

Desenvolvimento WEB

PARTE 2

Prof: Gilmar Luiz de Borba

1

2025-1



UNIVERSIDADE
FUMEC

Capítulo 6

- 1 – Considerações sobre escalas na viewport.
- 2 – Unidade de medida “rem”, seletor padrão (`html { ... }`) para fontes fixas.
- 3 – Alterando o tamanho da fonte de parágrafos com “rem”.
- 4 – Alterando o tamanho da fonte de títulos com “rem”.
- 5 – Ajustando parágrafos no bloco `<footer>`.
- 6 – Âncoras: considerações.
- 7 – Emmet Abbreviation: Lorem Ipsum.

ESCALAS NA VIEWPORT

MAXIMUM-SCALE=1.0

O atributo `maximum-scale=1.0` é usado para controlar o nível máximo de zoom nos dispositivos móveis. É usado para: prevenir zoom excessivo e melhorar experiência de usuário, evitando problemas de acessibilidade.

O atributo `maximum-scale=1.0`, em alguns casos, deve ser usado com cautela para não comprometer a acessibilidade, porque ele impede que usuários, com baixa visão, ampliem o conteúdo da página além do limite do dispositivo, dificultando a leitura nesses casos.

OBSERVAÇÕES:

`initial-scale=1.0`: use para definir que a página será carregada com escala normal (sem zoom inicial).

`maximum-scale=1.0`: usado para impedir que o usuário amplie a página além desse valor.

Se for usado, por exemplo, `maximum-scale=2.0`, por exemplo, o usuário poderia ampliar a página até o dobro do tamanho original.

ESCALAS NA VIEWPORT

EVITAR DESATIVAR O ZOOM COMPLETAMENTE

Alguns desenvolvedores usam: `user-scalable=no`, esse recurso desativa o zoom completamente.

Essa implementação deve ser evitada ou usada com moderação pois pode ser prejudicial a acessibilidade, use esse recurso somente em casos específicos.

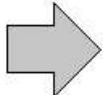
Unidades de medida mais usadas

%

Porcentagem, para auxiliar o entendimento das dimensões (altura e largura) ou área ocupada, faz a conversão responsiva.

em

É o tamanho da fonte. Em um elemento com a fonte de 1em será o tamanho da fonte padrão, 2em duas vezes o tamanho da fonte padrão, 0.5em a metade do tamanho da fonte padrão.



rem

A rem ("root em"), corresponde tamanho de fonte do elemento raiz do documento (seletor `html {...}` do estilo ou folha de estilo). A fonte rem o rem é constante em todo o documento.

pt

Ponto (1 pt é o mesmo que 1/72 de polegada). Também usado para entender visualmente a altura da linha em pontos.

px

Pixels (um ponto na tela do computador), usado para entender visualmente a altura da linha em pixels.

fr

É uma unidade usada para representar uma fração ou fragmento do espaço disponível em um *container* de um *grid*.

5

(*em*, significa algo como "element measure" ou medida do elemento, *rem* significa, relative to font-size of the root element).

Unidades de medida mais usadas

ex (“altura-x) e **ch** (character unit)

São semelhantes as unidades “em” e “rem” (baseadas nas fontes e tamanhos atuais).
São baseadas as fontes “font-family”.

A unidade **ch**, é definida como a “medida” da largura do caractere zero (“0”).
A unidade **ex** é definida a altura do caractere “x” em minúsculo dessa fonte.

vh (viewport height) e **VW** (viewport Width)

Tanto “vh” como “vw” usam a largura ou a altura da viewport.

“vh” corresponde a 1/100 da altura da viewport, se a altura do navegador é 900px, 1vh equivale a 9px.
“vw” se a largura da viewport é 750px, 1vw equivale a 7.5px.

Vh e vw, são usadas, normalmente no recurso dos carrocéis ou sliders.

Unidades de medida mais usadas

clamp()

Para fontes flexíveis use *clamp()*

```
h1 {  
    font-size: clamp(1.2rem, 5vw, 2.0rem);  
}
```

clamp() em CSS define um valor dinâmico e responsivo.

Sintaxe:

clamp(Mínimo, Default, Máximo)

Mínimo → O menor valor permitido.

Default (ou valor preferido) → O valor ideal (pode ser baseado %, vw, ou outros (de preferência relativos)).

Máximo → O maior valor permitido.

Unidades de medida mais usadas

clamp()

Para fontes flexíveis use clamp()

A unidade vw, como visto anteriormente, representa 1% da largura da viewport (área visível do navegador), é útil para criar layouts responsivos.

Vantagens do clamp()

- Evita media queries* desnecessárias.
- Mantém a responsividade e adaptação automática.
- Facilita a manutenção do código.

* Media queries

```
@media (max-width: 1440px) {  
    /* Ajustes para telas 1440px */  
}  
  
/* Para telas até 768pixels */  
@media only screen and (max-width: 768px) {  
    body {  
        background-color: #CD853F;  
        font-size: 1.4rem;  
    }  
}
```

A unidade “rem”

Diferença entre `em` e `rem`

(com base no uso de fontes absolutas em pixels)

O `em` usa para o cálculo do tamanho da fonte o elemento pai, criando uma hierarquia no cálculo, para tornar responsivo a visualização (aumentar ou diminuir a fonte).

Quando há a redução da fonte em *apps* (por exemplo) pode ocorrer problema de acessibilidade, ou seja, dificuldades de visualização de letras pequenas.

Basicamente usamos “`rem`” para deixar mais fácil o trabalho, no caso de deixar o projeto responsivo, em outras palavras, para adaptar a fonte para vários tamanhos de telas.

Usando `rem` teremos sempre o tamanho da fonte estático independente se for navegador ou um *app*. Por exemplo: No *Browser* 16 pixels, no *app* também 16 pixels.

A unidade “rem”

Acessibilidade

O Google dá preferência para os textos de sites que possuem um tamanho mínimo e máximo de fonte (que varia entre 12 e 16), mas recomenda normalmente o tamanho de fonte 16 (por questões de acessibilidade).

Logo a medida de 16 pixels se torna, normalmente, um padrão.

PASSO 1:

Após o seletor universal, criar o seletor HTML (que é o root). Qual é a raiz de um documento HTML? É a tag <html>!

```
html {  
  font-size: 62.5%;      Por que 62.5% ?  
}
```

A unidade “rem”

O font-size, dos parágrafos de nossa página, são por padrão 16 pixels.

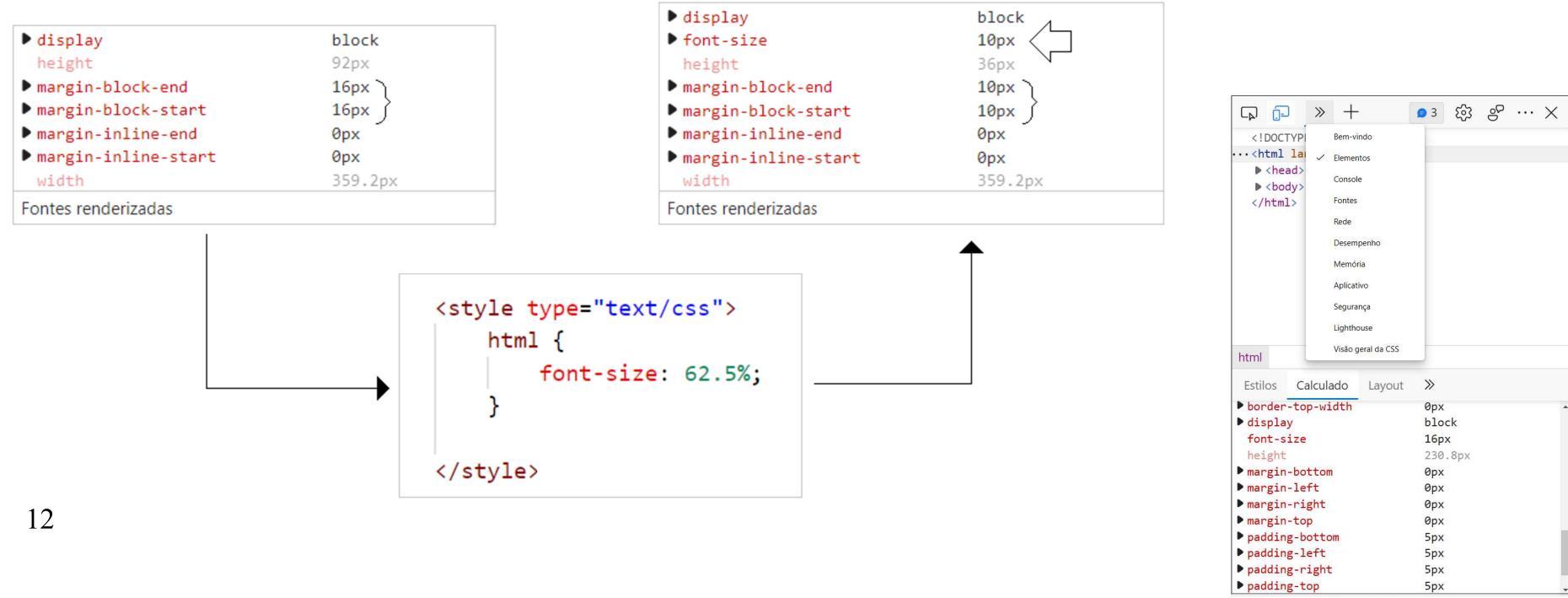
The screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. On the left, the DOM tree displays the structure of the HTML document, including an

, two s, several s, and a `Basicamente usamos "rem" para deixar mais fácil o trabalho, no caso de deixar o projeto responsivo, em outras palavras, para adaptar a fonte para vários tamanhos de telas. Usando rem teremos sempre o tamanho da fonte estático independente se for navegador ou um app. Por exemplo: No Browser 16 pixels, no app também 16 pixels."` comment. On the right, the 'Calculated' tab of the styles panel is active, showing the computed styles for the selected element. The `font-size` property is highlighted with a red border and has a value of `16px`. A large gray arrow points from the text above to this highlighted entry. Other properties listed include `color` (`rgb(0, 0, 0)`), `display` (`block`), `font-family` (`arial`), `height` (`147.2px`), `margin-block-end` (`16px`), `margin-block-start` (`16px`), `margin-inline-end` (`0px`), `margin-inline-start` (`0px`), and `width` (`359.2px`). Below the styles panel, the 'Fonts rendered' section shows 'Arial — Arquivo local (319 glyphs)'. ``` <!DOCTYPE html> <html lang="en"> <head>...</head> <body> <h1>...</h1> <h2>...</h2> <p>...</p> <h2>...</h2> <p> Basicamente usamos "rem" para deixar mais fácil o trabalho, no caso de deixar o projeto responsivo, em outras palavras, para adaptar a fonte para vários tamanhos de telas. Usando rem teremos sempre o tamanho da fonte estático independente se for navegador ou um app. Por exemplo: No Browser 16 pixels, no app também 16 pixels." </p> </body> </html> ```

A unidade “rem”

Ao criar um bloco de parágrafo HTML o padrão *start* e *end* é 16px.

Ao definir no root HTML a unidade relativa 62,5%, é criado o fonte-size para 10px.



A unidade “rem”

Por que 62.5% ?

Para facilitar a base de cálculo: $0,625 * 16 = 10$ (pixels)

O font-size de nossa página agora é por padrão 10 pixels.

Verificando o novo tamanho de fonte no Google Chrome:

The screenshot shows the Google Chrome DevTools interface with the 'Elements' tab selected. On the left, the DOM tree displays a `<p>` element containing a block of text about using 'rem' units for font sizes. On the right, the 'Calculated' tab of the styles panel is active, showing the computed styles for this element. The `font-size` property is highlighted with a red box and a large blue arrow pointing to it from the text above. Other properties listed include `color`, `display`, `font-family`, `height`, `margin-block-end`, `margin-block-start`, `margin-inline-end`, `margin-inline-start`, and `width`. The value for `font-size` is shown as `10px`.

Novo tamanho =

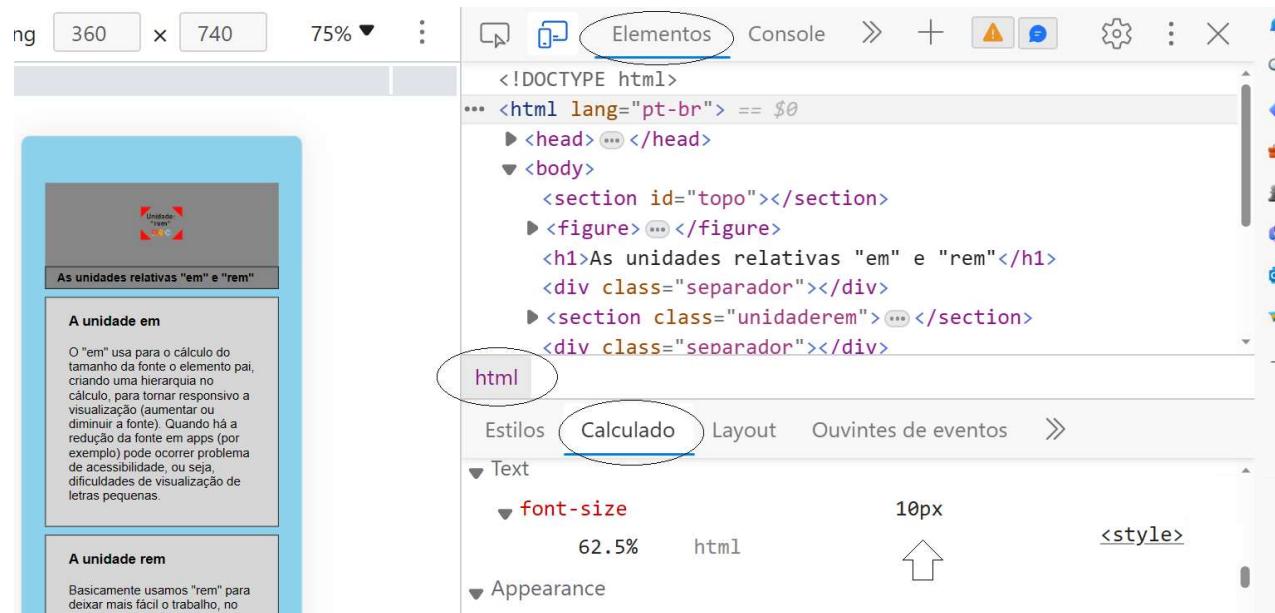
$$16 \text{ px} * 62,5\% = 10 \text{ px}$$

A unidade “rem”

Ao criar um bloco de parágrafo HTML o padrão é 16px.

Ao definir no root HTML a unidade relativa 62,5%, é criado o fonte-size para 10px.

Verificando o novo tamanho de fonte no Microsoft Edge:



Novo tamanho =

$$16 \text{ px} * 62,5\% = 10 \text{ px}$$

A unidade “rem”

PASSO 2:

Para configurar, por exemplo, o font-size do parágrafo para 16 pixels:

```
p {  
    font-family: arial;  
    color: #808080;  
    text-align: justify;  
    font-size: 1.6rem;    ← (10 * 1.6 = 16 pixels)  
}
```

Esse mesmo recurso pode ser usado para formatar outros elementos da página ...

