dd>`

`<dt>Item 3</dt>`

`<dd>Definicao do item 3</dd>`

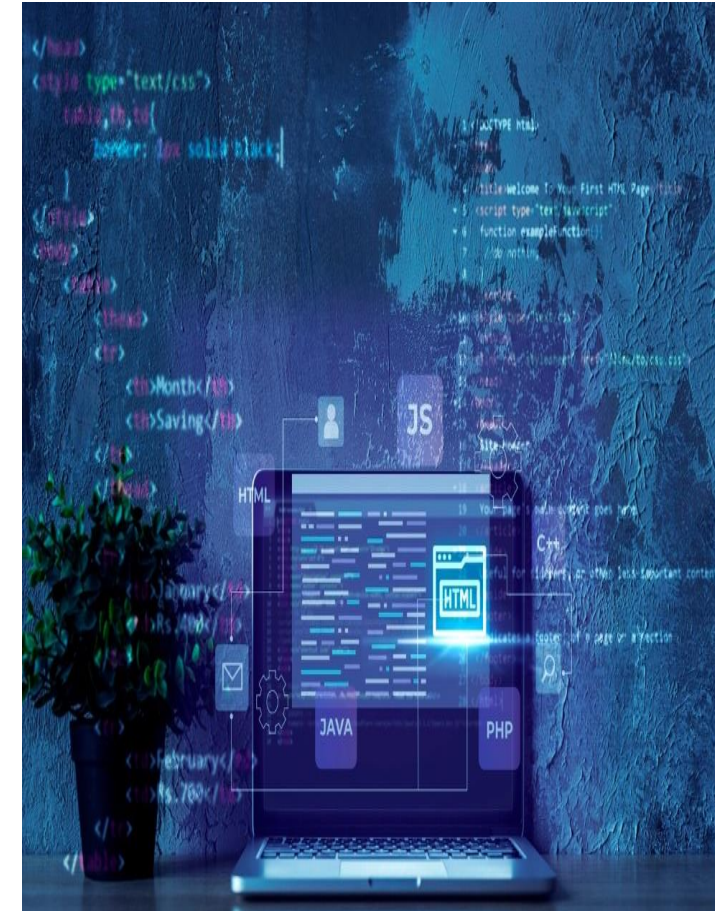
`</dl>`





# Tabelas

- As tabelas são criadas sobre a tag base `<table></table>` e é dividida em linhas `<tr></tr>` e colunas `<td></td>`.
- A ordem de hierarquia é essa: `table > tr > td`, uma dentro da outra. Ou seja, a tabela é dividida em linhas que, por sua vez, são divididas em colunas.
- A tag `table` possui o **atributo “border”**, que define a borda das células com um número inteiro representando a **espessura**.



# Exemplo de Uso de Tabela

```
<table border="1">
```

```
<tr>
```

```
<td>Linha 1, Coluna 1</td>
```

```
<td>Linha 1, Coluna 2</td>
```

```
<td>Linha 1, Coluna 3</td>
```

```
</tr>
```

```
<tr>
```

