

Objetivos:

- I. Bootstrap;
- II. CDN;
- III. Componentes;
- IV. Layout;
- V. Spacing e sizing.

I. Bootstrap

Bootstrap é um framework front-end, com código-fonte aberto, para o desenvolvimento de componentes de interface usando HTML, CSS e JavaScript.

O Bootstrap foi originalmente desenvolvido pela equipe do Twitter, como um instrumento para incentivar a padronização no desenvolvimento de elementos de interfaces web, e se tornou um dos frameworks de front-end mais populares.

O framework combina CSS e JavaScript para estilizar os elementos HTML. Trata-se de uma ferramenta que proporciona interatividade, oferecendo componentes que facilitam a comunicação com o usuário, tais como, menus de navegação, formulários e tabelas. Porém, o principal objetivo é permitir a construção de páginas responsivas em navegadores e dispositivos móveis – uma interface é responsiva se ela consegue se adaptar às diferentes dimensões da tela.

O Bootstrap é formado por arquivos CSS e JavaScript que são responsáveis por atribuir características específicas aos elementos HTML da página. Para incluir o framework numa página é necessário incluir os arquivos:

- bootstrap.css: contém os estilos necessários para a estilização dos elementos. É necessário incluir o elemento `<link>` no `<head>` da página com o link para o arquivo CSS:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
rel="stylesheet">
```

- bootstrap.js: contém o código JavaScript responsável por aplicar, interativamente, estilos CSS nos elementos HTML para produzir o visual responsivo. É necessário incluir o elemento `<script>` no `<head>` da página com o link para o arquivo JS. No corpo do elemento `<script>` é o local onde podemos colocar código JavaScript no documento HTML:

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"></script>
```

Os estilos do Bootstrap são aplicados nos elementos HTML através de classes CSS. No exemplo a seguir os botões receberam os estilos através das classes `btn` (para aplicar o estilo de botão) e `btn-primary` ou `btn-secondary` (para aplicar o estilo de cores de botões).

Código do arquivo index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Exemplo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"></script>
  </head>
  <body>
    <button type="button" class="btn btn-primary">botão</button>
    <button type="button" class="btn btn-secondary">botão</button>
  </body>
</html>
```

Resultado no navegador

botão

botão

Observações:

- Bootstrap requer o uso de doctype HTML5 `<!DOCTYPE html>`. Sem ele os estilos podem ser renderizados incompletos;
- O Bootstrap foi desenvolvido primeiro para dispositivos móveis, uma estratégia na qual os estilos foram otimizados para dispositivos móveis e depois foram incrementados conforme necessário. Para garantir a renderização adequada e o zoom de toque para todos os dispositivos, adicione a marcação de visualização responsiva de viewport (janela de visualização) no `<head>`:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- Os estilos do Bootstrap utilizam, por padrão, `box-sizing: border-box`. Desta forma, a borda e padding não afetam as dimensões do elemento na tela.

Para mais detalhes <https://getbootstrap.com/docs/5.3/getting-started/introduction>.

II. CDN

Um CDN (Content Delivery Network ou Content Distribution Network ou Rede de Distribuição de Conteúdo) é um centro de distribuição de conteúdo na internet. Na programação Web, os CDN são usados para armazenar bibliotecas de códigos Javascript e CSS que podem ser consumidos por diversos sites no mundo – assim como o framework Bootstrap.

Uma biblioteca pode ser armazenada no **seu servidor** ou consumida a partir de um CDN de **terceiros**. Vantagens de usar bibliotecas de CDN:

- O navegador do usuário para carregar a sua página precisa fazer o download do código HTML, CSS, imagens e bibliotecas armazenadas no **seu servidor**, fazendo uso da **sua banda** de rede, ou seja,

aumentando os custos. Ao incluir bibliotecas de terceiros, o navegador do usuário fará o download dessas bibliotecas de servidores de terceiros, diminuindo assim o tráfego de banda no seu servidor;

- Sites distintos acessados pelo cliente podem utilizar as mesmas bibliotecas. Alguns navegadores podem fazer o cache dessas bibliotecas e não carregar elas novamente. Porém, para evitar carregar a mesma biblioteca duas vezes o navegador precisa confiar que as bibliotecas de código não sofreram alterações. Geralmente, os navegadores confiam nas versões de bibliotecas distribuídas por CDN.

O CDN de referência para a biblioteca Bootstrap é:

- CSS: <https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css>
- JS: <https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js>

Observações:

- Os arquivos na versão **min** (minificados) não contém comentários, caracteres de nova linha, espaços duplos e os nomes de variáveis são encurtados, entre outros recursos para diminuir o tamanho do arquivo a ser transmitido pela rede;
- A versão do Bootstrap no momento da redação deste texto é 5.3.0. É sempre bom usar a última versão para mais detalhes veja <https://getbootstrap.com/docs/5.3/getting-started/introduction/#cdn-links>.

III. Componentes

No Bootstrap os elementos HTML que recebem as classes CSS são chamados de componentes, pois além dos estilos CSS, eles podem ter códigos JavaScript para dar interatividade ou produzir efeitos visuais. O Bootstrap possui vários componentes, porém aqui será apresentado um para exemplificar como utilizar as classes CSS do Bootstrap nos elementos HTML.

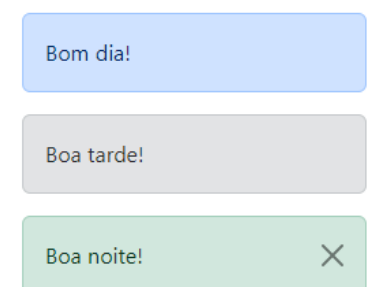
No exemplo a seguir foram criados três componentes Alert. Veja que os componentes foram construídos usando a marcação `<div>`, não é recomendado, mas poderia ter sido utilizado `<p>` ou `<h1>`.

O componente Alert é criado usando a combinação das classes `alert` (aplica o estilo de fonte e espaçamento) e `alert-primary` (aplica o estilo de cores). O atributo `role` fornece informações adicionais a serem utilizadas pelo código JavaScript da biblioteca Bootstrap.

Código do corpo do elemento `<body>` do arquivo index.html