A unidade “rem”

Para configurar, por exemplo, os títulos:

```
h1 {  
    font-size: 2.2rem; → 2.2 * 10 = 22 pixels  
}
```

```
h2, h3 {  
    font-size: 1.8rem; → 1.8 * 10 = 18 pixels  
}
```

```
h4 {  
    font-size: 1.6rem; → 1.6 * 10 = 16 pixels  
}
```

Ajustando o footer (rodapé)

Seja o rodapé a seguir:

```
<footer>
  <h3>HTML5</h3>
  <h4>Disciplina WEB, CSS e JS</h4>
  <p>contato: prof@prof.web.html.com.br</p>
  <p>whatsapp: 6070-6070</p>
</footer>
```

Ao alterar os parágrafos da página, os blocos de parágrafo do *footer* serão também afetados (veja a imagem a seguir).

Como solucionar essa questão?

E a formatação para os parágrafos:

```
p {
  font-family: arial;
  color: #808080;
  text-align: justify;
  font-size: 1.6rem;
}
```

Ajustando o footer (rodapé)

A página ficará assim:

The screenshot shows a Microsoft Word document window. The title 'As unidades relativas em e rem' is at the top center. Below it, 'A unidade em' is centered. A paragraph follows, explaining the use of 'em' for font size relative to the parent element. Then, 'A unidade rem' is centered, with a paragraph explaining its use for responsive design. The 'HTML5' section is centered, followed by 'Disciplina WEB,CSS e JS'. At the bottom left, contact information is listed: 'contato: prof@prof.web.html.com.br' and 'whatsapp: 6070-6070'. On the right side, there are several icons for styling and document management. To the right of the text blocks, there are labels indicating the CSS properties used: 'H1 (text-align: center)', 'H2 (text-align: center)', 'H2 (text-align: center)', 'H3 (text-align: center)', and 'H4 (text-align: center)'. A large red arrow points from the text 'Parágrafos "p" (text-align: justify)' to the 'justify' icon in the ribbon.

As unidades relativas *em* e *rem*

A unidade *em*

O *em* usa para o cálculo do tamanho da fonte o elemento pai, criando uma hierarquia no cálculo, para tornar responsivo a visualização (aumentar ou diminuir a fonte). Quando há a redução da fonte em apps (por exemplo) pode ocorrer problema de acessibilidade, ou seja, dificuldades de visualização de letras pequenas.

A unidade *rem*

Basicamente usamos "*rem*" para deixar mais fácil o trabalho, no caso de deixar o projeto responsivo, em outras palavras, para adaptar a fonte para vários tamanhos de telas. Usando *rem* teremos sempre o tamanho da fonte estático independente se for navegador ou um app. Por exemplo: No Browser 16 pixels, no app também 16 pixels.

HTML5

Disciplina WEB,CSS e JS

contato: prof@prof.web.html.com.br
whatsapp: 6070-6070

Parágrafos "p" (text-align: justify)

H1 (text-align: center)
H2 (text-align: center)
H2 (text-align: center)
H3 (text-align: center)
H4 (text-align: center)

Queremos todo o *footer* centralizado, como resolver essa questão?

Ajustando o *footer* (rodapé)

Definindo uma classe para os parágrafos do rodapé:

```
<footer>
    <h3>HTML5</h3>
    <h4>Disciplina WEB,CSS e JS</h4>
    <p class="rodape">contato: prof@prof.web.html.com.br</p>
    <p class="rodape">whatsapp: 6070-6070</p>
<footer>
```

Fazendo a configuração da classe:

```
.rodape {
    text-align: center;
}
```

Ajustando o *footer* (rodapé)

Definindo uma classe para os parágrafos do rodapé:

The screenshot shows a web browser window with the following content:

Arquivo | D:/ProjetosHTML/trabalhandocomrem/trabalhandocomrem.html

As unidades relativas *em* e *rem*

A unidade *em*

O *em* usa para o cálculo do tamanho da fonte o elemento pai, criando uma hierarquia no cálculo, para tornar responsivo a visualização (aumentar ou diminuir a fonte). Quando há a redução da fonte em apps (por exemplo) pode ocorrer problema de acessibilidade, ou seja, dificuldades de visualização de letras pequenas.

A unidade *rem*

Basicamente usamos "rem" para deixar mais fácil o trabalho, no caso de deixar o projeto responsivo, em outras palavras, para adaptar a fonte para vários tamanhos de telas. Usando *rem* teremos sempre o tamanho da fonte estático independente se for navegador ou um app. Por exemplo: No Browser 16 pixels, no app também 16 pixels.

HTML5
Disciplina WEB,CSS e JS
contato: prof@prof.web.html.com.br
whatsapp: 6070-6070

Bordas

As bordas separam o *padding* (preenchimento) da margem. São usadas para marcar, delimitar e separar um bloco HTML. Para criar uma borda use a propriedade [border-style](#).

```
* {  
    margin: 0px;  
    border: 1px; ←  
    padding: 10px;  
}
```

1

```
h1 {  
    border-style: double; ←  
    padding-left: 10px;  
    ...  
}
```

2

```
.unidaderem {  
    border-style: double; ←  
    font-size: 1.6rem;  
    ...  
}
```

3

...

border

- 🔧 [border](#)
- 🔧 [border-radius](#)
- 🔧 [border-bottom](#)
- 🔧 [border-width](#)
- 🔧 [border-right](#)
- 🔧 [border-style](#)
- 🔧 [border-top-left-radius](#)
- 🔧 [border-top-right-radius](#)
- 🔧 [border-bottom-left-radius](#)



Bordas

Criando bordas com “border”

```
* {  
    margin: 0px;  
    border: 0px; ← 1  
    padding: 10px;  
}  
  
h1 {  
    border: 1px solid #000; ← 2  
    padding-left: 10px;  
    ...  
}
```

```
.unidaderem {  
    border: 1px solid #000; ← 3  
    font-size: 1.6rem;  
    ...  
}
```

22

A unidade rem ————— subtítulo

Basicamente usamos "rem" para deixar mais fácil o trabalho, no caso de deixar o projeto responsivo, em outras palavras, para adaptar a fonte para vários tamanhos de telas. Usando "rem" teremos sempre o tamanho da fonte responsiva independente se for navegador ou um app. Por exemplo: No Browser 16 pixels, no app também 16 pixels.

borda
background-color

parágrafo

Âncoras

É usado para facilitar a UX (Experiência do Usuário), possilitando a navegação na própria página. É normalmente usada em páginas longas com várias referências internas.

Para criar uma âncora basta usar uma tag como `<p>`, `<div>`, `<section>` criando um id:

(1) Criar o "id"

```
<section id="topo"></section>
```

(2) Em outra posição da página, criar a referência para localizar esse id:

```
<a class="voltar" href="#topo" rel="nofollow">Voltar</a>
```

Âncoras

"rel"

O atributo "rel" (relationship), relacionamento em português, é importante usar a constante ou parâmetro "*nofollow*" (não siga) para informar o motor de busca (do Google, por exemplo), para não dar importância ao link sem ranqueá-lo.

Em outras palavras, usar âncora sem usar o "nofollow" pode configurar o uso de links pagos (paid links) que é um tipo de blackhat.

Alguns exemplos de Blackhat:

- . Uso de links pagos (paid links),
- . Texto oculto em um site (Hidden text),
- . Camuflagem (cloaking),
- . Conteúdo duplicado . . .

LOREM IPSUM

No VS Code, no bloco `<body>` quando digitamos `lorem10`, `lorem20` ... ao pressionar `Tab` ou `Enter`, o editor gera automaticamente um texto de preenchimento com a quantidade de palavras correspondente ao número indicado. Esse recurso é usado para gerar um texto com N palavras aleatórias em Latim (conhecido como "Lorem Ipsum"), usado para simular conteúdos.

```
<body>
  Lorem30
    ↴ Lorem30
    Abreviação Emmet
</body>
```



```
<body>
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Animi voluptatibus
  odio expedita consectetur ex porro ut esse, in voluptatum similique,
  accusantium ipsam, corporis sunt recusandae quibusdam hic quam
  sapiente totam?
</body>
```

Esse recurso ajuda no desenvolvimento rápido de protótipos e testes de layout sem precisar copiar longos textos manualmente. É possível usar o recurso já no contexto de um parágrafo `<p>`.

Projeto 6

Unidade de medida “rem”, seletor padrão html { ...}, alterando o tamanho da fonte de parágrafos e títulos com “rem”, ajustando parágrafos no bloco <footer>

Capítulo 7

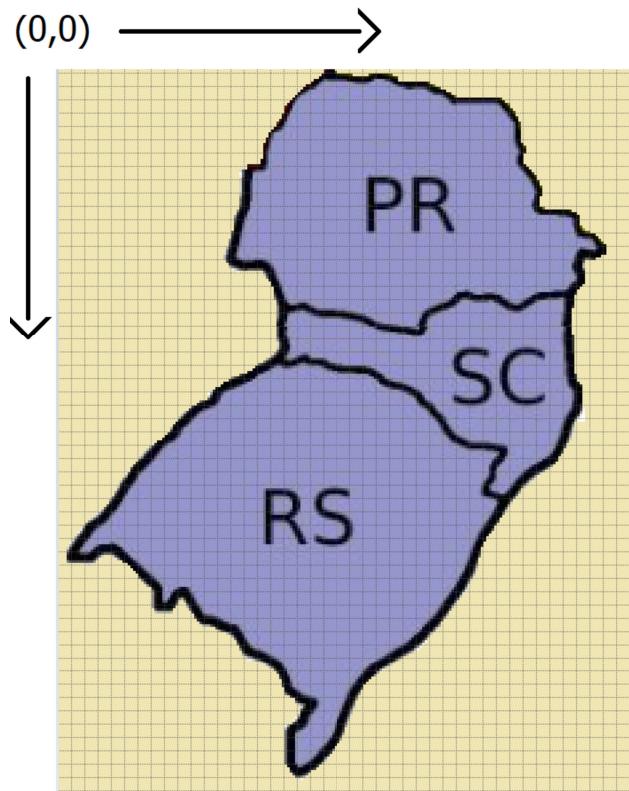
- 1 – Trabalhando com mapas
- 2 –
- 3 – Tag <area>
- 4 – Usando shapes: circle, rect e poly
- 5 – Trabalhando com Google Maps

Trabalhando com mapas - definições

- Como definido em “www.w3schools.com”, trata-se de um mapa de imagem, com áreas clicáveis.
- Recurso a partir do HTML5.
- Usado para criar um mapa (ou imagem) clicável a partir de um *shape* (forma, que pode ser círculo, retângulo ou polígono).
- O atributo (obrigatório) “*name*” é usado para fazer a associação com a imagem
- A tag "*map*" corresponde ao espaço das áreas clicáveis.

Capítulo 7 - Trabalhando com mapas

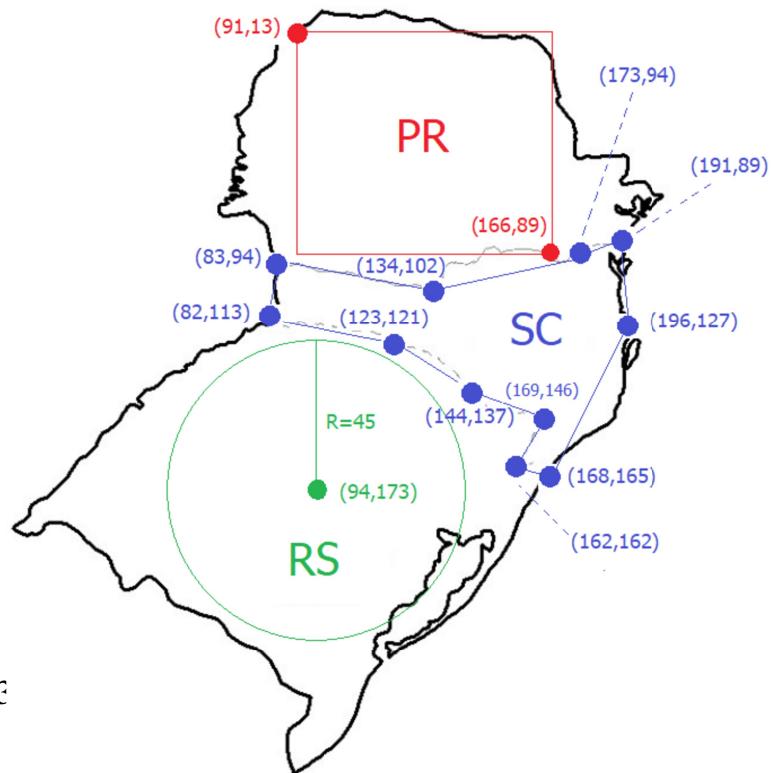
Passo 1 – Escolher a imagem base (um mapa, uma planta, um esboço etc.)



Trabalharemos com as coordenadas das regiões de interesse, na imagem.

Capítulo 7 - Trabalhando com mapas

Passo 2 – Mapear a imagem nos locais clicáveis



Há 3 tipos principais de *shapes*:

Rect (retângulo)

Circle (círculo)

Poly (Polígono)

Capítulo 7 - Trabalhando com mapas

Passo 3 – Informar qual imagem será mapeada

- (1) Carregar a imagem no HTML,
- (2) nomear a imagem (como mapa),
- (3) associar (usar) o mapa à imagem carregada.

```
<section class="mapa">
  <figure>
    |①➡
    </figure>
    <map name="mapasul">
  </section> ②
```



IMPORTANTE:

O uso da “#” é obrigatório
em **usemap**

Capítulo 7 - Trabalhando com mapas

Passo 4 – Criar as áreas clicáveis

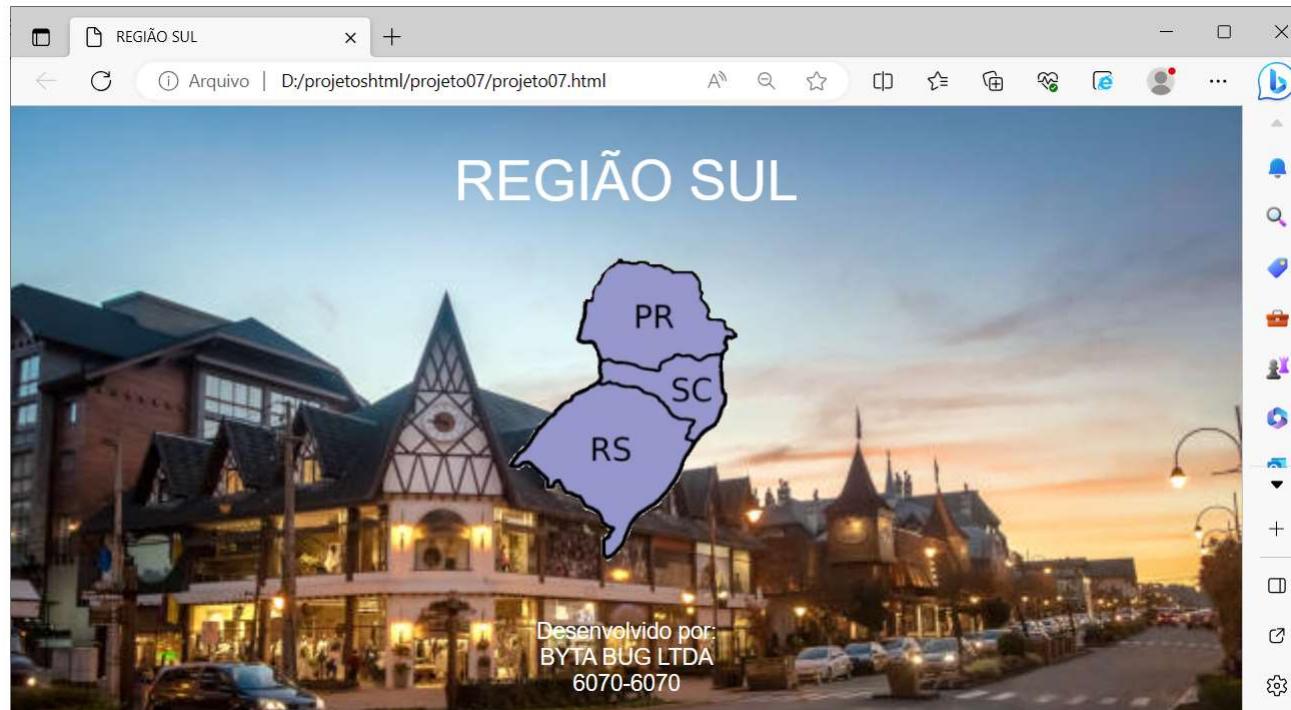
```
<map name="mapasul">
    <!-- USANDO RETÂNGULO --&gt;           Tipo de forma (shape) usado (circle, rect, poly)
    &lt;area alt="pr" shape="rect" href="https://pt.wikipedia.org/wiki/Paran%C3%A1"
          coords="91,13,166,89" title="Informações do Paraná" target="_blank"&gt;
        Coordenadas da área clicável

    <!-- USANDO CÍRCULO --&gt;
    &lt;area alt="rs" shape="circle" href="https://pt.wikipedia.org/wiki/Rio_Grande_do_Sul"
          coords="94,173,45" title="Informações do Rio Grande do Sul" target="_blank"&gt;

    <!-- USANDO POLÍGONO --&gt;
    &lt;area alt="rs" shape="poly" href="https://pt.wikipedia.org/wiki/Santa_Catarina"
          coords="82,113,83,94,134,102,154,88,173,94,191,89,196,127,168,165,162,162,169,146,
          144,137,123,121" title="Informações de Santa Catarina" target="_blank"&gt;</pre>
```

Capítulo 7 - Trabalhando com mapas

A página em execução ...