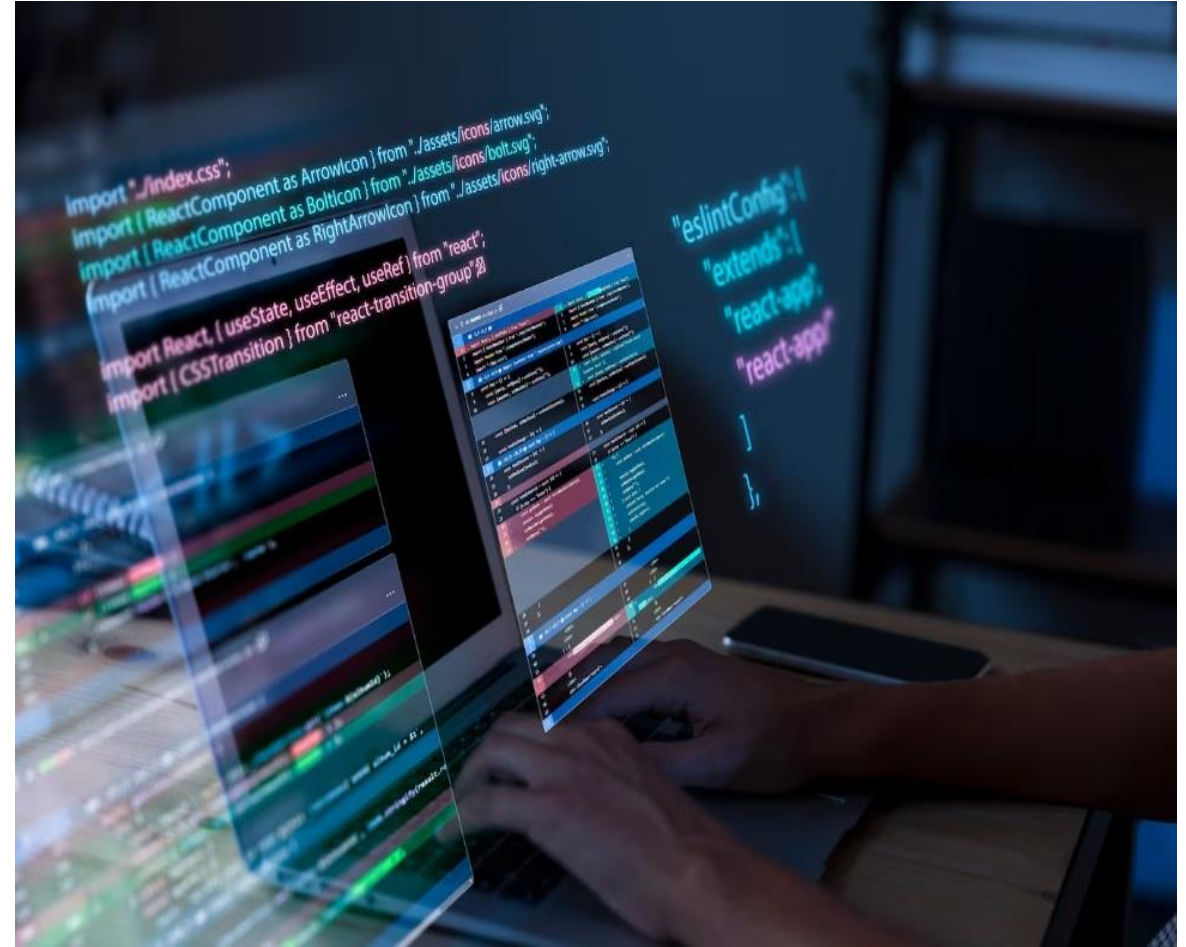
```
<td>Linha 2, Coluna 1</td>
```

```
<td>Linha 2, Coluna 2</td>
```

```
<td>Linha 2, Coluna 3</td>
```

```
</tr>
```

```
</table>
```