```
<body>
  <div class="alert alert-primary" role="alert">
    Bom dia!
  </div>
  <div class="alert alert-secondary" role="alert">
    Boa tarde!
  </div>
  <div class="alert alert-success alert-dismissible fade show"
role="alert">
    Boa noite!
    <button type="button" class="btn-close" data-bs-dismiss="alert">
```

Resultado no navegador



```
aria-label="Close"></button>
</div>
</body>
```

No exemplo anterior foram apresentados três estilos de cores dos componentes de Alert. O Bootstrap utiliza uma paleta de estilos com 8 cores identificadas pelos seguintes sufixos nas classes:

- alert-**primary**
- alert-**secondary**
- alert-**success**
- alert-**danger**
- alert-**warning**
- alert-**info**
- alert-**light**
- alert-**dark**

Veja o componente Alert com o texto “Boa noite!”. Ele pode ser retirado da tela usando o botão dismiss (liberar, em inglês). O componente Alert para ser retirado da página precisa ter:

- O componente Alert precisa ter a classe **alert-dismissible**;
- O componente Alert precisa ter no corpo um elemento **<button>** com o atributo **data-bs-dismiss="alert"**. Ao clicar no botão é disparado o código JavaScript para retirar o componente Alert que possui o botão como ancestral;
- As classes **fade** e **show** são usadas no componente Alert para efeitos de animação ao esconder e exibir o elemento HTML.

Para mais detalhes sobre o componente Alert <https://getbootstrap.com/docs/5.3/components/alerts>.

IV. Layout

Para proporcionar um layout responsivo o Bootstrap adota o sistema de grades que define pontos de interrupção (breakpoints) nas seguintes larguras:

Breakpoint	Classe	Largura
Extra small		<576px
Small	col-sm	≥576px
Medium	col-md	≥768px
Large	col-lg	≥992px
Extra large	col-xl	≥1200px
Extra extra large	col-xxl	≥1400px

Para mais detalhes sobre os pontos de quebra <https://getbootstrap.com/docs/5.3/layout/breakpoints>.

Os contêineres são os elementos base do layout que respeitam os pontos de quebra. Para criar um container precisamos utilizar alguma dessas classes:

- Classe `container`: define uma largura máxima em cada ponto de interrupção responsivo. Para verificar, veja o primeiro `<div>` do exemplo a seguir. Redimensione lentamente a largura da janela do navegador e observe os pontos de quebra nas medidas definidas. O elemento ocupa 100% da largura até 576px e após este valor ele se redimensiona apenas nos pontos de quebra 768px, 992px, 1200px e 1400px;
- Classe `container-{breakpoint}`: considere como exemplo o segundo `<div>` do exemplo a seguir. O elemento ocupa 100% da largura até 768px e após este valor ele se redimensiona apenas nos pontos de quebra 992px, 1200px e 1400px;
- Classe `container-fluid`: containers fluídos ocupam 100% de largura do viewport em todos os pontos de interrupção. Veja como exemplo a última `<div>` no exemplo a seguir.

```
<body>
  <div class="container bg-primary">
    100% de largura até 576px - small breakpoint
  </div>
  <div class="container-md bg-secondary">
    100% de largura até 768px - medium breakpoint
  </div>
  <div class="container-lg bg-success">
    100% de largura até 992px - large breakpoint
  </div>
  <div class="container-xl bg-danger">
    100% de largura até 1200px - extra large breakpoint
  </div>
  <div class="container-xxl bg-warning">
    100% de largura até 1400px - extra extra large breakpoint
  </div>
  <div class="container-fluid bg-info">
    100% de largura em todas as larguras
  </div>
</body>
```

Contêineres

O sistema de grade divide o espaço horizontal em 12 células ou colunas. Para obter as células o sistema utiliza a divisão em contêineres (classe `container`), linhas (classe `row`) e colunas (classe `col`). Internamente, o sistema de grade utiliza CSS flexbox.

O exemplo a seguir produz o seguinte resultado. Veja que a soma das colunas precisa totalizar 12:

- Na 1ª e 2ª linha não foram especificadas a quantidade de colunas, então as células ocuparam a mesma quantidade de colunas;
- Na 3ª linha usou-se os sufixos `-3` e `-4` para especificar a quantidade de colunas que a célula deveria ocupar. A última célula ocupa o espaço restante até o total de 12 colunas;
- Na 4ª linha existem apenas 7 colunas, fazendo o restante ficar sem colunas;

- Se a soma de colunas ultrapassar 12 a próxima célula será posicionada na linha seguinte. Isso aconteceu na 5ª linha.

1/6 = 6 colunas				1/6 = 6 colunas				
1/3 = 4 colunas			1/3 = 4 colunas			1/3 = 4 colunas		
3 colunas		4 colunas			5 colunas - o que sobrou			
3 colunas		4 colunas						
8 colunas						3 colunas		
5 colunas								

Código do corpo do elemento `<body>` do arquivo index.html:

```
<body>
  <div class="container">
    <div class="row">
      <div class="col border">1/6 = 6 colunas</div>
      <div class="col border">1/6 = 6 colunas</div>
    </div>
    <div class="row">
      <div class="col border">1/3 = 4 colunas</div>
      <div class="col border">1/3 = 4 colunas</div>
      <div class="col border">1/3 = 4 colunas</div>
    </div>
    <div class="row">
      <div class="col-3 border">3 colunas</div>
      <div class="col-4 border">4 colunas</div>
      <div class="col border">5 colunas - o que sobrou</div>
    </div>
    <div class="row">
      <div class="col-3 border">3 colunas</div>
      <div class="col-4 border">4 colunas</div>
    </div>
    <div class="row">
      <div class="col-8 border">8 colunas</div>
      <div class="col-3 border">3 colunas</div>
      <div class="col-5 border">5 colunas</div>
    </div>
  </div>
</body>
```

Para mais detalhes sobre os contêineres <https://getbootstrap.com/docs/5.3/layout/containers>.

Grid - grade com pontos de quebra

O sistema de grade acomoda as colunas usando 6 pontos de quebra e cada um utiliza os seguintes sufixos nas classes de colunas (`col`):

- Extra small (`col-xs`): <575px
- Small (`col-sm`): >=576px

- Medium (col-md): ≥768px
- Large (col-lg): ≥992px
- Extra large (col-xl): ≥1200px
- Extra extra large (col-xxl): ≥1400px

O exemplo a seguir possui um contêiner com apenas uma linha (classe row). Porém o resultado muda em cada ponto de quebra:

```
<body>
  <div class="container">
    <div class="row">
      <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3 col-xl-2 border">um</div>
      <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3 col-xl-2 border">dois</div>
      <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3 col-xl-2 border">três</div>
      <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3 col-xl-2 border">quatro</div>
      <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3 col-xl-2 border">cinco</div>
      <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3 col-xl-2 border">seis</div>
    </div>
  </div>
</body>
```

Resultado com a largura da janela do navegador <576px. Observe que neste caso foi utilizada a classe col-xs-12, ou seja, cada elemento ocupa 12 colunas da grade:

um
dois
três
quatro
cinco
seis

Resultado com a largura da janela do navegador ≥576px e <768px. Observe que neste caso foi utilizada a classe col-sm-6, ou seja, cada elemento ocupa 6 colunas da grade:

um	dois
três	quatro
cinco	seis

Resultado com a largura da janela do navegador ≥768px e <992px. Observe que neste caso foi utilizada a classe col-md-4, ou seja, cada elemento ocupa 4 colunas da grade:

um	dois	três
quatro	cinco	seis

Resultado com a largura da janela do navegador ≥992px e <1200px. Observe que neste caso foi utilizada a classe col-lg-3, ou seja, cada elemento ocupa 3 colunas da grade:

um	dois	três	quatro
cinco	seis		

Resultado com a largura da janela do navegador $\geq 992\text{px}$ e $< 1200\text{px}$. Observe que neste caso foi utilizada a classe `col-xl-2`, ou seja, cada elemento ocupa 2 colunas da grade:

um	dois	três	quatro	cinco	seis
----	------	------	--------	-------	------

A quantidade de células por coluna pode ser definida na linha usando a classe `row-cols-*`. No exemplo a seguir cada coluna ocupará 6 células independentemente da largura da tela:

um	dois
três	quatro
cinco	seis

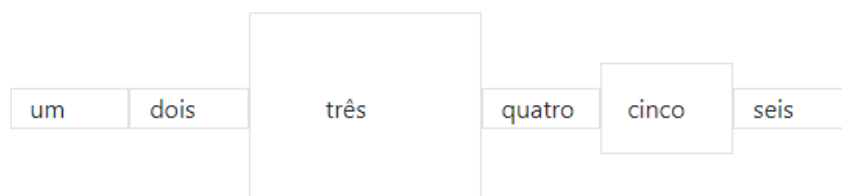
```
<body>
  <div class="container">
    <div class="row row-cols-2">
      <div class="col border">um</div>
      <div class="col border">dois</div>
      <div class="col border">três</div>
      <div class="col border">quatro</div>
      <div class="col border">cinco</div>
      <div class="col border">seis</div>
    </div>
  </div>
</body>
```

Para mais detalhes sobre a grade <https://getbootstrap.com/docs/5.3/layout/grid>.

Alinhamento vertical

O alinhamento dos componentes coluna (elementos HTML com a classe `col`) pode ser alterado utilizando classes Bootstrap que implementam as funcionalidades de alinhamento horizontal e vertical do flexbox CSS.

No exemplo a seguir a linha possui a classe `align-items-center` para alinhar os elementos filhos no centro verticalmente. Outros valores possíveis são: `align-items-start` e `align-items-end`.

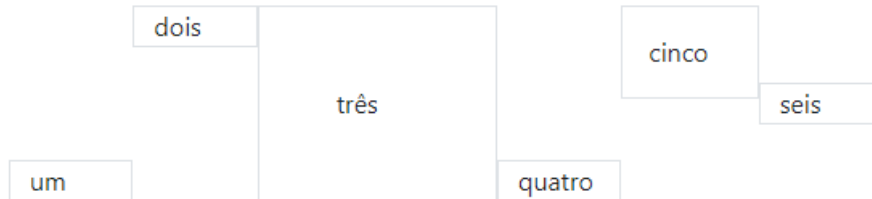


```
<body>
  <div class="container">
    <div class="row align-items-center">
      <div class="col border">um</div>
      <div class="col border">dois</div>
      <div class="col border p-5">três</div>
      <div class="col border">quatro</div>
    </div>
  </div>
```



```
<div class="col border p-3">cinco</div>
<div class="col border">seis</div>
</div>
</div>
</body>
```

Assim como nas classes CSS do flexbox, no Bootstrap podemos fazer o alinhamento vertical diretamente nos elementos filhos usando as classes `align-self-center`, `align-self-start` e `align-self-end`.

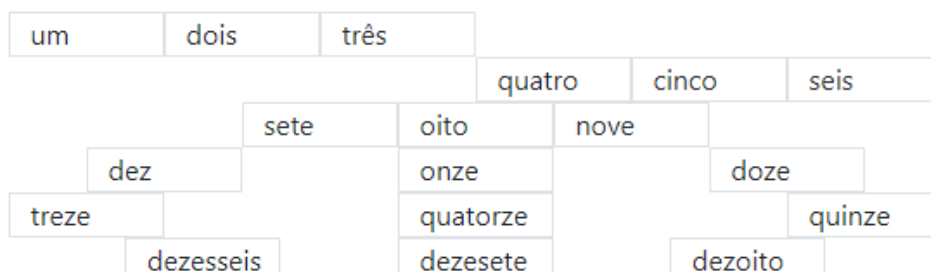


```
<body>
<div class="container">
  <div class="row">
    <div class="col border align-self-end">um</div>
    <div class="col border align-self-start">dois</div>
    <div class="col border p-5 align-self-center">três</div>
    <div class="col border align-self-end">quatro</div>
    <div class="col border p-3 align-self-start">cinco</div>
    <div class="col border align-self-center">seis</div>
  </div>
</div>
</body>
```

Alinhamento horizontal

As classes Bootstrap `justify-content-start`, `justify-content-end`, `justify-content-center`, `justify-content-around`, `justify-content-between` e `justify-content-evenly` são aplicadas no componente row e são usadas para alinhar as colunas.

Cada linha do exemplo a seguir faz uso de uma classe Bootstrap para alinhamento horizontal.