Capítulo 7 - Trabalhando com mapas

A página em execução, detalhe ...



Capítulo 7 – Google Maps

- O *Google Maps* recebe diariamente milhares de usuários.
- O *Google Maps* é interativo, dessa forma se adapta às páginas do site.
- Há muitas maneiras de adicionar o Google Maps em um site ou página HTML.
- O *Google Maps* auxilia visitantes do site encontrar um caminho para o negócio.
- Usar o *Google Maps* no site podem ajudar a reduzir a taxa de bounce* do site.
- O *Google Maps* no site podem ajuda a UX do site.
- A incorporação básica do *Google Maps* na página é gratuita.
- Recursos avançados (street view e outros) podem ser cobrados.
- O *Google Maps* não é responsive (programação adicional ...)

35

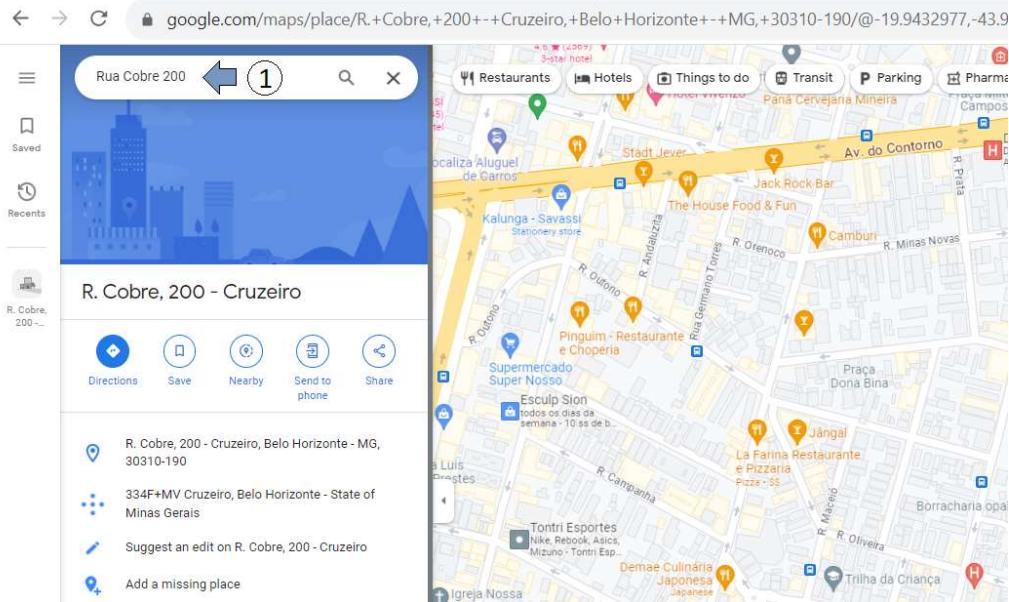
* Taxa de rejeição (marketing na Internet), é a porcentagem de visitantes que entram no site e depois saem, em vez de continuar visualizando outras páginas no mesmo site.

Capítulo 7 – Google Maps

Siga os passos

1 - Acessar o Google Maps.

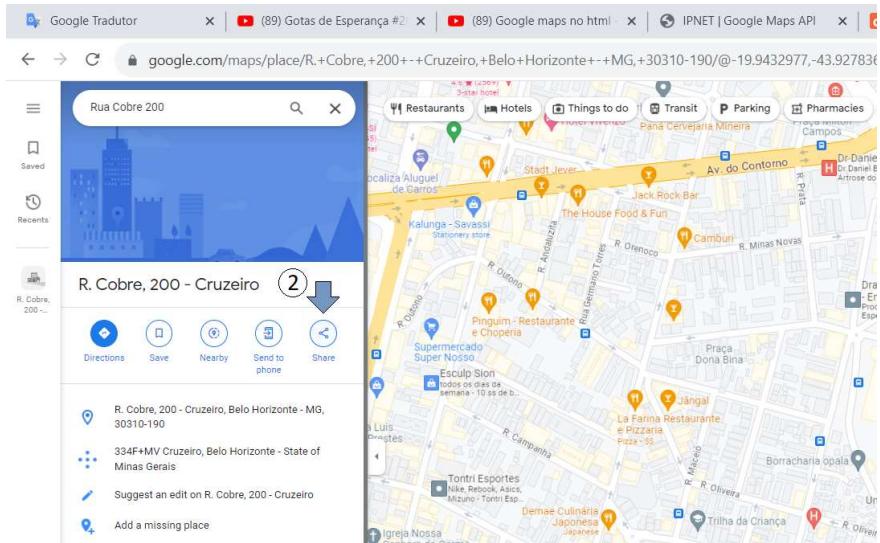
2 - Digitar o endereço na barra de pesquisa, confirmar ...



Capítulo 7 – Google Maps

Siga os passos

3 - Clique no ícone Compartilhar



Capítulo 7 – Google Maps

Siga os passos

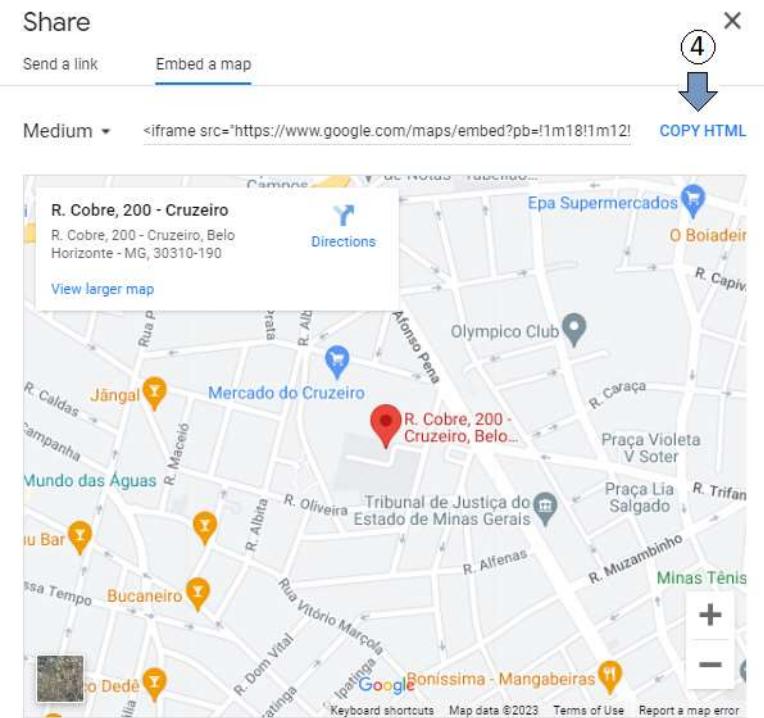
4 - Escolher "Incorporar mapa" ou “embed”.



Capítulo 7 – Google Maps

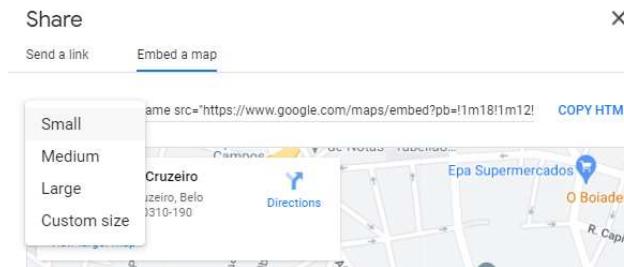
Siga os passos

5 - Copiar o código HTML de incorporação HTML



Antes de copiar, é possível
Escolher a dimensão da área.

Small, Medium ...



Custom = personalizado

Capítulo 7 – Google Maps

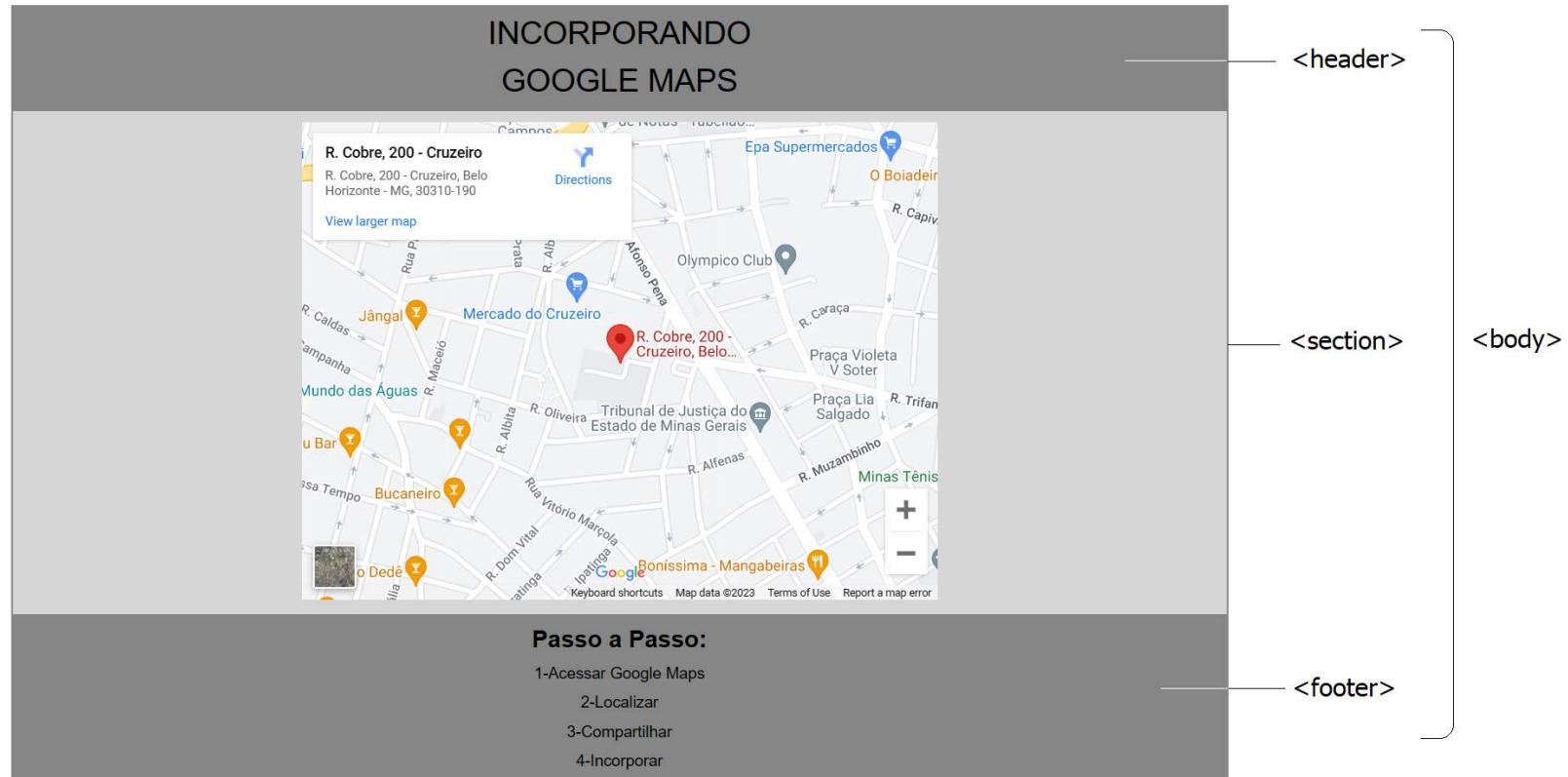
Exemplo de código HTML copiado (nesse caso envolto em uma seção)

```
<section class="maps">
  <iframe
    src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3750.551461646749!
    width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy"
    referrerpolicy="no-referrer-when-downgrade"></iframe>
</section>
```

Os valores das propriedades `width` e `height` podem ser alterados de modo a Aumentar ou diminuir a área de apresentação.

Capítulo 7 – Google Maps

Exemplo de código HTML copiado (nesse caso envolto em uma seção)



Projeto 7

Trabalhando com mapas, áreas, coordenadas e shapes.

Capítulo 8

- 1 – Fixando o *header* durante o *scroll* da página
- 2 – Aplicando transparência (parâmetro alfa do *RGBA*)
- 3 – Criando um botão com CSS
- 4 – Usando caracteres de escape nas mensagens alert (JavaScript)
- 5 – Configurando estilos em botões JavaScript
- 6 - Text-shadow

Fixando o *header* durante o scroll da página

Para fixar um elemento em uma posição da página é usada a propriedade *position fixed*. Ao fazer o scroll da página, esse elemento continua fixo na posição definida. Esse recurso é usado normalmente para fixar cabeçalhos da página.

```
header {  
    position: fixed;  
    ...  
}
```

Eventualmente algumas alterações devem ser feitas em alguns elementos da página, como margens e posição inicial d título.