# Atributos ID e CLASS

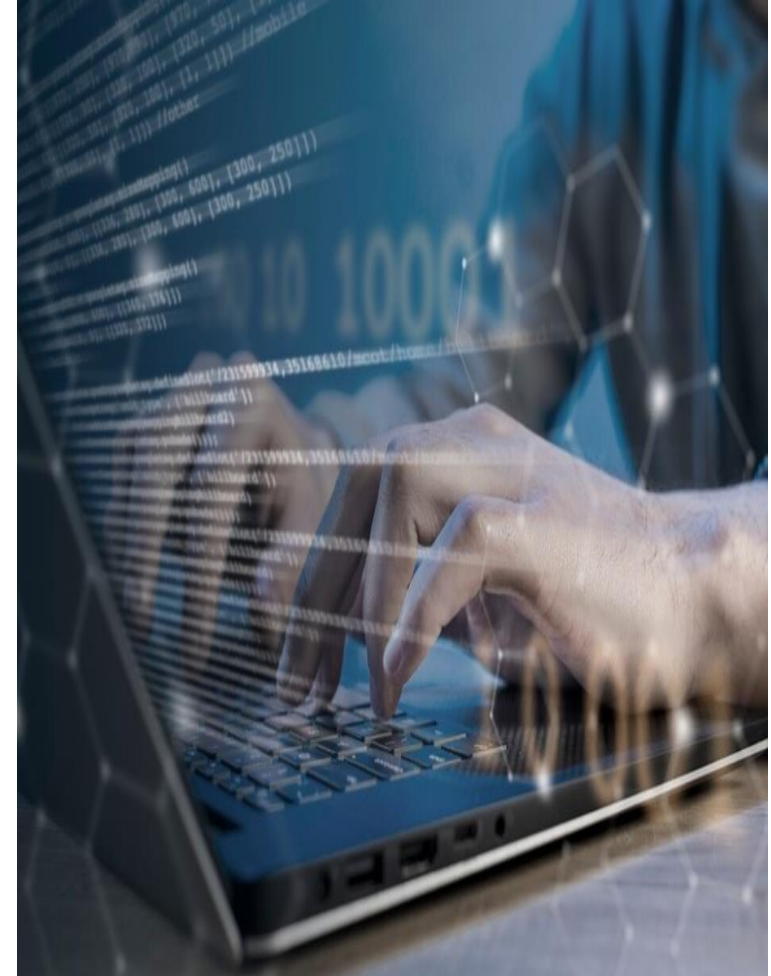
- Em resumo:
  - Como o próprio nome já diz, são formas de “selecionar” tags HTML, por seu tipo ou atributos.
  - Toda e qualquer tag pode receber os atributos ID e CLASS, muito utilizados para auxiliar no uso de CSS, Javascript e comunicação entre Back-End e Front-End.





# Atributos ID e CLASS

```
<table border="1" id="tabela1" class="modelo1">  
  <tr>  
    <td>Linha 1, Coluna 1</td>  
    <td>Linha 1, Coluna 2</td>  
    <td>Linha 1, Coluna 3</td>  
  </tr>  
</table>
```



# Atributos ID e CLASS

- É de conhecimento que existem seletores para as *tags* no HTML, tais como os atributos ID e a CLASS. No qual nos possibilita aplicar, via CSS, o mesmo estilo a diferentes *tag* distribuídas em nossa(s) página(s) HTML.





# Atributos ID

**id:** Identificador único

Deve ser usado apenas uma vez por página.

Serve para identificar um único elemento de forma exclusiva.

É muito útil para manipulação com JavaScript, ou estilos específicos.

HTML

```
<p id="destaque">Este parágrafo tem um ID único.</p>
```

Css

```
#destaque {  
  color: red;  
}
```

# Atributos Class

**class:** Grupo de elementos  
Pode ser usada em vários elementos.

Serve para aplicar o mesmo estilo ou comportamento a um grupo de elementos.

Permite reutilização e organização.

