

DESIGN PARA WEB

PROF RENAN COSTA FILGUEIRAS

TRABALHO DISCIPLINA

-

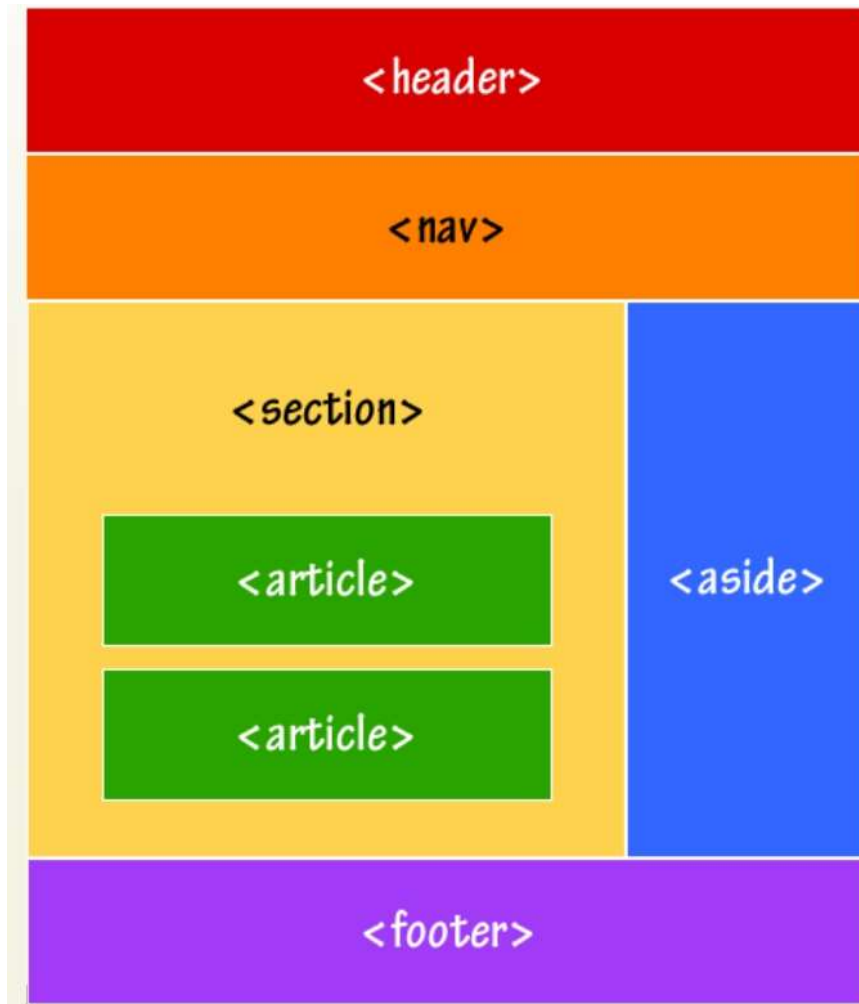
ENTREGA 27/11/2022

CRIAR UMA PÁGINA PESSOAL (Portfólio Pessoal)

5 pontos – Envio Domingo (27/11/2022)

Exigências

- **Permanecer com as exigências do trabalho anterior**
- **Possuir menu (Identificação, Formação Acadêmica, Atuação Profissional, Hobbies)**
- **Possuir migalha de pão**
- **Utilizar um layout de página**
- **Utilizar bootstrap ou outro framework CSS (materialize, bulma)**
- **Ser responsível**
- **Criatividade é um diferencial**
- **Estar hospedado em um servidor**
- **Possuir um domínio para acesso na rede WWW**
- **Pode utilizar templates, frameworks e exemplos**



BLOCO SEMÂNTICO DO HTML5

HEADER

O `<header>` é utilizado para representar o cabeçalho de um documento ou seção declarado no HTML. Nele podemos inserir elementos de `<h1>` a `<h6>`, até elementos para representar imagens, parágrafos ou mesmo listas de navegação.

Exemplo de uso de `<header>`:

```
<header>  
    <h1>Título da página</h1>  
    <h2>Subtítulo da página</h2>  
</header>
```

MAIN

O elemento `<main>` especifica o conteúdo principal e, conseqüentemente, de maior relevância dentro da página. Para ser considerada bem construída, uma página deve apresentar apenas um conteúdo principal.