Aplicando transparência no *header* durante efeito *scroll*

Para definir transparência para um elemento da página usa-se o parâmetro “alfa” propriedade *rgba* de um elemento.

```
header {  
    background-color: #rgba(128, 128, 128, 1);  
    ...  
}
```



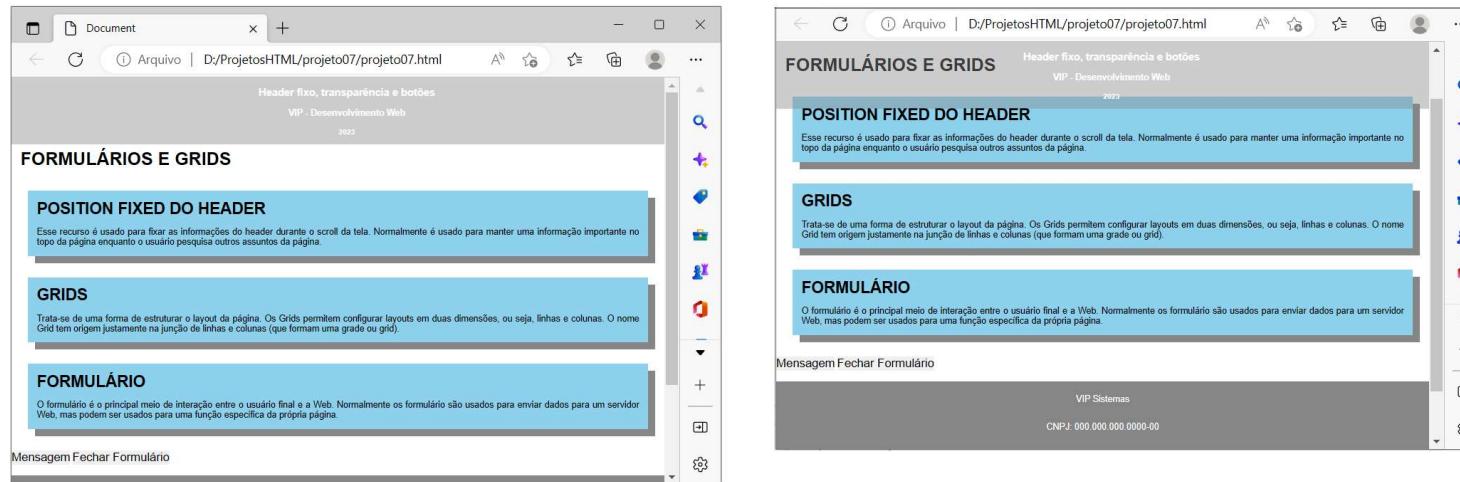
```
header {  
    background-color: #rgba(128, 128, 128, 0.42);  
}
```



Translucidez ou transparéncia de 42%

Aplicando transparência no *header* durante efeito scroll

Para definir transparência para um elemento da página usa-se o parâmetro “alfa” propriedade *rgba* de um elemento.



Criando um botão JavaScript na página

Buttom

Representa um botão que pode ser acionado para executar uma ação específica.

Parâmetros

Title: uma mensagem de ajuda "tip" ou "hint"

Name: nome de identificação do botão no código

Value: retorne o valor do atributo "value" de um botão

```
var x = document.getElementById("myBtn").value;
```

Type: tipo do elemento

Onclick: evento que ocorrerá quando o botão for "clicado"

Botão sem formatação:

[Mensagem] [Fechar]

Criando um botão JavaScript na página

Buttom

alert("Mensagem"): emite uma mensagem para o usuário
close(): Fecha a aba do navegador

```
<button title="Se você clicar aqui a página será fechada ..." name="btn_close"
    value="OK" type="button" onclick="fechar()">Fechar Formulário</button>

<script>
    function fechar() {
        alert("O formulário será fechado");
        close();
    }
</script>
```

d:\profgilmarborba\desenvweb\imagens\



Criando um botão JavaScript na página

Buttom

Alguns caracteres de *escape*:

\“ → insere o *caracter* “ dentro de uma *string* (cadeia de caracteres)

\n → insere uma quebra de linha em uma *string*.

```
<button title="Ver mensagem da página ..." name="btn_msg" value="OK" type="button"
    onclick="mensagem()">Mensagem</button>

<script>
    function mensagem() {
        alert("MENSAGEM DO SISTEMA\n\n" +
            "Olá! Você clicou no botão \"Mensagem\". Essa página trata o uso de \"header\" " +
            "transparente e também botões de comando.");
    }
</script>
```

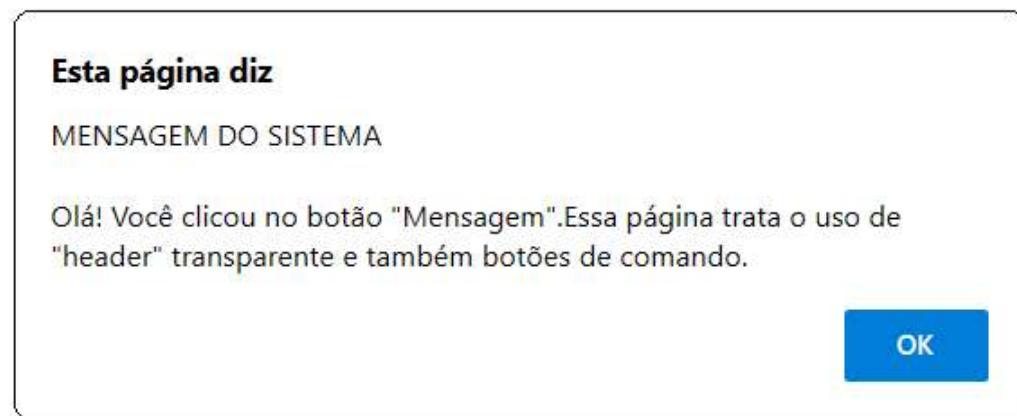
Criando um botão JavaScript na página

Buttom

Alguns caracteres de *escape*:

\“ → insere o *caracter* “ dentro de uma *string* (cadeia de caracteres)

\n → insere uma quebra de linha em uma *string*.



Criando um botão JavaScript na página

Buttom - Formatação

Inserir um id no botão:

```
<button id="botoes" class="botao" title="Mostrar mensagem da página" ...
```

Aplicar o estilo:

```
#botoes {  
    border: 3px;  
    border-style: groove;  
    background-color: #808080;  
    color: #fff;  
    text-align: center;  
    padding: 5px;  
    font-weight: bold;  
}
```

Detalhe do botão "fechar"
formatado:



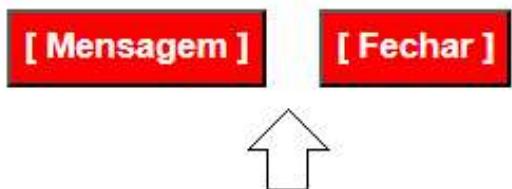
Criando um botão JavaScript na página

Buttom – Inserindo espaço(s) entre um botão e outro:

para inserir espaços entre botões é usado o recurso:

```
...
<button id="botoes" class="botao" title="Mostrar mensagem da página" name="btn_msg" value="OK" type="button"
|   onclick="mensagem()">[ Mensagem ]</button>&nbsp&nbsp&nbsp&nbsp
<!-- Foram inseridos 4 espaços (&nbsp = no break spaces )-->
<!-- entre esse botão e o próximo -->
...
...
```

Resultado:



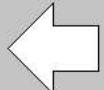
Criando um botão JavaScript na página

Text-shadow

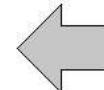
A propriedade CSS text-shadow inclui sombras ao texto. Essa propriedade aceita 3 parâmetros básicos: o deslocamento X, o deslocamento Y e a cor.

```
h1,  
h2 {  
    font-family: arial;  
    text-align: left;  
  
    padding-top: 15px;  
    padding-bottom: 15px;  
    padding-left: 20px;  
  
    text-shadow: 2px 2px □ #D3D3D3;  
}
```

FORMULÁRIOS, GRIDS



POSITION FIXED DO HEADER



Projeto 8

Fixando o *header* durante o *scroll* da página, aplicando transparência, criando um botão com CSS, caracteres de escape, text-shadow

Capítulo 9

- 1 – Trabalhando com seções, *divs* e *grids*
- 2 – a unidade “fr” (*fragment*)
- 3 – Seções com identificadores
- 4 – A propriedade *grid-template-columns* e suas variações
- 5 – Grids responsivos
- 6 - Display grid; function "repeat()" de *grid-template-columns*; gap (antigo *grid-gap*).

Trabalhando com DIV

O bloco `<div>...</div>` é usado para especificar um elemento de divisão, trata-se de um *container* genérico dentro do site. De certa forma não representa nada visual, é usado para agrupar itens do documento com a finalidade de estilos, usando para isso: *class* ou *ids*.

Lembrar que as seções `<section> ... </section>` são usadas para representar uma parte de um documento HTML, geralmente com um título (`<h1>`, `<h2>`) e contém conteúdos semânticos ou seja, agrupam partes do documento de forma lógica.

```
<section class="secao" id="grid-secao">  
    <div class="grid">  
        <div>[ Produto 1 ]</div>  
        <div>[ Produto 2 ]</div>  
        <div>[ Produto 3 ]</div>  
        <div>[ Produto 4 ]</div>  
        <div>[ Produto 5 ]</div>  
        <div>[ Produto 6 ]</div>  
    </div>  
</section>
```

56

Observação:

No código ao lado:

- 1 – Foi criado uma seção com nome de classe “`secao`”
- 2 – A seção recebeu um “`id`” de nome “`grid-secao`”. Pois nesse caso, há outras seções com o mesmo nome.
- 3 – Foi criado uma div mais externa de nome “`grid`”, para permitir a configuração geral.
- 4 – Foram criadas 6 “`divs`” que podem ser configuradas isoladamente.

Trabalhando com Grids

O que são Grids?

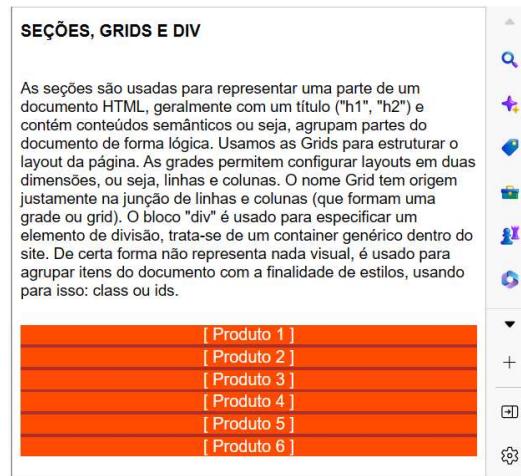
Grid é um conceito dentro do contexto do CSS. As grades (grids) auxiliam no design (desenho) de layouts de uma página Web.

O nome Grid tem origem justamente na junção de linhas e colunas (que formam uma grade ou grid).

SEÇÕES, GRIDS E DIV

As seções são usadas para representar uma parte de um documento HTML, geralmente com um título ("h1", "h2") e contém conteúdos semânticos ou seja, agrupam partes do documento de forma lógica. Usamos as Grids para estruturar o layout da página. As grades permitem configurar layouts em duas dimensões, ou seja, linhas e colunas. O nome Grid tem origem justamente na junção de linhas e colunas (que formam uma grade ou grid). O bloco "div" é usado para especificar um elemento de divisão, trata-se de um container genérico dentro do site. De certa forma não representa nada visual, é usado para agrupar itens do documento com a finalidade de estilos, usando para isso: class ou ids.

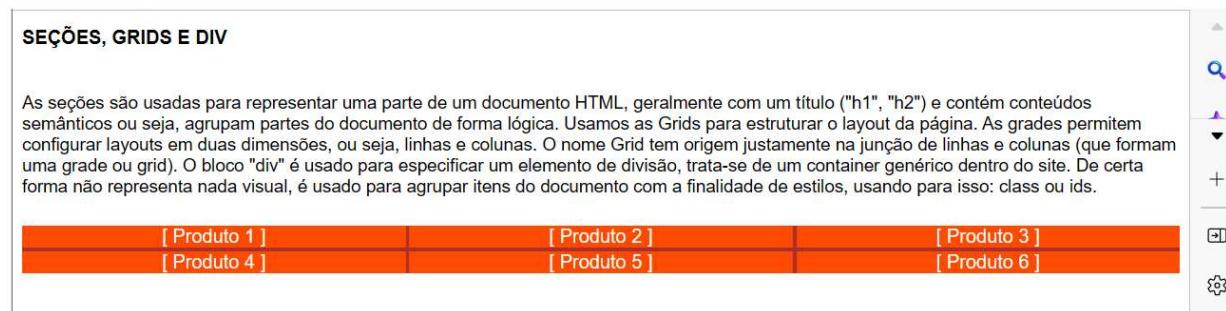
[Produto 1]	[Produto 2]	[Produto 3]
[Produto 4]	[Produto 5]	[Produto 6]



SEÇÕES, GRIDS E DIV

As seções são usadas para representar uma parte de um documento HTML, geralmente com um título ("h1", "h2") e contém conteúdos semânticos ou seja, agrupam partes do documento de forma lógica. Usamos as Grids para estruturar o layout da página. As grades permitem configurar layouts em duas dimensões, ou seja, linhas e colunas. O nome Grid tem origem justamente na junção de linhas e colunas (que formam uma grade ou grid). O bloco "div" é usado para especificar um elemento de divisão, trata-se de um container genérico dentro do site. De certa forma não representa nada visual, é usado para agrupar itens do documento com a finalidade de estilos, usando para isso: class ou ids.

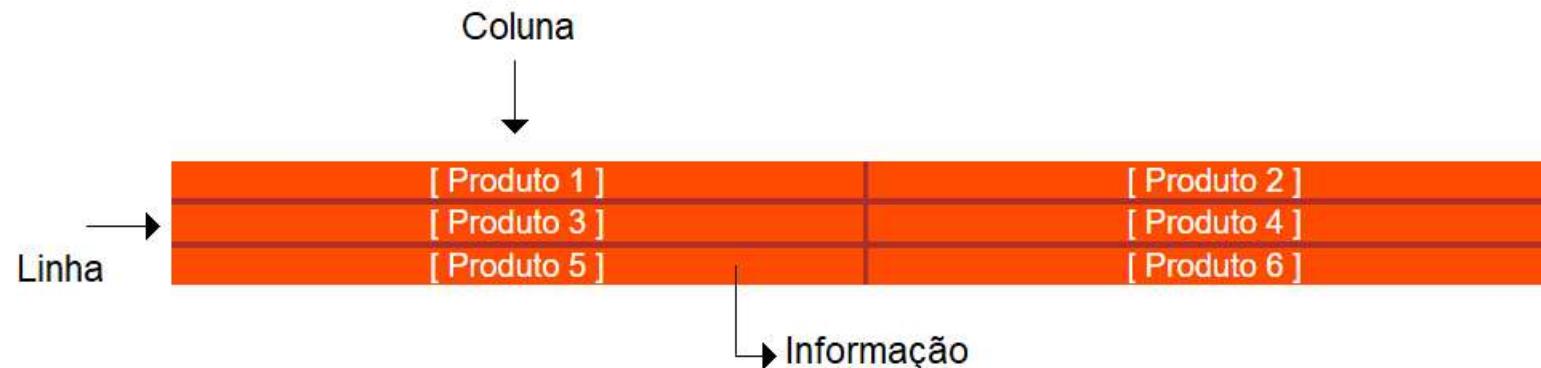
[Produto 1]	[Produto 2]	[Produto 3]
[Produto 4]	[Produto 5]	[Produto 6]



Trabalhando com Grids

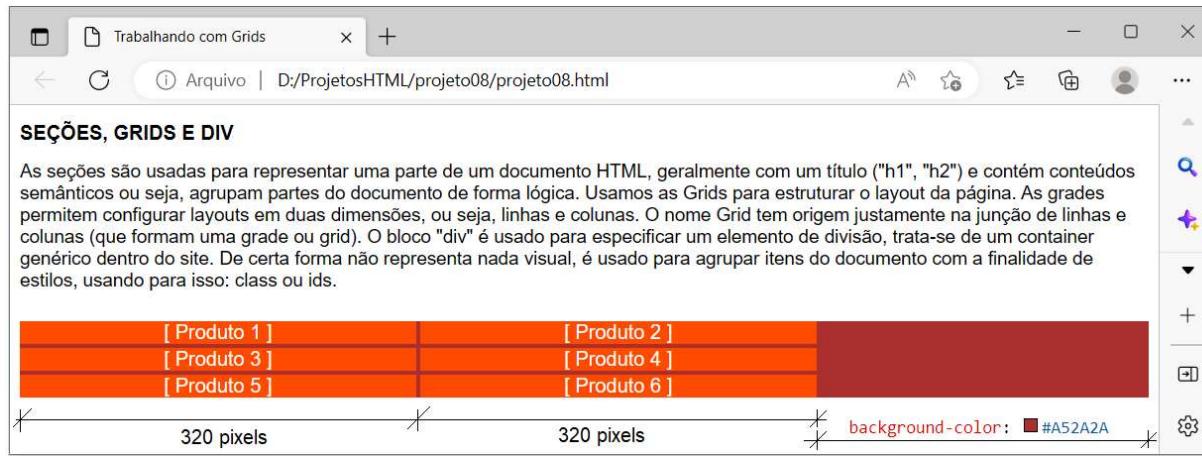
O que são Grids?