```
<body>
<div class="container">
  <div class="row justify-content-start">
    <div class="col-2 border">um</div>
    <div class="col-2 border">dois</div>
    <div class="col-2 border">três</div>
  </div>
</div>
```

```

</div>
<div class="row justify-content-end">
  <div class="col-2 border">quatro</div>
  <div class="col-2 border">cinco</div>
  <div class="col-2 border">seis</div>
</div>
<div class="row justify-content-center">
  <div class="col-2 border">sete</div>
  <div class="col-2 border">oito</div>
  <div class="col-2 border">nove</div>
</div>
<div class="row justify-content-around">
  <div class="col-2 border">dez</div>
  <div class="col-2 border">onze</div>
  <div class="col-2 border">doze</div>
</div>
<div class="row justify-content-between">
  <div class="col-2 border">treze</div>
  <div class="col-2 border">quatorze</div>
  <div class="col-2 border">quinze</div>
</div>
<div class="row justify-content-evenly">
  <div class="col-2 border">dezesesseis</div>
  <div class="col-2 border">dezesete</div>
  <div class="col-2 border">dezoito</div>
</div>
</div>
</body>

```

Offset column – deslocamento de coluna

A classe Bootstrap `offset-*`, especifica um deslocamento no ponto de início do elemento na tela. No exemplo a seguir, na 1ª linha a classe `offset-3` faz o 1º `<div>` começar na 4ª coluna, e na 2ª linha a classe `offset-2`, faz o 2º `<div>` ser deslocado duas colunas para direita.

	um	dois	três
quatro		cinco	seis

```

<body>
  <div class="container">
    <div class="row">
      <div class="col-2 border offset-3">um</div>
      <div class="col-2 border">dois</div>
      <div class="col-2 border">três</div>
    </div>
    <div class="row">
      <div class="col-3 border ">quatro</div>
      <div class="col-2 border offset-2">cinco</div>
      <div class="col-2 border">seis</div>
    </div>
  </div>
</body>

```

```
</div>
</div>
</body>
```

Para mais detalhes <https://getbootstrap.com/docs/5.3/layout/columns>.

V. Spacing e sizing

O Bootstrap possui classes para margin e padding (preenchimento). As classes são nomeadas usando o formato:

`{propriedade}{lado}-{tamanho}` neste caso considera o ponto de quebra xs (extra small);

`{propriedade}{lado}-{breakpoint}-{tamanho}` para os pontos de quebra (breakpoints) sm, md, lg, xl e xxl.

As propriedades são:

- `m` para margem;
- `p` para padding.

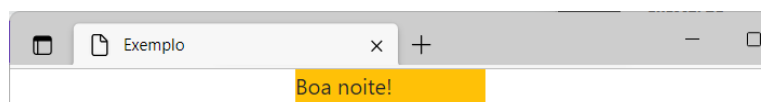
Os lados são:

- `t` para margin-top ou padding-top;
- `b` para margin-bottom ou padding-bottom;
- `s` (start) para margin-left ou padding-bottom;
- `e` (end) para margin-right ou padding-right;
- `x` para setar o lado esquerdo e direito da margem ou padding;
- `y` para setar o lado superior e inferior da margem e padding;
- “em branco” para a margem e padding nos 4 lados (top, right, bottom e left).

O Bootstrap adota 6 tamanhos de 0 a 5, onde 0 é menor e 5 é o maior. O valor `auto` seta a margem em todas as direções.

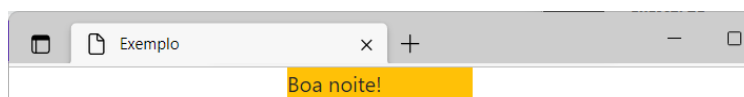
O exemplo a seguir faz uso da classe `mx-auto` para setar a margem em horizontal em `auto`. Desta forma, o elemento é centralizado horizontalmente.

```
<body>
  <div class="mx-auto p-2 w-25 bg-warning">
    Boa noite!
  </div>
</body>
```

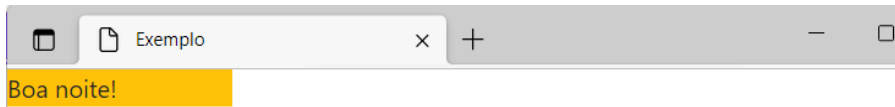


No exemplo a seguir a margem será:

- `auto` para larguras <576px;



- 0 para larguras de ≥ 576 a < 768 px;



- 2 para larguras de ≥ 768 a < 992 px;



- 5 para larguras ≥ 992 px.



```
<body>
  <div class="mx-auto mx-sm-0 mx-md-2 mx-lg-5 w-25 bg-warning">
    Boa noite!
  </div>
</body>
```

Para mais detalhes <https://getbootstrap.com/docs/5.3/utilities/spacing>.

O Bootstrap possui as classes **w** e **h** para definir a largura e altura do elemento, respectivamente. As classes podem receber os valores **25** (25% da largura ou altura), **50** (50%), **75** (75%), **100** (100%) e **auto**.

No exemplo a seguir os elementos recebem 25%, 50%, 75%, 100% e auto da largura do elemento pai, que por sua vez possui 300px de largura. Seria considerada a largura da janela se os elementos não estivessem dentro de um elemento com largura.