Exemplo de uso de `<main>`:

```
<main>  
  <h1>Titulo</h1>  
  
  <p>Lorem ipsumectetur adipisicing elit, sed do eiusmod ...</p>  
  
  <article>  
    <h2>Subtítulo</h2>  
    <p>Duis aute irure dolor in reprehlit esse cillum...</p>  
  </article>  
</main>
```

NAV

Este elemento agrupa recursos de navegação do sítio;

Representa um grupo de links que conduzem a outras páginas da aplicação, ou a seções da página corrente;

Correspondem à área de "menus" da aplicação web.

Exemplo do uso do <nav>:

```
<nav>  
  <ul>  
    <li><a href="#">pagina 1</a></li>  
    <li><a href="#">pagina 2</a></li>  
    <li><a href="#">pagina 3</a></li>  
    <li><a href="#">pagina 4</a></li>  
  </ul>  
</nav>
```

SECTION

O elemento `<section>` representa uma seção dentro de um documento e geralmente contém um título, o qual é definido por meio de um dos elementos entre `<h1>` e `<h6>`. Podemos utilizar o `<section>`, por exemplo, para descrever as seções/tópicos de um documento.

Exemplo de uso de `<section>`:

```
<section>  
  <h3>Seção 1</h3>  
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing  
elit, sed do eiusmod ...</p>  
</section>
```

ARTICLE

Utilizamos o elemento `<article>` quando precisamos declarar um conteúdo que não precisa de outro para fazer sentido em um documento HTML, por exemplo, um artigo em um blog. É recomendado identificar cada `<article>` com um título.

Exemplo

```
<article>  
  <h3>Título do artigo 1</h3>  
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod ...</p>  
</article>  
<article>  
  <h3>Título do artigo 2</h3>  
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod ...</p>  
</article>
```


ASIDE

O elemento `<aside>` é utilizado quando precisamos criar um conteúdo de apoio/adicional ao conteúdo principal. Por exemplo, ao falar de HTML semântico, podemos indicar ao leitor outros conteúdos sobre a linguagem HTML como sugestão de leitura complementar.

Exemplo de uso de `<aside>`:

```
<aside>
  <nav>
    <ul>
      <li>Link 1</li>
      <li>Link 2</li>
      <li>Link 3</li>
      <li>Link 4</li>
    </ul>
  </nav>
</aside>
```

FOOTER

O elemento `<footer>` representa um rodapé de um documento, como a área presente no final de uma página web.

Normalmente é utilizado para descrever informações de autoria, como nome e contato do autor, e data de criação do conteúdo.

Exemplo de uso de `<footer>`:

```
<footer>
```

```
  <p>Empresa xyz, endereço: QSD 50 LOTE 15</p>
```

```
  <p>Publicado em 25/07/2019</p>
```

```
</footer>
```

Cursos de programação

Java

Curso de JavaServer Faces

Neste curso você aprenderá a desenvolver aplicações com JSF

Curso de Spring

Neste curso você aprenderá utilizar o framework Spring.

PHP

Curso básico de PHP

Conheça a linguagem de programação PHP.

Curso de PDO

Aprenda a acessar bancos de dados com PHP Data Objects.

MinhaEmpresa.com.br. Todos os direitos reservados.

```
<html>
<body>
  <header>
    <h1>Cursos de programação</h1>
  </header>
  <main>
    <section>
      <h2>Java</h2>
      <article>
        <h3>Curso de JavaServer Faces</h3>
        <p>Neste curso você aprenderá a desenvolver aplicações com JSF.</p>
      </article>
      <article>
        <h3>Curso de Spring</h3>
        <p>Neste curso você aprenderá utilizar o framework Spring.</p>
      </article>
    </section>
    <section>
      <h2>PHP</h2>
      <article>
        <h3>Curso básico de PHP</h3>
        <p>Conheça a linguagem de programação PHP.</p>
      </article>
      <article>
        <h3>Curso de PDO</h3>
        <p>Aprenda a acessar bancos de dados com PHP Data Objects.</p>
      </article>
    </section>
  </main>
  <footer>
    <p>MinhaEmpresa.com.br. Todos os direitos reservados.</p>
  </footer>
</body>
</html>
```

GRID SYSTEM

Esse mecanismo funciona como **uma espécie de tabela abstrata**, e é responsivo (responsive), orientado a dispositivos móveis (mobile first) e se ajusta de acordo com a tela (fluid), quando ela muda de tamanho ou de orientação.

MOBILE FIRST

Isso quer dizer que o framework assume, inicialmente, que a tela é de um dispositivo móvel, com tamanho pequeno. Assim, ele adapta todos os conteúdos para o tamanho menor.

Depois, ele verifica o tamanho real da tela e vai ajustando os itens para que fiquem posicionados corretamente, conforme o tamanho e a resolução.