HTML

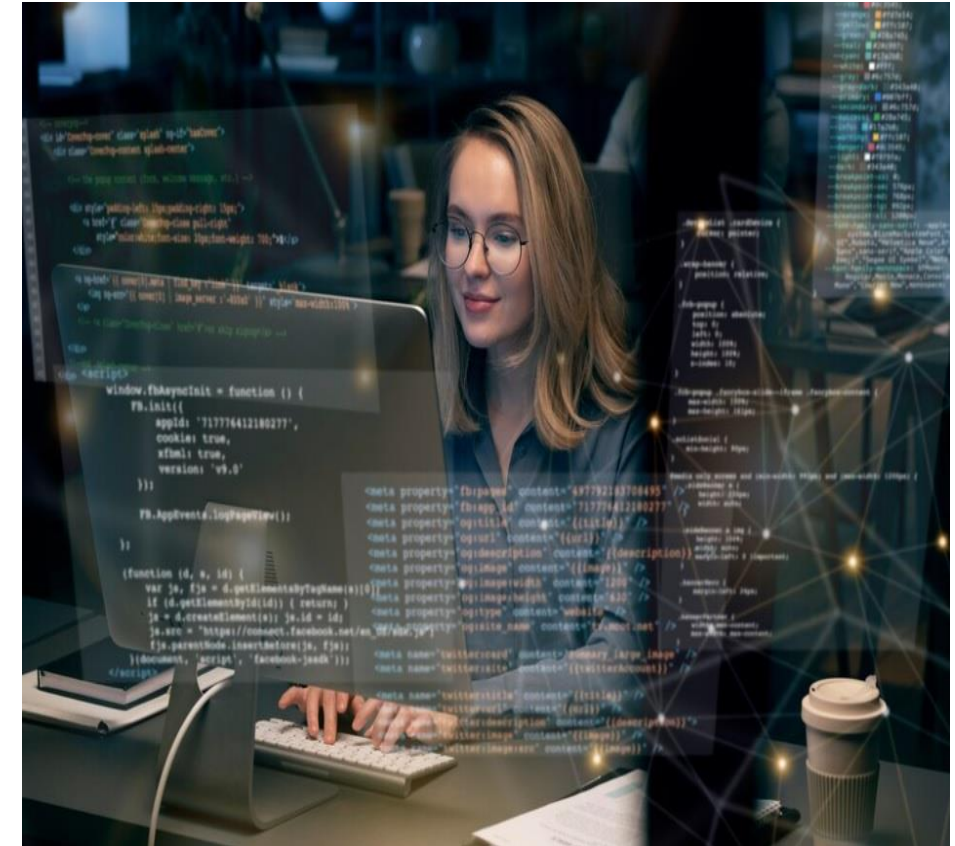
```
<p class="aviso">Aviso importante!</p>  
<p class="aviso">Outro aviso com o mesmo estilo.</p>
```