```
<body>
  <div style="max-width: 300px">
    <div class="w-25 bg-primary">Bom dia</div>
    <div class="w-50 bg-secondary">Bom dia</div>
    <div class="w-75 bg-success">Bom dia</div>
    <div class="w-100 bg-danger">Bom dia</div>
    <div class="w-auto bg-warning">Bom dia</div>
  </div>
</body>
```

Para mais detalhes <https://getbootstrap.com/docs/5.3/utilities/sizing>.

Exercícios

Utilize as imagens fornecidas para fazer os exercícios.

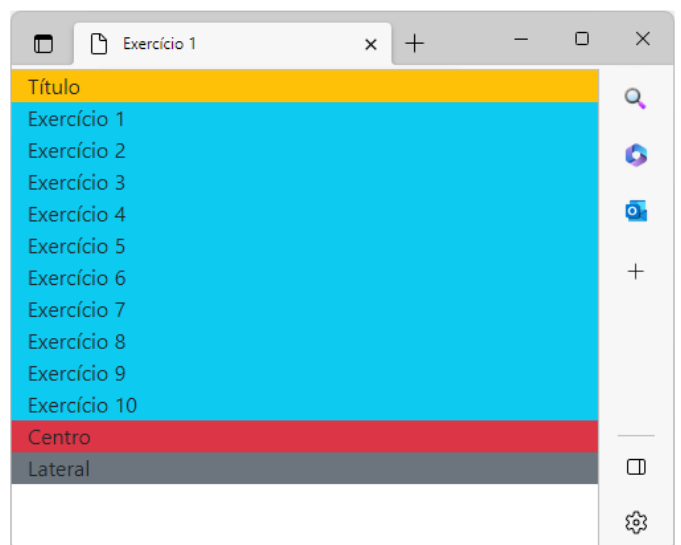
Exercício 1: Aplicar as classes Bootstrap no documento HTML para reproduzir o exemplo.

Requisitos do exercício:

- Não é permitido alterar as marcações HTML e incluir estilos CSS. Toda a formatação deverá ocorrer por classes Bootstrap;
- O visual deverá ser responsivo. A página deverá ter o seguinte layout com a largura do viewport $\geq 992\text{px}$:



- A página deverá ter o layout mostrado ao lado com a largura do viewport $< 992\text{px}$;
- Crie os blocos usando container, row e col.
- A página deverá ocupar toda a largura do viewport. Dica: use a classe container-fluid.



```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Exercício 1</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
```

```
    rel="stylesheet"
  />
  <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js">
  </script>
</head>
<body>
  <div class=" bg-warning">
    <div>
      <div>
        Título
      </div>
    </div>
    <div>
      <div class="bg-info">
        <div>Exercício 1</div>
        <div>Exercício 2</div>
        <div>Exercício 3</div>
        <div>Exercício 4</div>
        <div>Exercício 5</div>
        <div>Exercício 6</div>
        <div>Exercício 7</div>
        <div>Exercício 8</div>
        <div>Exercício 9</div>
        <div>Exercício 10</div>
      </div>
      <div class="bg-danger">
        Centro
      </div>
      <div class="bg-secondary">
        Lateral
      </div>
    </div>
  </div>
</body>
</html>
```

Exercício 2: O Exercício 2 é uma continuação do Exercício 1. Aplicar as classes Bootstrap no documento HTML para as imagens ter o visual proposto no exemplo.

Requisitos do exercício:

- Não é permitido alterar as marcações HTML e incluir estilos CSS. Toda a formatação deverá ocorrer por classes Bootstrap;
- As imagens deverão ser apresentadas na vertical com a largura do viewport $\geq 992\text{px}$:
 - As imagens centralizadas na horizontal. Dica: use a classe text-center no componente pai. A classe mx-auto só funciona se o componente pai tiver largura definida;

- As imagens centralizadas na vertical: Dica: use a classe my-auto (<https://getbootstrap.com/docs/5.3/utilities/spacing>);
- Padding de 3 unidades na vertical. Dica: use a classe de padding.



- As imagens deverão ser apresentadas na horizontal com a largura do viewport <992px:
 - As imagens centralizadas na horizontal;
 - As imagens centralizadas na vertical;
 - Padding de 2 unidades na vertical.



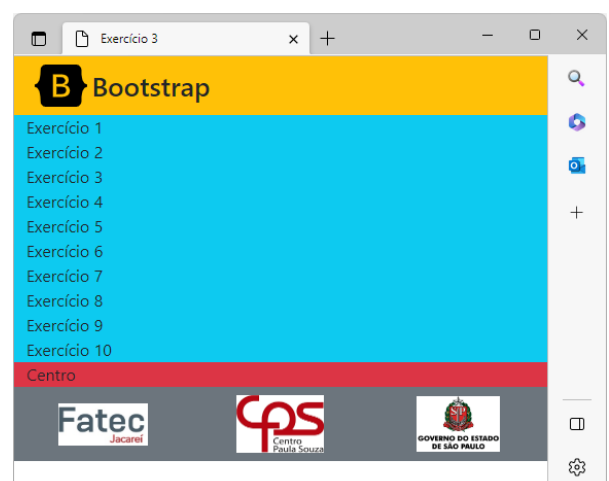
```
<body>
  <div class="bg-warning">
    <div>
      <div>
        Título
      </div>
    </div>
  </div>
  <div>
    <div class="bg-info">
      <div>Exercício 1</div>
      <div>Exercício 2</div>
      <div>Exercício 3</div>
      <div>Exercício 4</div>
      <div>Exercício 5</div>
    </div>
  </div>
  <div>
    <div class="bg-danger">
      Centro
    </div>
  </div>
  <div>
    <div class="bg-gray-800">
      <div>
        <div>Fatec</div>
        <div>CPS</div>
        <div>GOVERNO DO ESTADO DE SÃO PAULO</div>
      </div>
    </div>
  </div>
</body>
```

```
<div>Exercício 6</div>
<div>Exercício 7</div>
<div>Exercício 8</div>
<div>Exercício 9</div>
<div>Exercício 10</div>
</div>
<div class="bg-danger">
  Centro
</div>
<div class="bg-secondary">
  <div>
    <div>
      