Quando você for projetar, e implementar, o layout e o design do site (ou app) você deve projetar primeiro para as telas menores (celulares), depois para telas médias (tablets) e, por fim, para telas maiores (desktops, etc.).

CONTAINER

O container é uma *div*, que garante que o seu layout vai ficar alinhado na página, e com margens para as laterais. Ele também centraliza o conteúdo na tela do browser.

Dependendo do tamanho da tela, o container definirá automaticamente as larguras do seu layout, para que o conteúdo seja melhor visualizado.

Você deve usar um container para englobar o posicionamento de todos os elementos do layout da página.

CONTAINER

Então, dependendo do dispositivo e da orientação tela, o seu container pode ficar com o tamanho de acordo com a tabela abaixo:

Em Celulares	Em Tablets	Desktops	Telas Grandes
Largura Automática	Largura Máx. de 750px	Largura Máx. de 970px	Largura Máx. de 1170px

Para criar um container com largura fixa, e ajustada ao tamanho da tela, você pode criar uma div com a classe **.container**, como no exemplo:

```
<div class="container">  
...  
</div>
```

Ou, você pode usar a classe **.container-fluid** para fazer o container (e o layout) ficar com 100% de largura:

```
<div class="container-fluid">  
...  
</div>
```

A SUA PÁGINA DEVE TER, PELO MENOS, UM CONTAINER GERAL (TAMBÉM CONHECIDO COMO WRAPPER). ELE ABRANGE TODO O ESCOPO DA PÁGINA. E TODOS OS OUTROS ELEMENTOS VISUAIS DA PÁGINA DEVEM ESTAR DENTRO DELE.

ROW

As rows (linhas), no Bootstrap, definem as divisões horizontais do seu layout.

Essas rows devem ficar dentro do container, e podem ser aplicadas a qualquer tag que defina estrutura, como div, header e footer.

Para criar uma *row* você pode definir uma *div* com a classe **.row**, como no exemplo:

```
<div class="container">
  <div class="row">
    ...
  </div>
</div>
```

Seu layout pode ter quantas rows forem necessárias. E você pode colocar *rows* dentro de *rows*, também.

As *rows* sempre irão ficar uma abaixo da outra.

COLUMNS

As columns (colunas), no Bootstrap, definem as divisões verticais das rows (linhas) do seu layout.

Columns devem estar sempre dentro das rows, e elas definem espaços na row para que você coloque os itens visuais ou conteúdos que foram projetados.

Então, no Bootstrap, você tem linhas (rows) e colunas (columns) para definir um layout. É, exatamente, como uma grade ou matriz (ou tabela), porém, utilizando div's com classes.

Para criar uma *column* você pode criar uma *div* com os prefixos pré-definidos pelo Bootstrap, como no exemplo:

```
<div class="container">
  <div class="row">
    <div class="col-md-6"> </div>
    <div class="col-md-6"> </div>
  </div>
</div>
```

COLUMNS

Os prefixos de colunas servem para indicar em quais tipos de tela a coluna vai se manter posicionada como no design principal. Os prefixos têm o seguinte padrão:

Para Celulares	Para Tablets	Para Desktops	Para Telas Grandes
<code>.col-xs-*</code>	<code>.col-sm-*</code>	<code>.col-md-*</code>	<code>.col-lg-*</code>

Você deve substituir os asteriscos pelo tamanho da coluna (de 1 a 12).

O mais comum é utilizar o prefixo `.col-md-*` para os sites e web apps, porque ele mantém o design principal em computadores e nos principais tablets. Apenas nos celulares o conteúdo passa a ficar vertical, ou seja, “um embaixo do outro” em uma única coluna.

BOOTSTRAP GRID

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

LAYOUT BÁSICO

```
<div class="container-fluid">
  <header class="row">
  </header>
  <div class="row">
    <div role="main">
      <!-- conteudo principal -->
    </div>
  </div>
  <footer class="row">
  </footer>
</div>
```

LAYOUT BÁSICO

`<header>`

`<div role="main">`

`<footer>`

LAYOUT DUAS COLUNAS



LAYOUT TRÊS COLUNAS

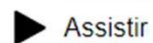


NETFLIX[Início](#) [Séries](#) [Filmes](#) [Bombando](#) [Minha lista](#) [Navegar por idiomas](#)**N** SÉRIE

INSIDE MAN

TOP 10 Top 9 em séries hoje

Um prisioneiro americano que está no corredor da morte e resolve mistérios para passar o tempo ajuda uma jornalista britânica a procurar uma amiga desaparecida.



Mais informações

**A16**

Lançamentos