Css

```
.aviso {  
  color: orange;  
  font-weight: bold;  
}
```

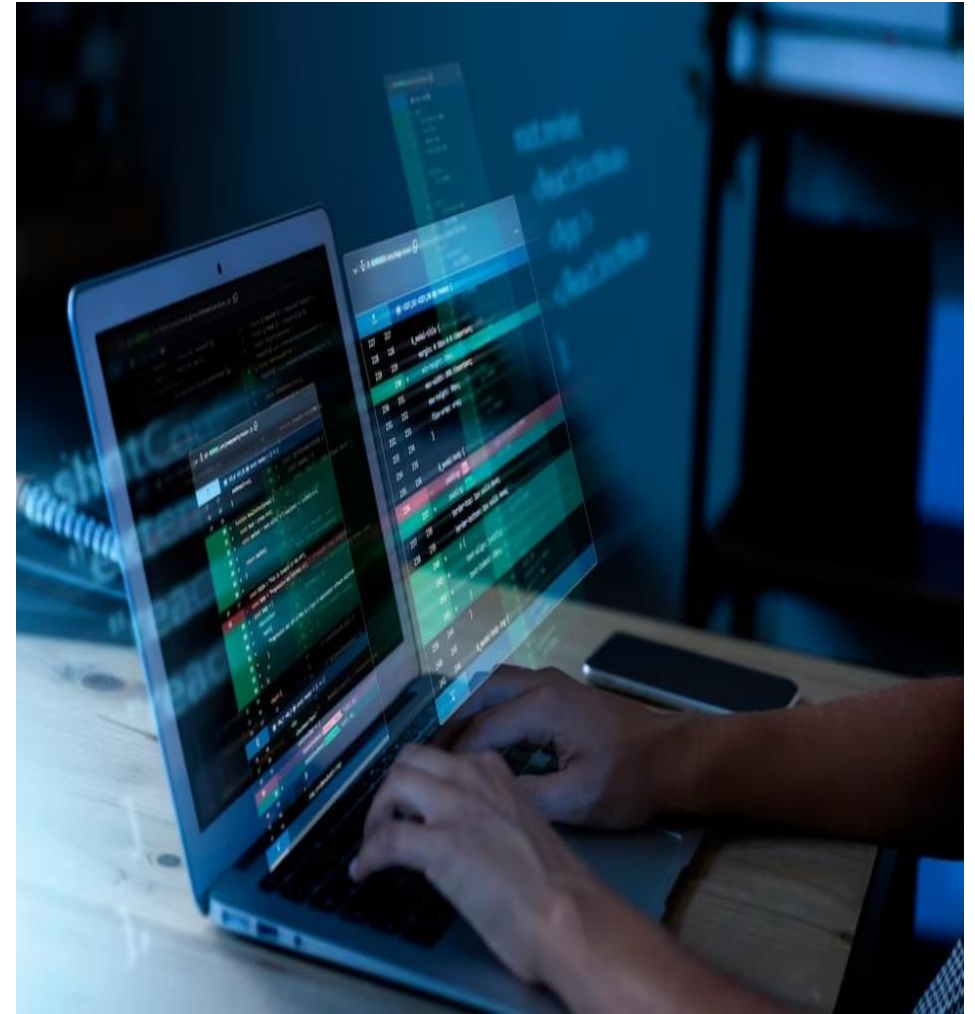
# Atributos ID e CLASS

- Isso nos possibilita grande produtividade e criatividade, mas ainda assim pode não solucionar todos os nossos problemas.
- Um menu muito grande, vamos repetir ID e CLASS em cada um dos itens?



# <div> </div>

- O nome ***div*** vem de divisão, e esta *tag* nos permite dividir qualquer trecho de seu código. Isso mesmo, você pode criar um bloco, uma divisão, e dentro deste bloco ter uma imagem, links, textos e o que mais você quiser.



## <div> </div>

- Pode-se aplicar o CSS nesse bloco (div), e tudo que estiver dentro da **div**, seja imagem, texto, link ou o que mais for, vai receber aquelas regras de estilo.
- Se você fizer isso usando CLASS, por exemplo, vai ter que colocar esse atributo (seletor) em cada *tag*.



# <div> </div>

- É bem comum dizer que a *tag* <div> é um container para armazenar diversos elementos.
- As *divs* são IMPORTANTÍSSIMAS para estruturar, criar uma ordem lógica e organizar um site.



## `<div>` `</div>`

- Tu podes criar uma div para o cabeçalho, outra para o conteúdo, outra div para os menus e uma última para o rodapé, então usa o CSS para estilizar cada uma dessas divs separadamente.
- Também poderá criar outras divisões dentro destas divs. Ou seja, criar div dentro de div.



<div> </div>

<div>

Todo, qualquer e quantos  
elementos do HTML você queira  
colocar aqui.