    </div>
    <div>
      
    </div>
    <div>
      
    </div>
  </div>
</div>
</div>
</body>
```

Exercício 3: O Exercício 3 é uma continuação do Exercício 2. Aplicar as classes Bootstrap no documento HTML para o título ter o visual proposto no exemplo.

Requisitos do exercício:

- Não é permitido alterar as marcações HTML e incluir estilos CSS. Toda a formatação deverá ocorrer por classes Bootstrap;
- O elemento apresentado em amarelo precisa ter padding de 2 unidades;
- Use a classe `d-inline` (<https://getbootstrap.com/docs/5.3/utilities/display/>) para posicionar a imagem e título na mesma linha;
- Utilize a classe `h2` (<https://getbootstrap.com/docs/5.3/content/typography/#headings>) para formatar o título;
- Utilize a classe `align-middle` (<https://getbootstrap.com/docs/5.3/utilities/vertical-align/>) para alinhar o texto verticalmente.

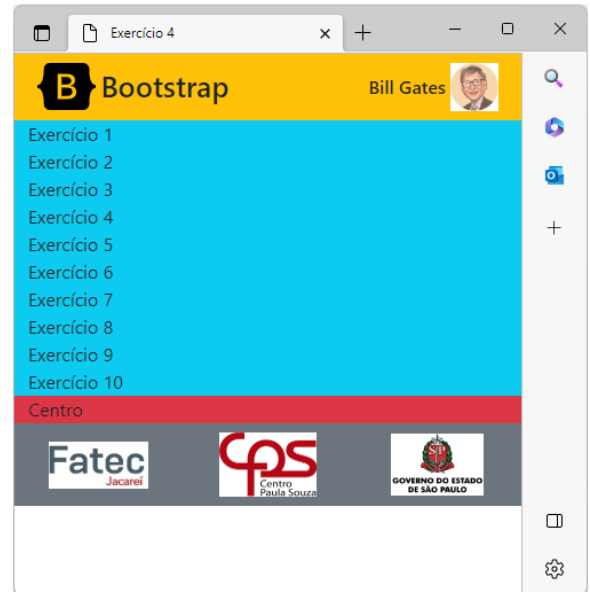



```
<body>
  <div class=" bg-warning">
    <div>
      <div>
        
        <div>Bootstrap</div>
      </div>
    </div>
    <div>
      <div class="bg-info">
        <div>Exercício 1</div>
        <div>Exercício 2</div>
        <div>Exercício 3</div>
        <div>Exercício 4</div>
        <div>Exercício 5</div>
        <div>Exercício 6</div>
        <div>Exercício 7</div>
        <div>Exercício 8</div>
        <div>Exercício 9</div>
        <div>Exercício 10</div>
      </div>
      <div class="bg-danger">
        Centro
      </div>
      <div class="bg-secondary">
        <div>
          <div>
            
          </div>
          <div>
            
          </div>
          <div>
            
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
```

Exercício 4: O Exercício 4 é uma continuação do Exercício 3. Aplicar as classes Bootstrap no documento HTML para o nome e avatar ter o visual proposto no exemplo.

Requisitos do exercício:

- Não é permitido alterar as marcações HTML e incluir estilos CSS. Toda a formatação deverá ocorrer por classes Bootstrap;
- O texto precisa estar alinhado verticalmente. Dica: use a classe align-middle;
- O texto precisa ter a classe h6;
- O texto e avatar precisam estar alinhados à direita.
Dica: use a classe text-end
(<https://getbootstrap.com/docs/5.3/utilities/text>).



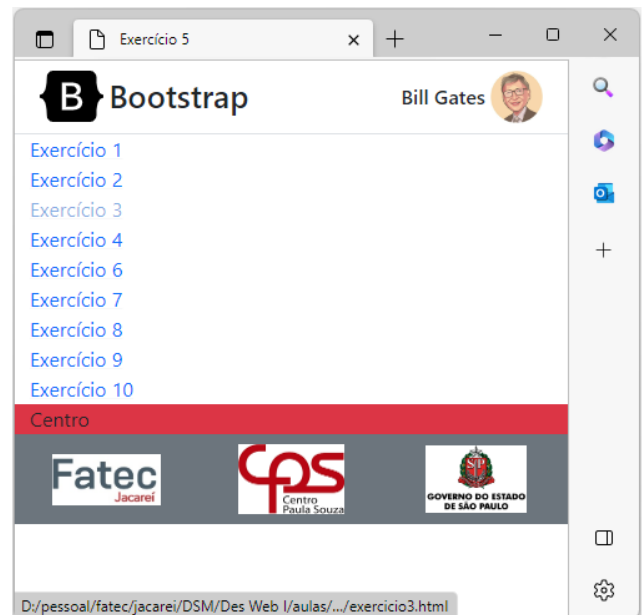
```
<body>
  <div class=" bg-warning">
    <div>
      <div>
        
      <div>Bootstrap</div>
    </div>
    <div>
      <div>Bill Gates</div>
      
    </div>
  </div>
  <div>
    <div class="bg-info">
      <div>Exercício 1</div>
      <div>Exercício 2</div>
      <div>Exercício 3</div>
      <div>Exercício 4</div>
      <div>Exercício 5</div>
      <div>Exercício 6</div>
      <div>Exercício 7</div>
      <div>Exercício 8</div>
      <div>Exercício 9</div>
      <div>Exercício 10</div>
    </div>
    <div class="bg-danger">
      Centro
    </div>
  </div>
```

```
<div class="bg-secondary">
  <div>
    <div>
      
    </div>
    <div>
      
    </div>
    <div>
      
    </div>
  </div>
</div>
</div>
</body>
```

Exercício 5: O Exercício 5 é uma continuação do Exercício 4. Aplicar as classes Bootstrap no documento HTML para ter os links para os exercícios.

Requisitos do exercício:

- Não é permitido alterar as marcações HTML e incluir estilos CSS. Toda a formatação deverá ocorrer por classes Bootstrap;
- A linha de título deverá ter borda inferior. Dica: use a classe `border-bottom` (<https://getbootstrap.com/docs/5.3/utilities/borders>);
- Os links não devem estar sublinhados e ao posicionar o cursor sobre o link, a opacidade/transparência deverá ser de 50%. No exemplo ao lado o curso está sobre Exercício 3. (<https://getbootstrap.com/docs/5.3/utilities/link>).