Grids são grades que vão dividir nossa tela em linhas e colunas, na interseção, estarão as informações, que podem ser um texto, uma imagem, um *hyperlink* etc.



Trabalhando com Grids

Grid-Template-Columns – Abordagem 1 (Valores absolutos)



```
.grid {  
    display: grid;  
    grid-template-columns: 320px 320px;  
    gap: 3px;  
    text-align: center;  
}
```

Observação:

Cria um *grid* com duas colunas fixas de tamanho (*width*) de 320 *pixels*. Nesse caso, não haverá renderização à medida que a página é redimensionada.

gap: É a separação entre as grades essa separação é o "gap".

Observação: o correspondente *grid-gap* (*deprecated*) foi substituído por "gap"

Veja exemplo no próximo slide ...

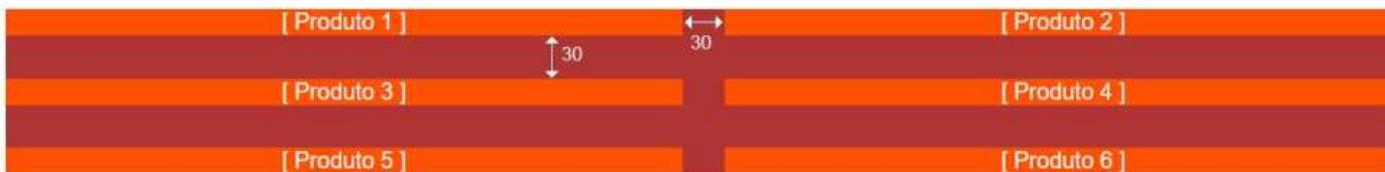
Trabalhando com Grids

Grid-Template-COLUMNS – Abordagem 1 (Valores absolutos)

Entendendo “gap”:

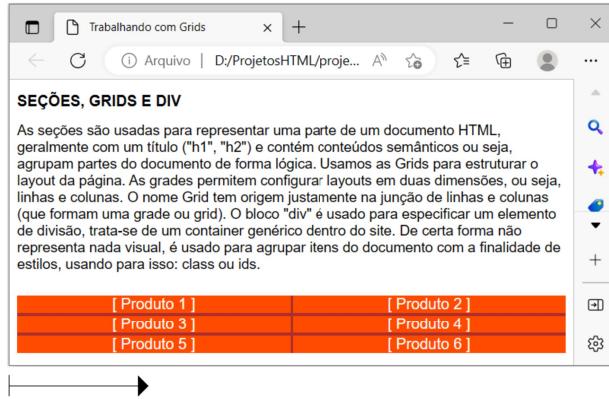
gap: É a separação entre as grades essa separação é o "gap". Observação: o correspondente grid-gap (*deprecated*) foi substituído por "gap"

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(320px, 1fr));  
  gap: 30px; ◀◀  
  text-align: center;  
}
```

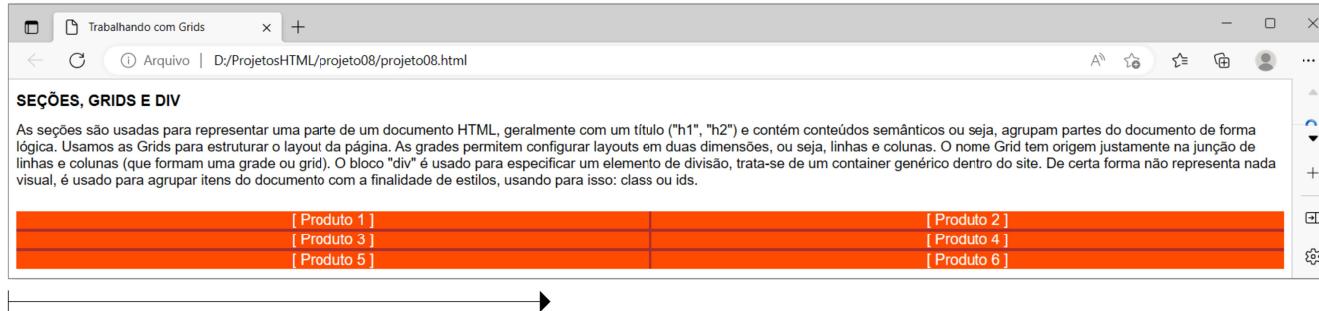


Trabalhando com Grids

Grid-Template-COLUMNS – Abordagem 2 (duas colunas usando fragmentos)



```
.grid {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    gap: 3px;  
    text-align: center;  
}
```



Observação:

Fixa a grid em duas colunas usando fragmentos, que se ajusta a renderização no processo de alterar o tamanho da página.

Para três colunas:

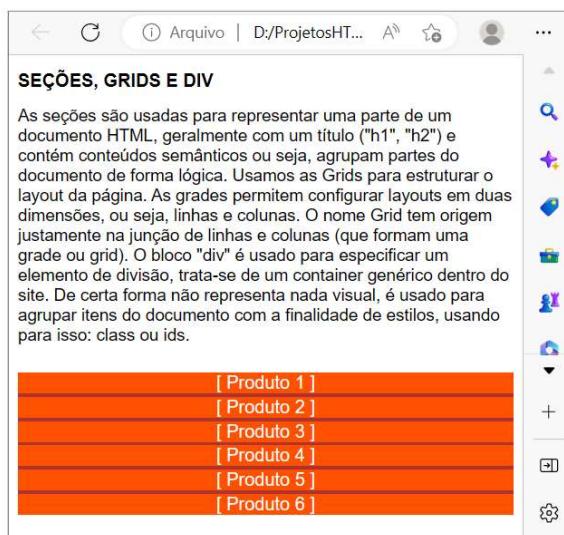
Grid-template-columns: 1fr 1fr 1fr;

fr

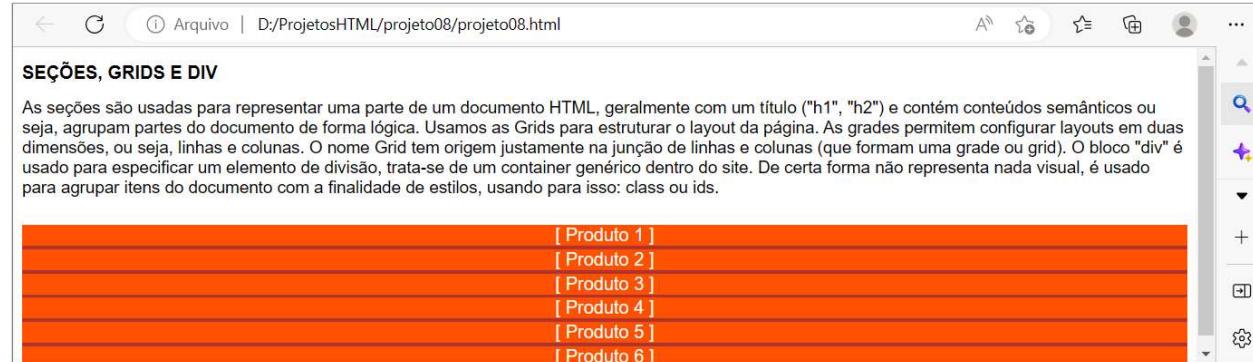
É uma unidade usada para representar uma fração ou fragmento do espaço disponível em um *container* de um *grid*.

Trabalhando com Grids

Grid-Template-COLUMNS – Abordagem 3 – Função (CSS) repeat()



```
.grid {  
    display: grid;  
    grid-template-columns:repeat(1, 1fr);  
    gap: 3px;  
    text-align: center;  
}
```



Observação:

62

Repeat() é uma função (function), gera automaticamente, a coluna (no caso uma), usando fragmentação, ou seja, aproveita o espaço atual da tela renderiza. Para duas colunas use:

Grid-template-columns: repeat(2, 1fr); e assim sucessivamente ...

Ainda há problemas com essas abordagens, por exemplo ao usar imagens nas grades ...

Projeto 9

Seções, divs e grids, a unidade “fr” (fragment),
grid-template-columns e suas variações,
grids responsivos, function "repeat()", template-columns; gap

Capítulo 10

1 – Formulários

2 – A tag "form",

3 – As propriedades: action, method (*get e post*), target

4 – Caixas de texto, menu, labels, botões, checkbox, radiobuttons etc.

5 – A tag input (*type, name, id, placeholder*)

6 – Menus

Formulários

São páginas HTML que permitem a inserção de dados (*como nome, endereço, telefone etc.*) , esse dados posteriormente serão enviados a um servidor WEB para processamento (*inserção dos dados em um banco de dados, execução de um cálculo etc.*).

Os formulários são definidos com a tag "**form**", dentro do bloco "**form**" são incluídos, *labels*, caixas de texto, botões etc. que farão parte do funcionamento do formulário.

Formulários

A tag "form"

Informa que estamos iniciando um formulário a ser enviado (*http*, métodos *get* e *post* por exemplo). Cria um bloco que contém as *tags* do formulário. Os dados contidos nesse bloco são agrupados e enviados para um destino (outro documento, servidor etc.).

Atributos da tag “form”

action: local (a partir da URL) onde serão enviados os dados do formulário. Use a "#" para informar que não haverá envio (o *back-end* fará a configuração posteriormente).

method: corresponde ao método HTTP que será usado para enviar os dados, (pode ser "get" ou "post").

get: mostra as informações da requisição (por exemplo login/senha) na barra do browser.

post: Não mostra as informações da requisição na barra do browser (*mascara os dados*).

Formulários

Sobre os métodos get e post

Get

Mostra as informações da requisição (por exemplo nome, sobrenome, login/senha etc.) na barra do browser. O método [get](#) envia os parâmetros na própria URL, é usado quando não há a preocupação de ocultar esses parâmetros, as pessoas podem acessá-las (OBTER) diretamente.

Post

Não mostra as informações da requisição na barra do browser (*mascara os dados*). O método [post](#) (POSTAR) é usado quando não for necessário passar nenhum parâmetro pela URL. Faz sentido usar o método [post](#) no envio de dados sensíveis, como senha, cpf, rg, número do cartão etc.

Formulários

Observação:

HTTP

É o protocolo responsável pela comunicação de sites na web. Quando acessamos um site, é esse protocolo que usamos.

“get” e “post” são métodos desse protocolo.

A string usada mostrando os dados enviados é denominado URI.

A [URI](#) (*Uniform Resource Identifier, ou Identificador Uniforme de Recursos*) é uma string que se refere a um recurso.

A [URL](#) é a *URI* mais comum (*Uniform Resource Locator*), se refere a um recurso localizado na Web.

Formulários

Atributos da tag “form”

target

Informa o navegador como será o direcionamento, abrindo a página na mesma janela ou aba do navegador ou abrindo uma nova janela.

`target="_blank"` : Mostra a página em uma nova janela do navegador.

`target="_self"`: Mostra a página no próprio frame.

`target="_top"`: Mostra a página usando a extensão da janela do navegador.

```
<form action="#" method="get" target="_blank">  
    <p class="form">  
        </p>  
</form>
```

Formulários

A tag "input"

Os formulários são baseados em "inputs". "Input" é um elemento de entrada que representa um campo de dados digitado, geralmente com um controle de formulário para permitir que o usuário edite os dados.

Atributos:

Id: serve para identificar o controle e aplicar estilos CSS. Também é usado para manipularmos elementos via Javascript (client-side).

name: é usado pra identificar valores do lado do servidor (server-side).

placeholder: usado para indicar ao navegador o texto de ajuda que aparecerá escrito no campo do formulário

Formulários

A tag "input"

Atributos (continuação):

size: tamanho inicial (*default* 20). É um valor em pixels, exceto para os tipos (type) text ou password, que nesse caso é medido em caracter.

É aplicável apenas para: os tipos (type): *text*, *search*, *tel*, *url*, *email* ou *password*; caso contrário, ele é ignorado.

```
<p class="form">
    <label for="nome">Nome:</label><br>
    <input type="text" id="nome" name="nome" placeholder="Insira o seu nome" size="32">
</p>
```

Formulários

Os diferentes tipos de entrada são os seguintes:

<input type="button">	<input type="checkbox">
<input type="color">	<input type="date">
<input type="datetime-local">	<input type="email">
<input type="file">	<input type="hidden">
<input type="image">	<input type="month">
<input type="number">	<input type="password">
<input type="radio">	<input type="range">
<input type="reset">	<input type="search">
<input type="submit">	<input type="tel">
<input type="time">	<input type="url">
<input type="week">	<input type="text"> (default value)

Formulários

input (continuação)

```
<p>
  <label for="matricula">Matrícula: </label><br>
  <input type="text" id="matricula" name="matricula" placeholder="Informe a matrícula" size="10">
  <br>
</p>

<p>
  <label for="nome">Nome:</label><br>
  <input type="text" id="nome" name="nome" placeholder="Informe o nome" size="20">
</p>

<p>
  <label for="fone">Telefone:</label><br>
  <input type="text" id="fone" name="fone" placeholder="Informe o telefone">
</p>

<p>
  <label for="email">Email:</label><br>
  <input type="text" id="email" name="email" placeholder="Informe o email">
</p>
```

73

Matrícula:	<input type="text" value="Informe a matrícula"/>
Nome:	<input type="text" value="Informe o nome"/>
Telefone:	<input type="text" value="Informe o telefone"/>
Email:	<input type="text" value="Informe o email"/>

Formulários

Checkbox

```
<input type="checkbox">
```

Usado para definir uma caixa de seleção que são usadas para permitir que um usuário selecione uma ou mais opções de um número limitado de opções.

```
<p>
    <label for="curso">Escolha o curso</label><br>
    <label class="aviso" for="curso">(máximo 3 cursos)</label><br><br>

    <input type="checkbox" id="uml" name="curso" value="UML">
    <label for="uml">Treinamento UML</label><br>

    <input type="checkbox" id="testes" name="curso" value="TESTES">
    <label for="testes">Treinamento Testes</label><br>

    <input type="checkbox" id="html" name="curso" value="HTML">
    <label for="html">Treinamento HTML</label><br>

    <input type="checkbox" id="css" name="curso" value="CSS">
    <label for="css">Treinamento CSS</label><br>

    <input type="checkbox" id="js" name="curso" value="JS">
    <label for="js">Treinamento Javascript</label><br>

    <input type="checkbox" id="devops" name="curso" value="DEVOPS">
    <label for="devops">Treinamento Devops</label>
</p>
```

Escolha o curso
(máximo 3 cursos)

- Treinamento UML
- Treinamento Testes
- Treinamento HTML
- Treinamento CSS
- Treinamento Javascript
- Treinamento Devops

id: usado para aplicar estilos CSS,

name: usado para referenciar o campo, identificar do lado do servidor o que será enviado na url.

for: atributo de entrada para qual o `<label>` se destina. O "for" normalmente é igual ao valor do atributo "id".

Formulários

Quadro de lista dropdown

Usado para escolher uma opção específica a partir de uma lista pré determinada.