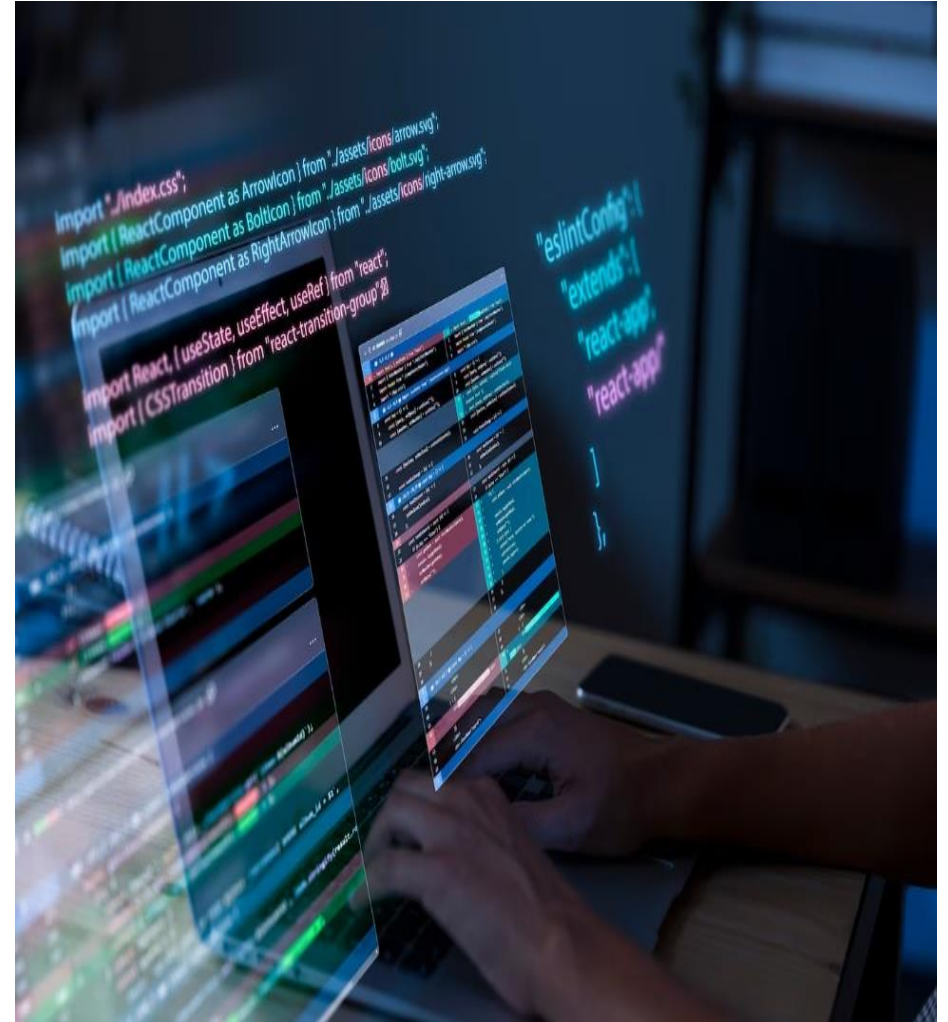
</div>





# <div> </div>

- Não iremos repetir os **atributos seletores** em cada um dos componentes internos de uma div, mas apenas na div.



**<div> </div>**

<body>

<div id="header">

</div>

<div id="content">

</div>

<div id="foot">

<p>© Copyright 2018 UNIFRA</p>

</div>

</body>





# Exemplo

```
<!DOCTYPE html>
```

```
<html>
```

```
  <body>
```

```
    <p>Parágrafo fora da div.</p>
```

```
    <div style="color:#0000FF">
```

```
      <h3>Tag H3 dentro da div</h3>
```