```
<body>
  <div class="bg-warning">
    <div>
      <div>
        
      </div>
    </div>
  </div>
```

```

    />
    <div>Bootstrap</div>
</div>
<div>
    <div>Bill Gates</div>
    
</div>
</div>
<div>
    <div class="bg-info">
        <div><a href="./exercicio1.html">Exercício 1</a></div>
        <div><a href="./exercicio2.html">Exercício 2</a></div>
        <div><a href="./exercicio3.html">Exercício 3</a></div>
        <div><a href="./exercicio4.html">Exercício 4</a></div>
        <div><a href="./exercicio6.html">Exercício 6</a></div>
        <div><a href="./exercicio7.html">Exercício 7</a></div>
        <div><a href="./exercicio8.html">Exercício 8</a></div>
        <div><a href="./exercicio9.html">Exercício 9</a></div>
        <div><a href="./exercicio10.html">Exercício 10</a></div>
    </div>
    <div class="bg-danger">
        Centro
    </div>
    <div class="bg-secondary">
        <div>
            <div>
                
            </div>
            <div>
                
            </div>
            <div>
                
            </div>
        </div>
    </div>
</div>
</body>

```

Exercício 6: O Exercício 6 é uma continuação do Exercício 5. Adicionar um formulário no elemento “centro” do documento HTML.

Requisitos do exercício:

- O formulário deverá ter os campos e botões mostrados ao lado. O campo Escolaridade deverá ter as opções: médio, técnico, superior (selecionado por padrão) e pós-graduação (<https://getbootstrap.com/docs/5.3/forms/overview>).

Exercício 7: O Exercício 7 é uma continuação do Exercício 6. Tornar o layout do formulário responsivo.

Requisitos do exercício:

Layout com a largura do viewport até <768px:

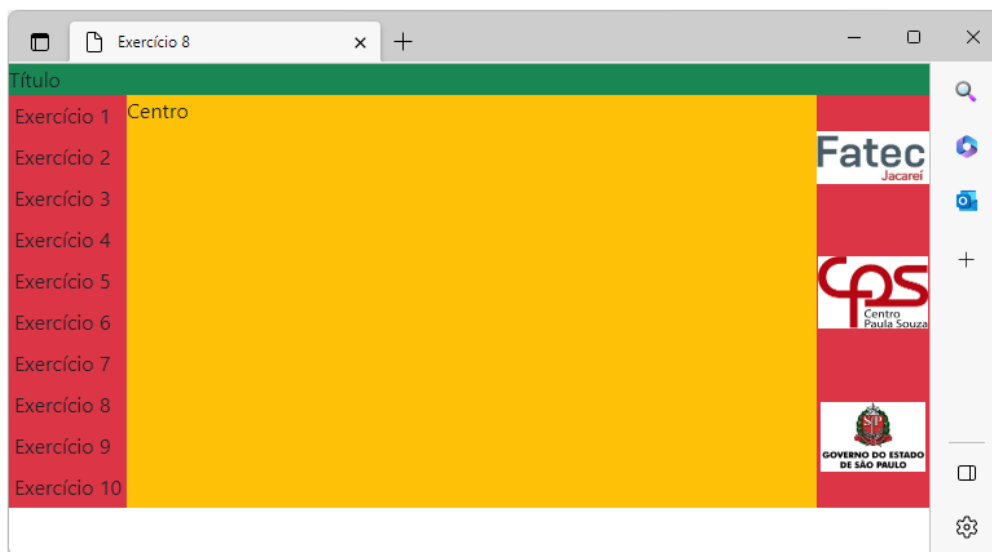
Layout com a largura do viewport até ≥768px:

Dica: use as classes `row` e `col` com pontos de interrupção (<https://getbootstrap.com/docs/5.3/forms/layout/#horizontal-form>).

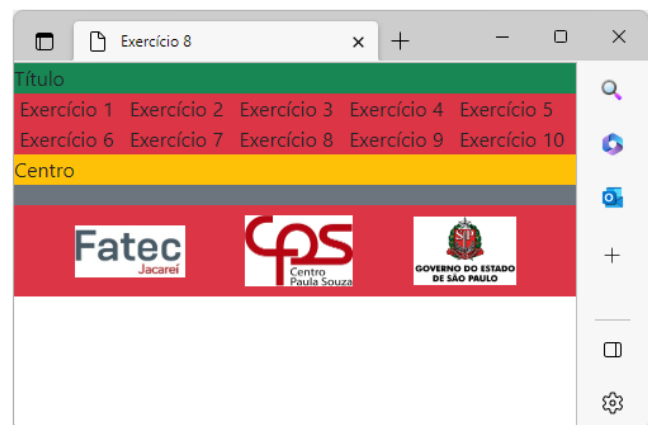
Exercício 8: Aplicar as classes Bootstrap de layout flex (<https://getbootstrap.com/docs/5.3/utilities/flex>) no documento HTML para reproduzir o exemplo.

Requisitos do exercício:

- Não é permitido alterar as marcações HTML, incluir estilos CSS e não poderá usar as classes `container`, `row` e `col` para organizar o layout. Toda a formatação deverá ocorrer por classes de layout flex do Bootstrap;
- O visual deverá ser responsivo. A página deverá ter o seguinte layout com a largura do viewport $\geq 992\text{px}$:



- A página deverá ter o layout mostrado ao lado com a largura do viewport $< 992\text{px}$.



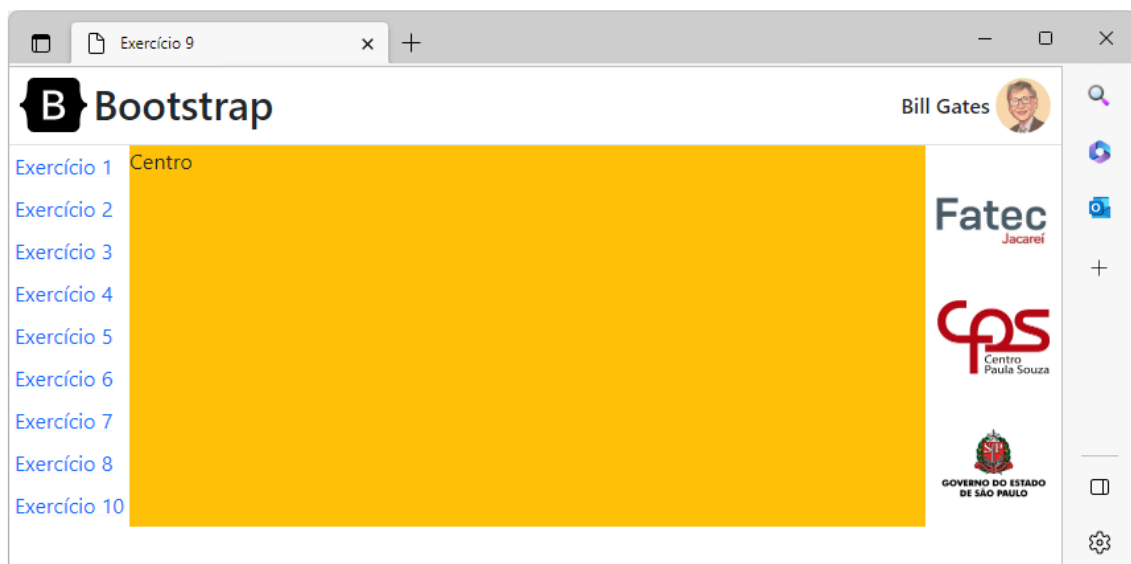
```
<body>
  <div class="bg-primary">
    <div class="bg-success">Título</div>
    <div class="bg-secondary">
      <div class="bg-danger">
        <div>Exercício 1</div>
        <div>Exercício 2</div>
        <div>Exercício 3</div>
```