```
<p>
    <label for="local">Escolha o local</label><br><br>

    <select id="local" name="local">
        <option value="0">Selecione o local:</option>
        <option value="CENTRO">Centro</option>
        <option value="FUNCIONARIOS">Funcionarios</option>
        <option value="STAFIGENIA">Sta Efigênia</option>
        <option value="SAVASSI">Savassi</option>
    </select>

</p>
/2
```



Formulários

Radio button

Escolha de uma única opção, as opções são mutuamente excluivas.

```
<p>
  <label for="">Escolha o Tipo de Pagamento</label><br><br>
    <input type="radio" id="pix" name="tipopg" value="PIX">
    <label for="pix">PIX</label><br>
      ↓
    <input type="radio" id="debito" name="tipopg" value="DEBITO">
    <label for="debito">Débito</label><br>
      ↓
    <input type="radio" id="debito" name="tipopg" value="DINHEIRO">
    <label for="debito">Dinheiro/Espécie</label><br>
      ↓
    <input type="radio" id="debito" name="tipopg" value="CREDITO">
    <label for="credito">Crédito</label>
</p>
```

Escolha o Tipo de Pagamento

- PIX
- Débito
- Dinheiro/Espécie
- Crédito

Para as opções serem mutuamente
Exclusivas a propriedade **name** de todas
deve ser a mesma.

Formulários

Text area

É uma variação do input (text), porém com várias linhas.

```
<p>
  <label for="">Informe a sua formação básica</label><br><br>
  <textarea class="conhecimento" name="conhecimento" id="conhecimento" cols="20" rows="5"
    title="Informe seu conhecimento (opcional), máximo 5 linhas">
  </textarea>
</p>
```

Informe a sua formação básica

Informe seu conhecimento (opcional), máximo 5 linhas

Formulários

Botões (reset e submit)

O botão com type="reset" tem a função de limpar as entradas do campos e, como acontece com o botão submit, o botão "reset" já vem pré programado.

O botão type "submit" automática a realização do submit, ou seja, dessa maneira criar um tipo botão (type="button") não é necessário "codar" algo usando o JavaScript.

```
<p>
  <input type="reset" value="[ CANCELAR ]" title="Limpar dados">
  <input type="submit" value="[ CONFIRMAR ]" title="Enviar dados">
</p>
```

[CANCELAR] [CONFIRMAR]

Enviar dados

Formulários

Formulário completo

Dados do aluno

Matrícula:

1234

name = "matricula"

Nome:

Gilmar Borba

name = "nome"

Telefone:

6070-6070

name = "fone"

Email:

gilmar.borba@fumec.br

name = "email"

name = "curso"

Escolha o curso

(máximo 3 cursos)

Treinamento UML Treinamento Testes Treinamento HTML Treinamento CSS Treinamento Javascript Treinamento Devops

Escolha o local

Funcionários

name = "local"

Escolha o Tipo de Pagamento

PIX Débito Dinheiro/Espécie Crédito name = "tipopag"

Informe a sua formação básica

Conhecimento básico em
tudo!

name="conhecimento"

[CANCELAR] [CONFIRMAR]

Formulários

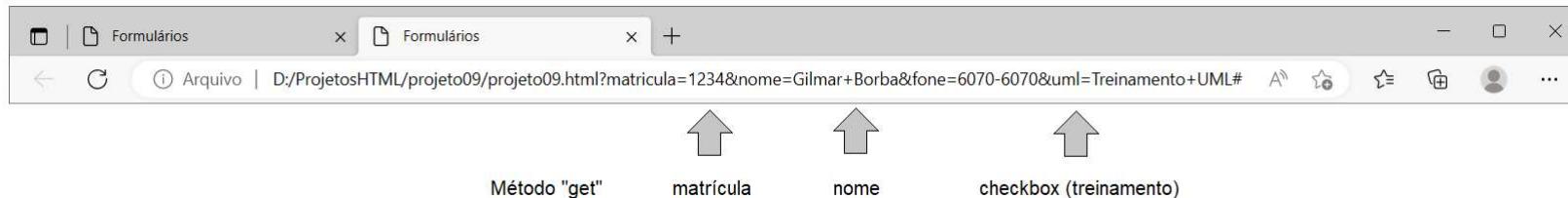
Dados enviados (método get)

```
file:///D:/projetoshtml/projeto09teste/projeto09teste.html?matricula=1234&nome=Gilmar+Borba&fone=6070-6070&email=gilmar.borba%40fumec.br&curso=UML&curso=HTML&local=FUNCIONARIOS&tipopg=DEBITO&conhecimento=Conhecimento+b%C3%A1sico+em+tudo%21++++++#
```

```
file:///D:/projetoshtml/projeto09teste/projeto09teste.html?  
matricula=1234&  
nome=Gilmar+Borba&  
fone=6070-6070&  
email=gilmar.borba40fumec.br&  
curso=UML&curso=HTML&  
local=FUNCIONARIOS&  
tipopg=DEBITO&  
conhecimento=Conhecimento+b%C3%A1sico+em+tudo%21++++++#
```

Formulários

Ao pressionar o botão Confirmar (*submit*) o método *get* envia os dados inseridos no formulário.



Opção *title* do marcador <input> ou botão *submit*.



Menus

Um menu (*ou barra de navegação*) em HTML é um recurso que permite o acesso a outras opções do *site* (sejam internos ou externos). Os estilos CSS deixam os menus com apresentação profissional, com visual mais agradável e melhor usabilidade.

Propriedade Transition (CSS)

Faz a transição entre dois estados de uma opção do menu.

transition: all 400ms ease-in-out;

all = Aplicar a todas as propriedades alteradas (pode ser usado none)

400ms = tempo (delay) de renderização da opção do menu

Ease-In: mais rapidez no ínicio da animação, há desaceleração no final.

Ease Out: menor rapidez no inicio e acelera mais rápido no final.

Ease In Out: inicio lento, meio rápido e final lento.

Menus

Hover

É um efeito “foco” (algo parecido com “On mouse move”) Ocorre quando um usuário passa o mouse sobre uma opção do menu e essa opção responde com um efeito de transição. Hover tem por objetivo melhorar a experiência do usuário.

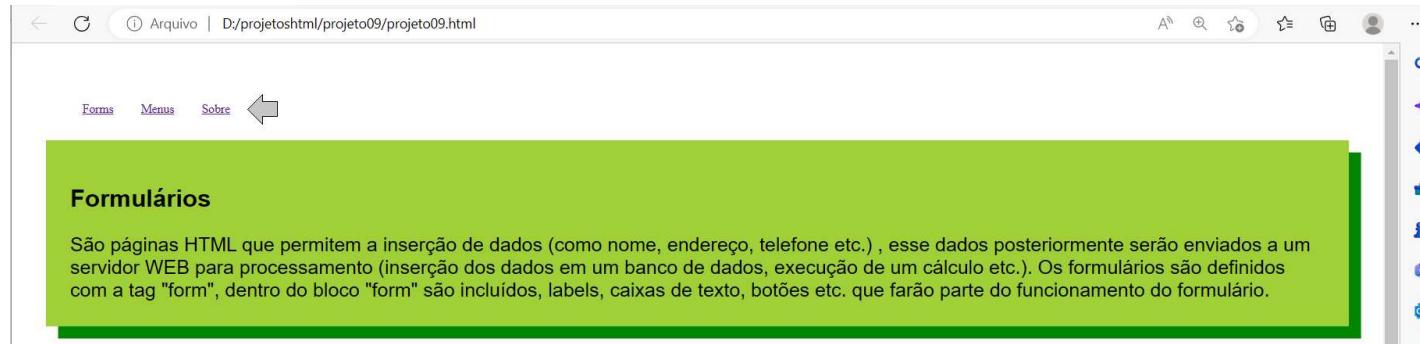
The screenshot shows a web page with a navigation bar at the top. The navigation bar includes a back arrow icon, three menu items ("Forms", "Menus", "Sobre") with a grey background, and a search icon. To the right of the menu is a vertical sidebar containing various icons: a magnifying glass, a star, a location pin, a folder, a person, a gear, a cloud, and a plus sign. Below the sidebar is a horizontal line followed by a gear icon. The main content area has a green header with the title "Formulários". The "Forms" menu item is highlighted with a grey background and a white border. The main content area contains a form for "Dados do cliente" with fields for "Matrícula", "Nome", and "Telefone", each with an input field. There is also a list of checkboxes for "Treinamento UML", "Treinamento Testes", and "Treinamento DEVOPS". At the bottom of the form are two buttons: "Cancelar" and "Confirmar". The URL "https://www.w3schools.com/html/html_forms.asp" is visible at the bottom of the page.

Menus

Passo a Passo

1 – Criar, dentro do <header> o menu usando (nesse caso) o marcador <nav> e <a href: ...>

```
<body>
  <header>
    <nav>
      <a href="https://www.w3schools.com/html/html_forms.asp">Forms</a>
      <a href="https://www.w3.org/WAI/tutorials/menus/structure/">Menus</a>
      <a href="#">Sobre</a>
    </nav>
  </header>
  . . .
```



Menus

Passo a Passo

2 – Fazer as formatações de estilo para o marcador <nav>

O marcador <nav> é uma seção de uma página com o objetivo de fornecer links de navegação, dentro do documento atual ou para outros documentos. O marcador <nav> é muito usado em menus, tabelas de conteúdo e índices.

```
nav {  
    text-align: right;  
    font-weight: bold;  
    font-family: arial;  
    font-size: 1.6rem;  
    margin-right: 2%;  
    word-spacing: 10px;  
    margin-top: 10px;  
}
```

OBERVAÇÕES:

85

Font-weight: espessura da fonte

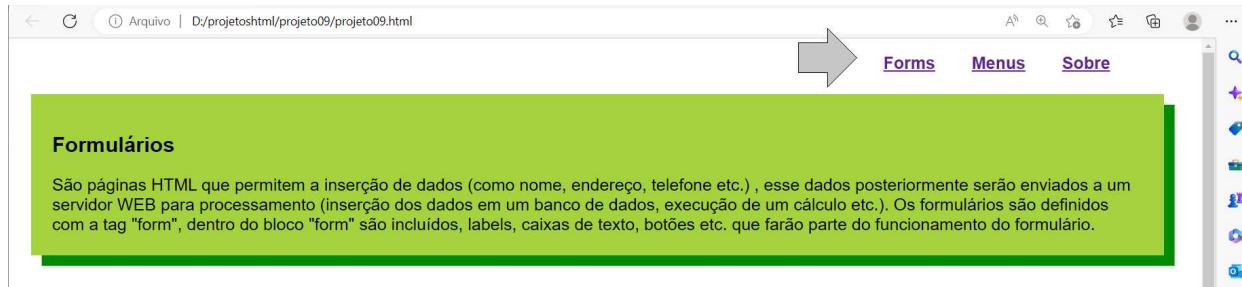
Word-spacing: espaço entre as opções do menu

Margin-right: espaço da margem direito do padding do menu

Menus

Passo a Passo

Página após as formatações de estilo para o marcador <nav>



3 - Fazer as formatações de estilo para o marcador <a>

```
a {  
    text-decoration: none;  
    color: #000000;  
    transition: all 400ms ease-in-out;  
}  
  
a:hover {  
    background-color: #DEDEDE;  
}
```

Menus

Observações

text-decoration

adiciona uma decoração ao texto.

```
text-decoration:overline;  
color: #a2cda2;  
a:hover : background-color: #DEDEDE;
```

color

É a cor da fonte do menu

transition:

all = Aplicar a todas as propriedades alteradas (pode ser usado none)

400ms = tempo (delay) de renderização da opção do menu

ease-In: mais rapidez no inicio da animação, há desaceleração no final.

ease Out: menor rapidez no inicio e acelera mais rápido no final.

ease In Out: inicio lento, meio rápido e final lento.

:hover

É uma pseudo-classe ":hover" é usado quando o usuário aponta para o elemento com o mouse mas não necessariamente o ativa, faz um sombreado na opção de menu.

Projeto 10

Formulários, a tag "form", as propriedades action, method (get e post), target, caixas de texto, menu, Labels, botões, a tag input (type, name, id, placeholder), menus

Capítulo 11

1 – Tabelas

2 – Tags: <table>, <tr> e <td>

3 – Listas

Trabalhando com Tabelas

<table>

Uma tabela é uma representação baseada em linhas e colunas. São usadas para representar dados tabulares (matriciais) ou para posicionar elementos, como imagens e textos lado a lados.

<tr>

Significa “*table row*”, define uma [linha](#) da tabela. A tabela possuirá exatamente a quantidade de linhas correspondente ao número de tags <tr> encontradas. O “**tr**” significa: [table row](#).

<td>

Significa “*table data*” e corresponde a cada uma das células da tabela. As tags <td> são inseridas dentro de tags <tr>. Dentro das tags <td> são inseridos os elementos HTML, como imagens, animações e textos.

Trabalhando com Tabelas

Exemplos

```
<table border="1">
  <tr>
    <td>Célula 1</td>
  </tr>
  <tr>
    <td>Célula 2</td>
  </tr>
  <tr>
    <td>Célula 3</td>
  </tr>
</table>
```



Célula 1
Célula 2
Célula 3

```
<table border="3">
```

```
  <tr>
    <td>Célula1</td>
    <td>Célula2</td>
  </tr>
  <tr>
    <td>Célula3</td>
    <td>Célula4</td>
  </tr>
</table>
```



Célula1	Célula2
Célula3	Célula4

```
<table border="2">
  <tr>
    <td>Célula1</td>
    <td>Célula2 </td>
    <td>Célula3</td>
  </tr>
</table>
```



Célula1	Célula2	Célula3
---------	---------	---------

Tabelas X Grids

Dessa forma, as [tabelas](#) *HTML* são usadas para organizar os dados em um modelo matricial ou tabular, como uma tabela de preços, uma tabela de ranqueamento relacionada a alguma área etc. As tabelas não são usadas para desenhar layouts, nesse caso há outros elementos para esse fim, como as [Grids](#) apresentadas no capítulo 9.

As [Grids](#) são elementos *HTML/CSS* criadas para ajudar no *layout* ou apresentação dos dados posicionados em duas dimensões, linhas e colunas.

Trabalhando com Listas

Lista é um recurso que permite criar tópicos de textos, que podem ser ordenados ou não. São usados também no contexto dos menus em *HTML*.

`` é usado para representar um item que faz parte de uma lista. Este item deve estar contido em um elemento pai (que pode ser `` lista ordenada, `` lista desordenada).

`` *ordered list*

`` *unordered list*

A diferença entre `` e `` está no contexto semântico. Ao alterar um item de uma lista, se a significação for alterada devemos usar ``, caso contrário, usaremos ``.

Trabalhando com Listas

 ordered list

 unordered list

```
<section>
  <ul>
    <h2>Linguagens</h2><br>
    <li>SQL</li>
    <li>Java</li>
    <li>JavaScript</li>
    <li>GO</li>
    <li>R</li>
    <li>Python</li>
  </ul>
</section>
```



Linguagens

- SQL
- Java
- JavaScript
- GO
- R
- Python