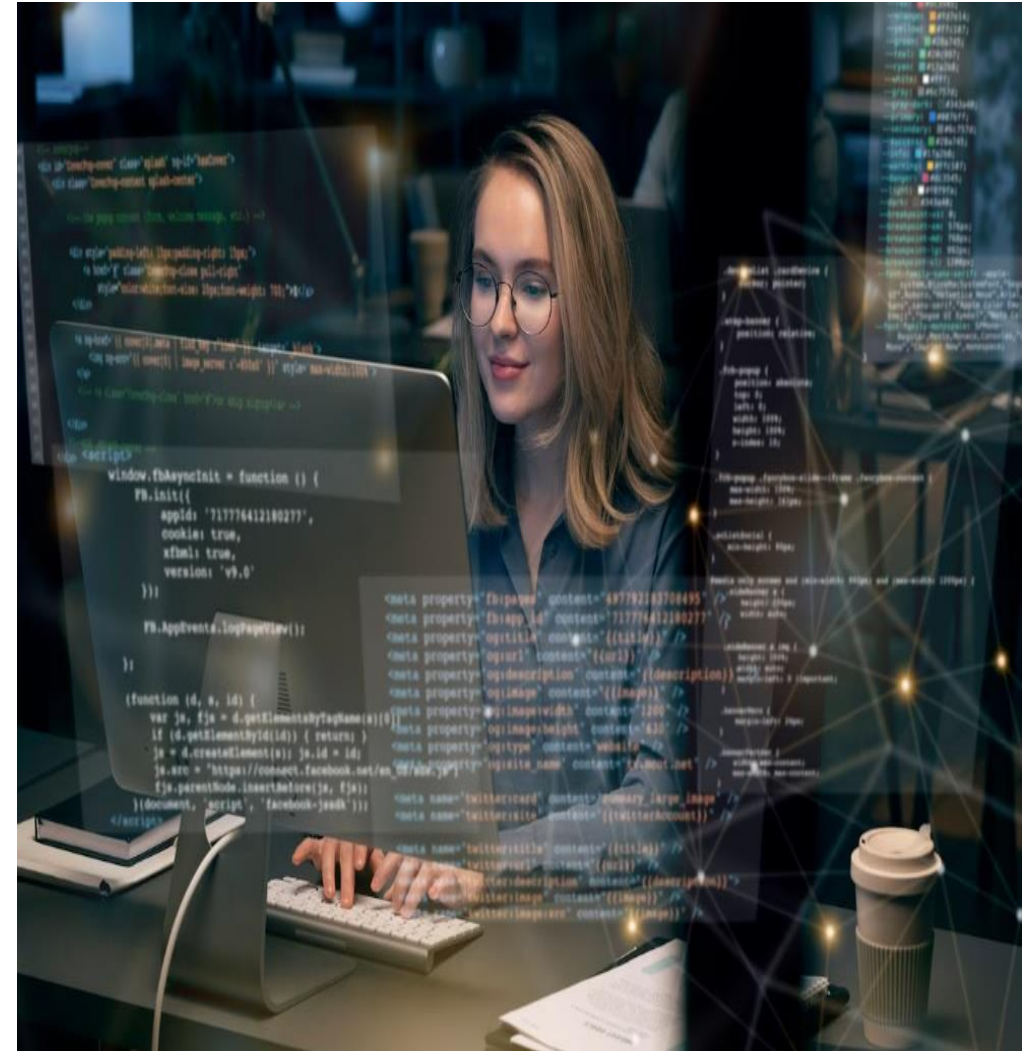
```
      <p>Parágrafo dentro da div.</p>
```

```
    </div>
```

```
    <p>Parágrafo fora da div.</p>
```

```
  </body>
```

```
</html>
```



# Referências

HOGAN, Brian P. HTML 5 e CSS3: Desenvolva hoje com o padrão de amanhã . Rio de Janeiro (RJ): Ciência Moderna Ltda., 2012 282 p. ISBN 978-85-399-0260-6

Fábio Flatschart. HTML 5 - Embarque Imediato, 2011. (Biblioteca Digital)

Deitel, Paul J.; Deitel, Harvey M.. Ajax, Rich Internet Applications e Desenvolvimento Web para Programadores, 2008. (Biblioteca Digital)

SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. 2. ed. São Paulo, SP: Novatec, 2014. 335 p. ISBN 978-85-7522-403-8.

Denilson Bonatti. Desenvolvimento de Jogos em HTML5, 2014. (Biblioteca Digital)

W3SCHOOL. The world's largest web development site. Acessado em 2018. Disponível em: <http://www.w3schools.com/>.

W3C. World Wide Web Consortium. Acessado em 2018. Disponível em: <http://www.w3.org>

MORRISON, Michael. Use a cabeça JavaScript. Rio de Janeiro (RJ): Alta Books, 2008. 606 p. <https://br.freepik.com/fotos-vetores-gratis/desenvolvimento-web>

Material do Professor Fabrício Tonetto Londero, 2023.

Thank you for your attention!!

---



Email: [andre.flores@ufn.edu.br](mailto:andre.flores@ufn.edu.br)