```
<section>
  <ol>
    <h2>Linguagens</h2><br>
    <li>SQL</li>
    <li>Java</li>
    <li>JavaScript</li>
    <li>GO</li>
    <li>R</li>
    <li>Python</li>
  </ol>
</section>
```



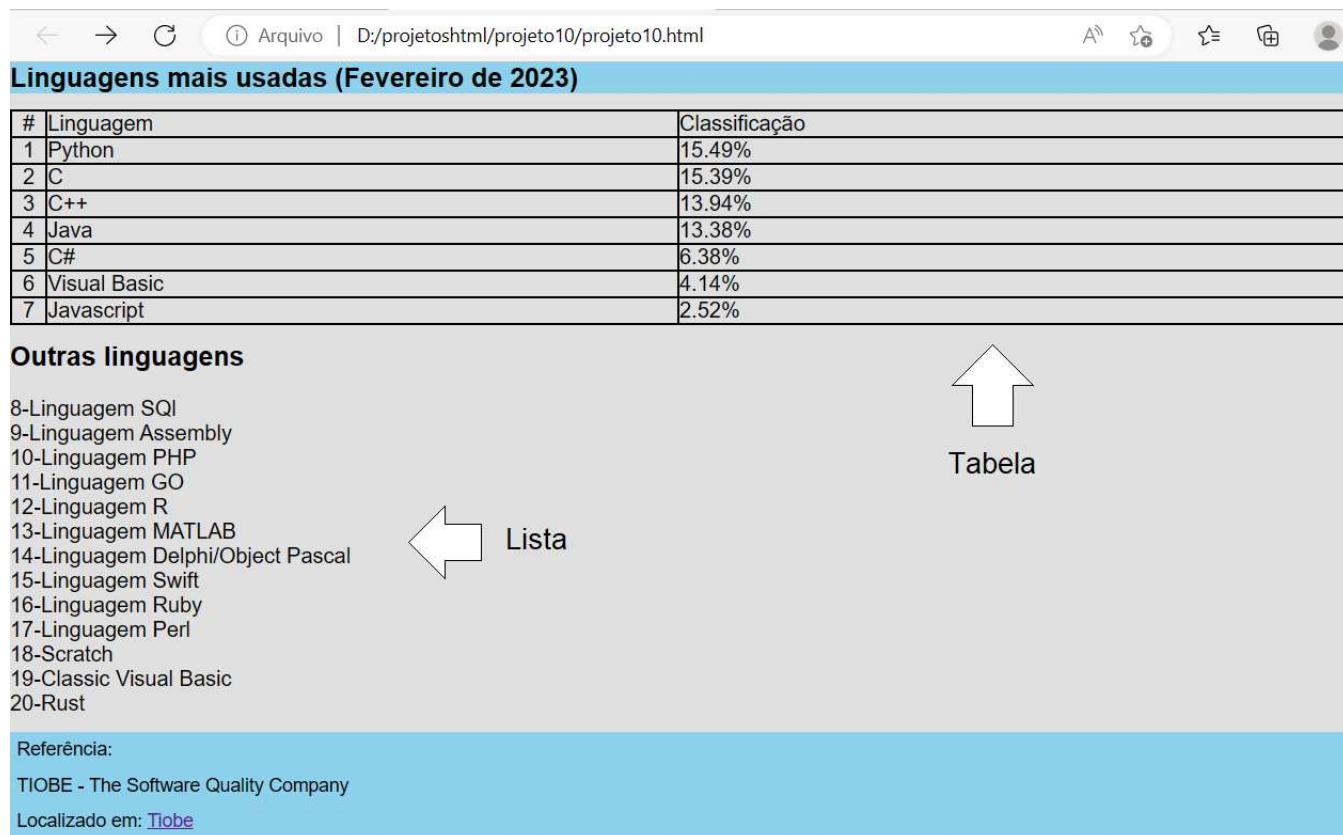
Linguagens

1. SQL
2. Java
3. JavaScript
4. GO
5. R
6. Python

Projeto 11

Trabalhando com tabelas. Veja a imagem no próximo slide.

Trabalhando com Tabelas e Listas



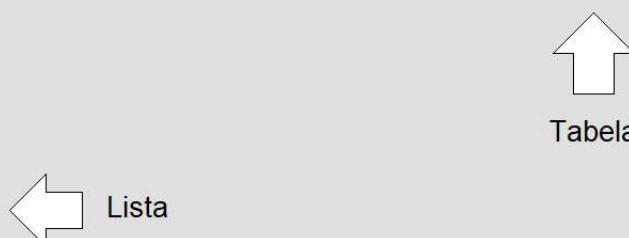
A screenshot of a web browser displaying a table titled "Linguagens mais usadas (Fevereiro de 2023)". The table lists the top 7 most used languages with their classification percentages. Below the table, there is a section titled "Outras linguagens" listing 20 additional languages. At the bottom, there is a reference note and a link to the source.

#	Linguagem	Classificação
1	Python	15.49%
2	C	15.39%
3	C++	13.94%
4	Java	13.38%
5	C#	6.38%
6	Visual Basic	4.14%
7	Javascript	2.52%

Outras linguagens

- 8-Linguagem SQL
- 9-Linguagem Assembly
- 10-Linguagem PHP
- 11-Linguagem GO
- 12-Linguagem R
- 13-Linguagem MATLAB
- 14-Linguagem Delphi/Object Pascal
- 15-Linguagem Swift
- 16-Linguagem Ruby
- 17-Linguagem Perl
- 18-Scratch
- 19-Classic Visual Basic
- 20-Rust

Referência:
TIOBE - The Software Quality Company
Localizado em: [Tiobe](#)



The diagram consists of two arrows pointing from the text labels "Lista" and "Tabela" to their respective counterparts in the screenshot. A white arrow points upwards from the word "Lista" to the "Outras linguagens" list. Another white arrow points upwards from the word "Tabela" to the table at the top of the page.

Capítulo 12

- 1 – Publicando o Site
- 2 – Domínio e Hospedagem
- 3 – Publicação Gratuita de Sites (Sugestão)

Publicando o Site

Como publicar

1 - Comprar o domínio (nome do site)

Exemplo: minhaempresa.com.br

2 - Comprar a hospedagem

Espaço na internet para colocar o site
(mensal, semestral anual ...)



Publicando o Site – Hospedagem gratuita

Não é profissional, não conseguimos monetizar etc. ...

.. mas quando publicamos, conseguimos testar o site em diversos recursos diferentes, telas, navegadores, dispositivos reais etc.

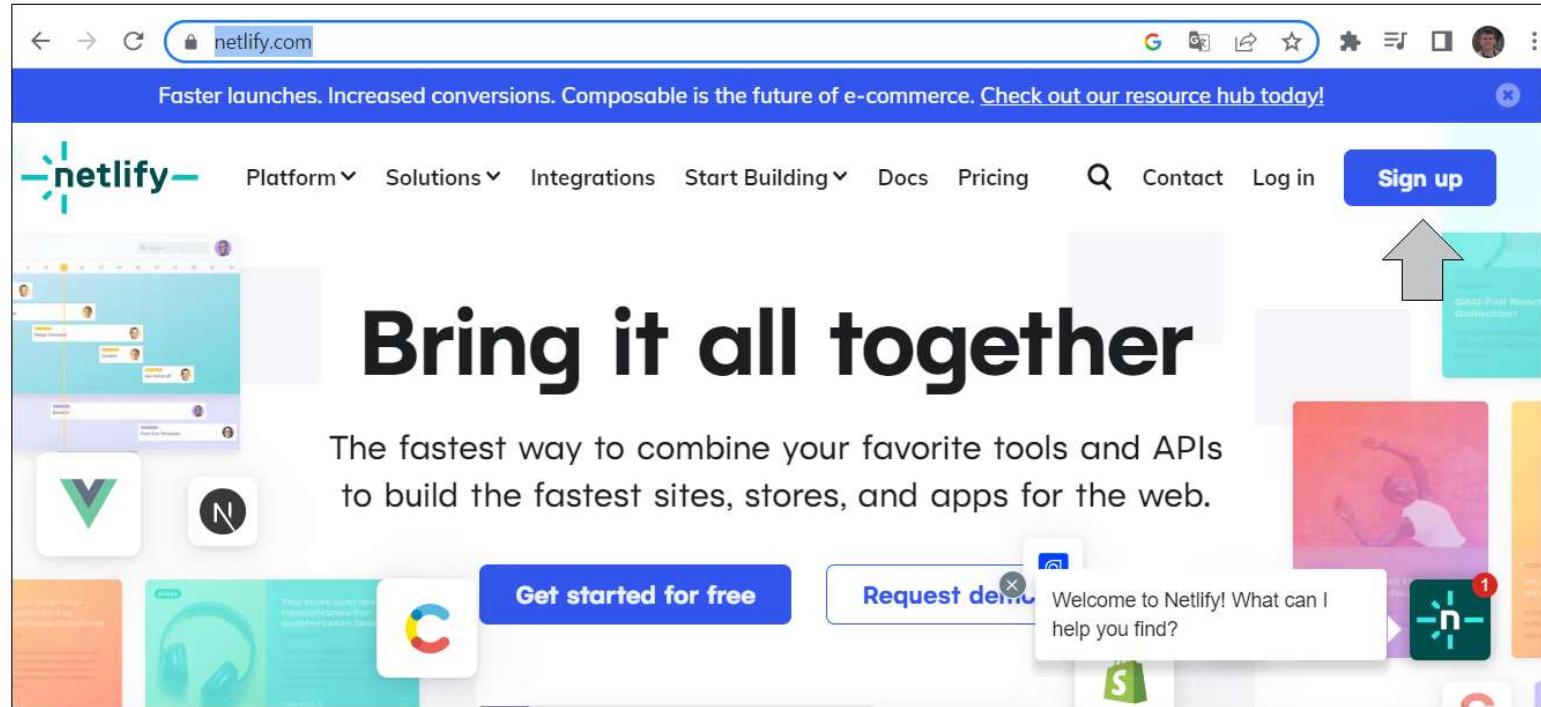
Sugestão:

<https://www.netlify.com/>

Publicando o Site – Hospedagem gratuita

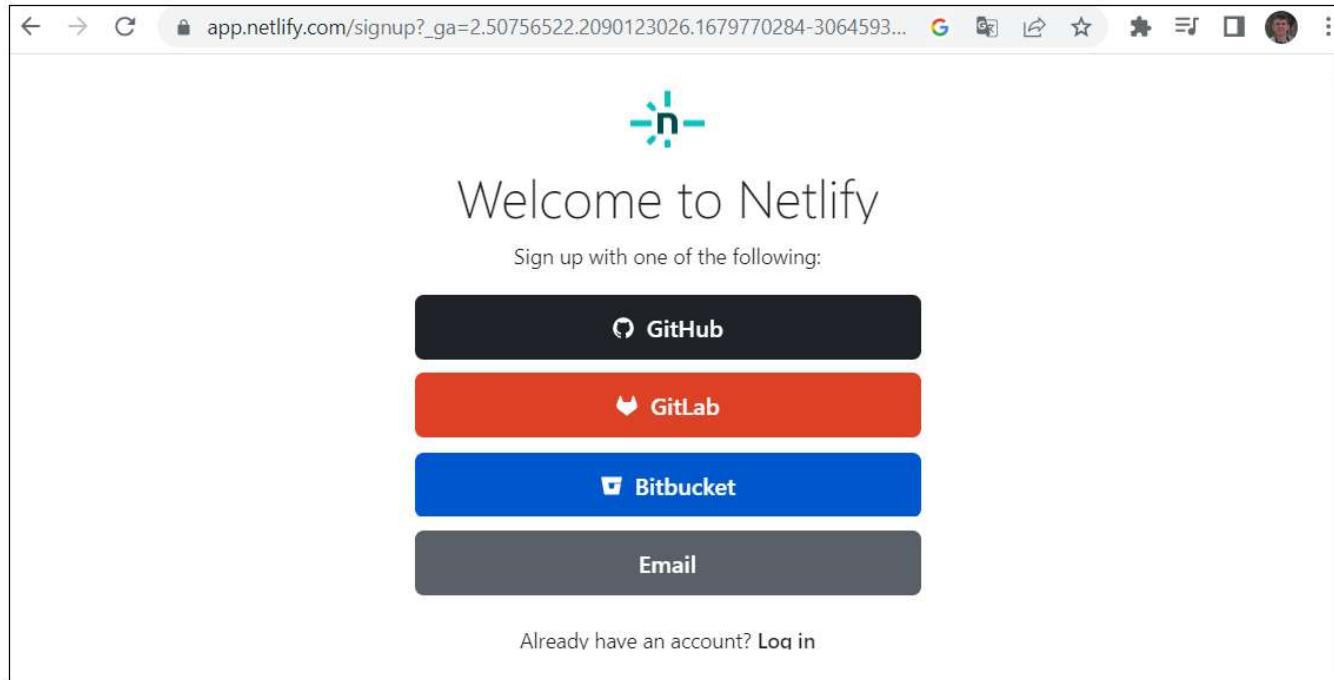
Acessar o endereço abaixo, acionar o botão sign up (inscrever-se):

<https://www.netlify.com/>



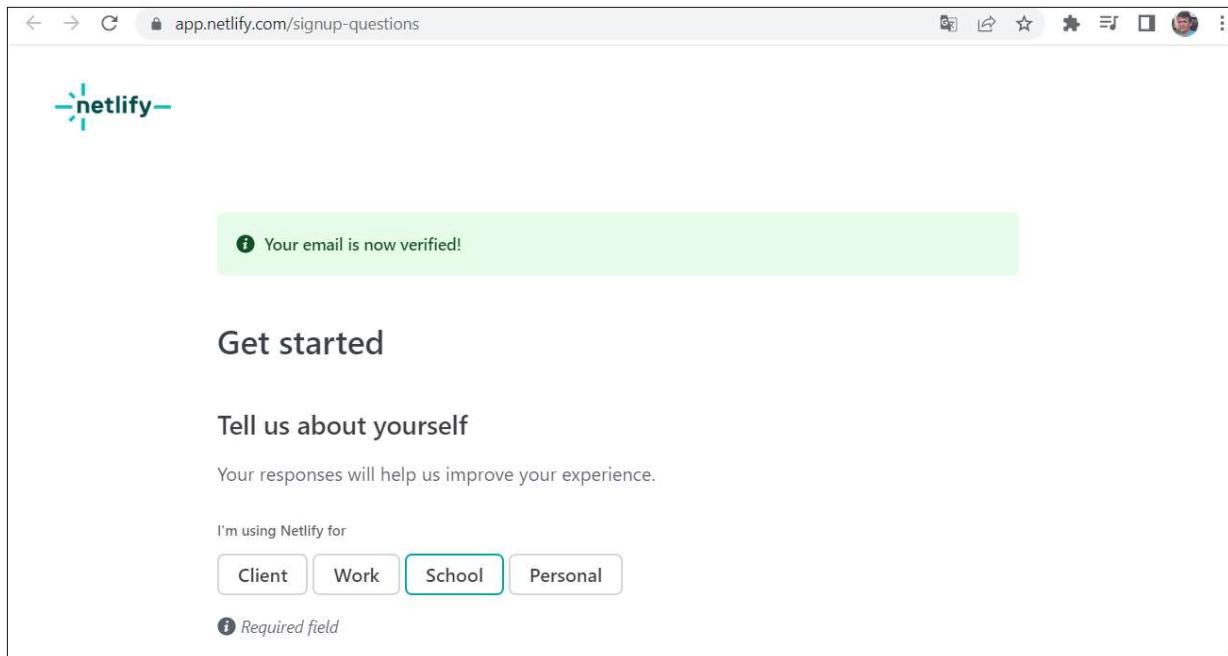
Publicando o Site – Hospedagem gratuita

Escolher o tipo de inscrição (*usaremos o e-mail para exemplo*):



Publicando o Site – Hospedagem gratuita

Após informar o e-mail válido, e uma senha acesso de acesso (10 caracteres) o e-mail será verificado:



Publicando o Site – Hospedagem gratuita

Informe o nome da equipe de desenvolvimento ou do seu projeto:

Personalize your team space

Make your team your own.

Name your team

projeto01

 People who use Netlify for school often use a school, cohort or project name.

[Set up and continue](#)

Publicando o Site – Hospedagem gratuita

O projeto pode ser: [importado do GitHub](#), iniciado a partir de um [Template](#) ou [implantado \(deployed\)](#) manualmente a partir do seu computador:

The screenshot shows the Netlify team overview interface. At the top, there's a navigation bar with links for Team overview, Sites, Builds, Integrations, Domains, Members, Audit log, Billing, and Team settings. A prominent 'Upgrade' button is at the top right, along with a search bar and various icons. A message encourages trying the Pro free trial for one month. Below the header, the team name 'projeto01' is displayed with a 'Starter' badge. A 'Welcome to Netlify' message is shown, along with a note about bandwidth usage (0 KB/reqs). To the right, there's a 'Team members' section showing 1 member. A large blue button with a white 'N' logo is visible. A tooltip for 'Import environment variables' provides instructions on how to import .env files. The main area is titled 'Sites' and contains two main options: 'Import an existing project' (with a 'Import from Git' button) and 'Start from a template' (with a 'Browse templates' button). A note at the bottom says '...or deploy manually'.

Usaremos a última opção ...

Publicando o Site – Hospedagem gratuita

Selecione e confirme, ou arraste e solte a pasta do projeto

IMPORTANTE: A página principal do projeto deve ter o nome: [index.html](#)

Sites →

Import an existing project
[Import from Git](#)

Start from a template
[Browse templates](#)

...or deploy manually

Drag and drop your site output folder here
Or, [browse to upload](#)



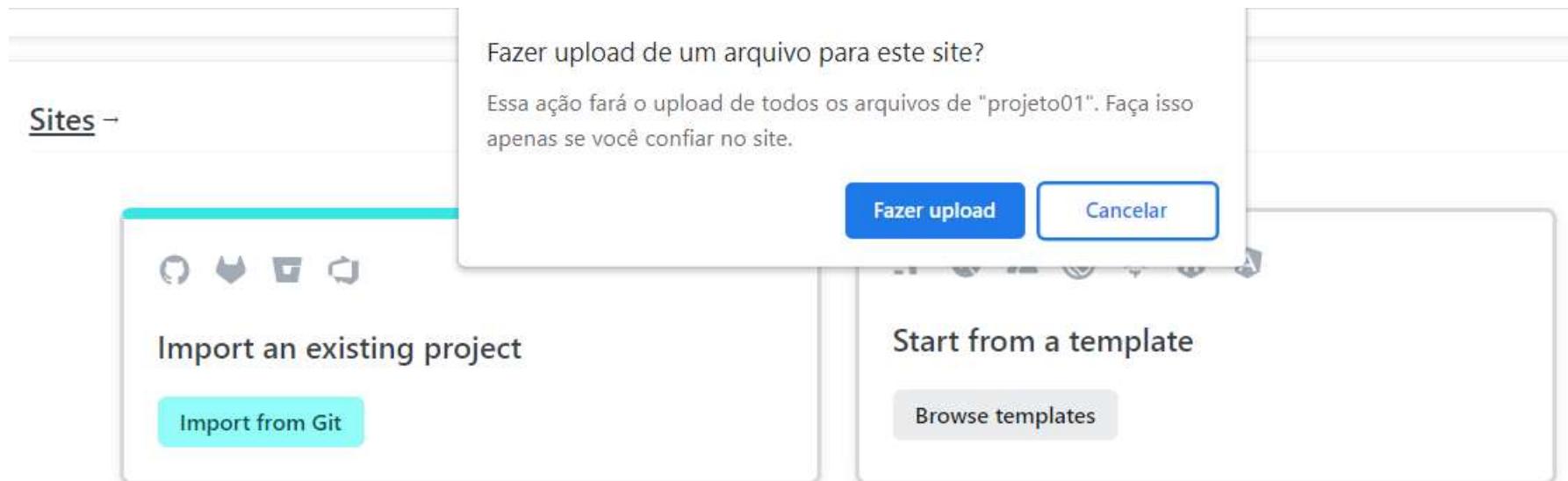
Members →

gilmar ...@gmail.com
Owner

[Upgrade to add members](#)

Publicando o Site – Hospedagem gratuita

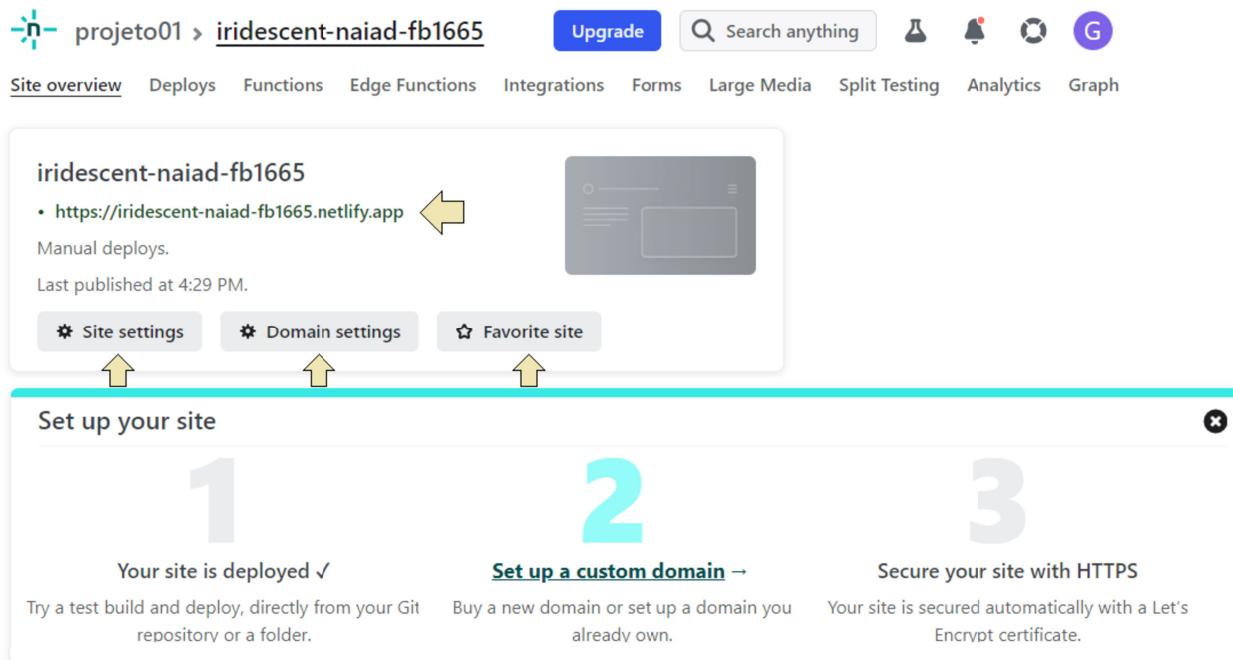
Confirme a operação



Publicando o Site – Hospedagem gratuita

A url do site foi criada, nesse caso: <https://iridecente-naiad-fb1665.netlify.app>

Esse nome pode ser alterado na opção Set Settings



The screenshot shows the Netlify project dashboard for 'projeto01 > iridescent-naiad-fb1665'. The top navigation bar includes 'Upgrade', a search bar, and various icons. Below the navigation, there are links for 'Site overview', 'Deploys', 'Functions', 'Edge Functions', 'Integrations', 'Forms', 'Large Media', 'Split Testing', 'Analytics', and 'Graph'. The main content area displays the deployed site 'iridescent-naiad-fb1665' with the URL <https://iridescent-naiad-fb1665.netlify.app>. A yellow arrow points to this URL. Below the URL, there's a note about manual deployment and the last publish time at 4:29 PM. At the bottom of the dashboard, there are three buttons: 'Site settings', 'Domain settings', and 'Favorite site', each with a yellow arrow pointing to it. A large callout box titled 'Set up your site' provides three steps: 1. Your site is deployed ✓ (with a green checkmark), 2. Set up a custom domain → (with a blue arrow), and 3. Secure your site with HTTPS (with a grey checkmark). The callout also includes instructions for testing builds and securing the site with a Let's Encrypt certificate.



Publicando o Site – Hospedagem gratuita

Agora a url do site será: <https://meuprojeto01basico.netlify.app>

Esse nome pode ser alterado na opção Set Settings

Site information

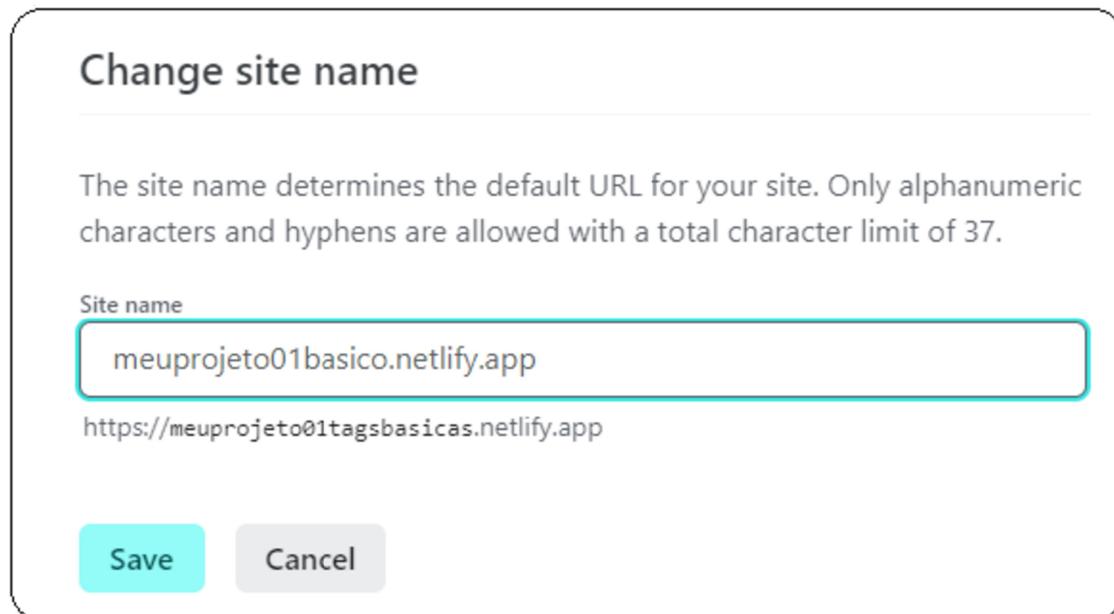
Site name:	dazzling-haupia-162789
Owner:	projeto01
Site ID:	c4bede75-dbbd-4f77-b759-869447d3f063 
Created:	Today at 4:55 PM
Last update:	Today at 4:55 PM

 [Change site name](#) [Transfer site ▾](#)

Publicando o Site – Hospedagem gratuita

Agora a “url” do site será: <https://meuprojeto01basico.netlify.app>

Esse nome pode ser alterado na opção Set Settings



Publicando o Site – Hospedagem gratuita

Note que o novo nome foi propagado para o projeto:

Settings for meuprojeto01basico

meuprojeto01basico.netlify.app

Manual deploys.

Owned by [projeto01](#).

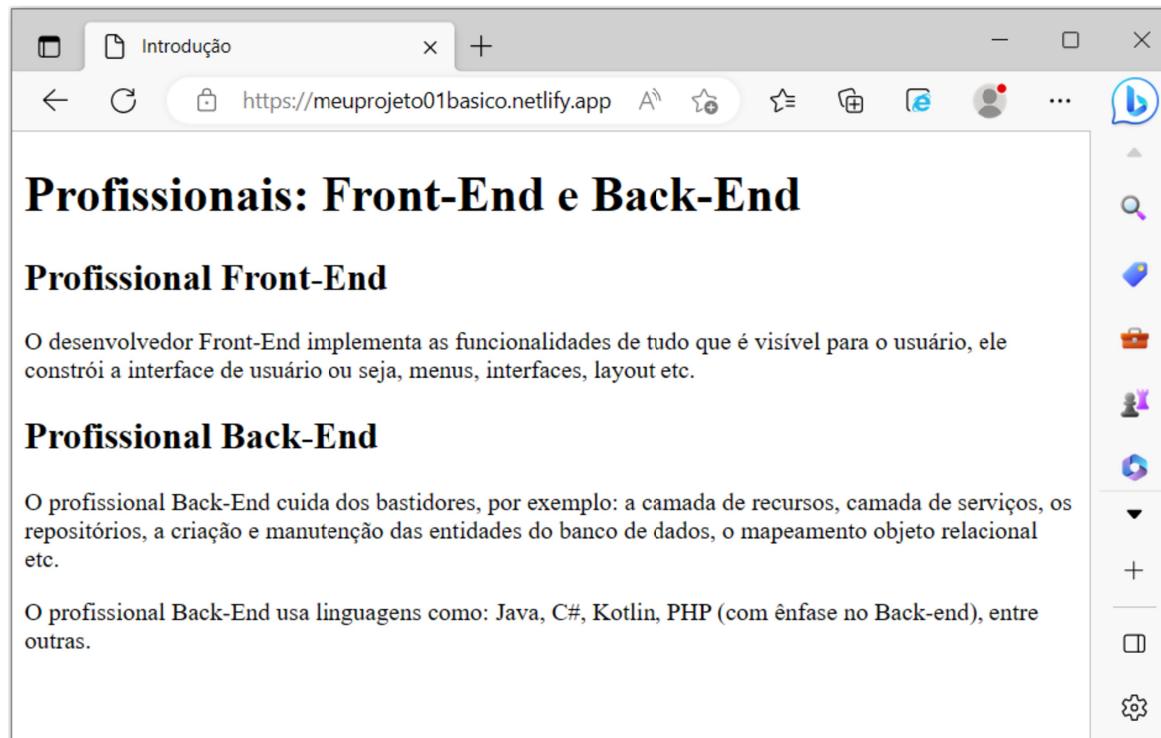
Last update at 4:56 PM (a few seconds ago)



Publicando o Site – Hospedagem gratuita

Site em execução no navegador Microsoft Edge:

URL: <https://meuprojeto01basico.netlify.app>



REFERÊNCIAS

Construção de Páginas Web com HTML - Nível básico (curso a distância)

Localizado em: <http://www.nce.ufrj.br/ginape/cursohtml/conteudo/ligacoes/absrel.htm>

Acessado em: Janeiro de 2023

HTML Basics and Web Standards Concepts teaching materials

<https://www.w3.org/community/webed/wiki/>

[HTML_Basics_and_Web_Standards_Concepts_teaching_materials#Intro](#)

Internacionalização - Dicas rápidas para a Web

Localizado em: <https://www.w3.org/International/quicktips/index.pt>

Acessado em: janeiro de 2023

Iperius Backup

Entendendo os conceitos entre os modelos TCP/IP e OSI

Localizado em: <https://www.iperiusbackup.net/pt-br/entendendo-os-conceitos-entre-os-modelos-tcpip-e-osi/>

Acessado em: Dezembro de 2022

JavaScript - mdn web docs

Localizado em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Acessado em: Dezembro de 2022

REFERÊNCIAS

Learn Responsive Design

Localizado em: <https://web.dev/learn/design/>

Acessado em: janeiro de 2023

Learn to Code - With the world's largest web developer site.

Localizado em: <https://www.w3schools.com/>

Acessado em: Dezembro de 2022

Tabela de Cores HTML (hexadecimal e RGB) - Amostras de cores para HTML e CSS em código hexadecimal e código RGB

Localizado em: <https://erikasarti.com/html/tabela-cores>

Acessado em: dezembro de 2022

Udemy

HTML e CSS Essencial - Front End Completo para Iniciantes

Professor Wagner Cardoso

Fim . . .