```
<div>Exercício 4</div>
<div>Exercício 5</div>
<div>Exercício 6</div>
<div>Exercício 7</div>
<div>Exercício 8</div>
<div>Exercício 9</div>
<div>Exercício 10</div>
</div>
<div class="bg-warning">Centro</div>
<div class="bg-danger">
  <div>
    
  </div>
  <div>
    
  </div>
  <div>
    
  </div>
</div>
</div>
</div>
</body>
```

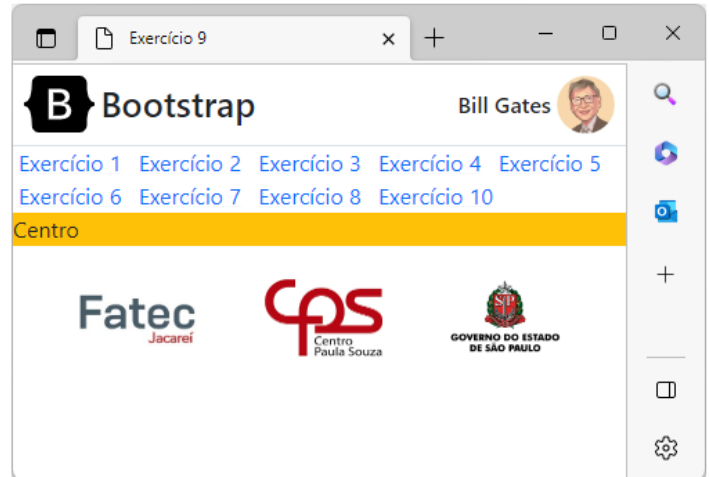
Exercício 9: Adicionar o título, avatar e links no documento HTML do Exercício 8.

Requisitos do exercício:

- O visual deverá ser responsivo. A página deverá ter o seguinte layout com a largura do viewport $\geq 992\text{px}$:



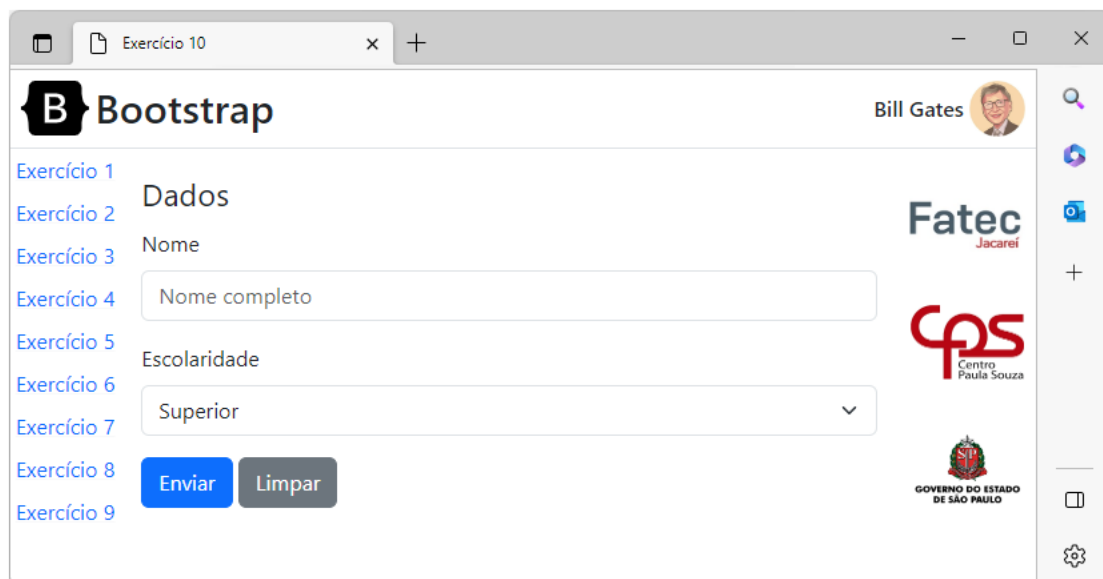
- A página deverá ter o layout mostrado ao lado com a largura do viewport <992px;
- Dicas:
 - Os links devem ser apresentados usando d-inline, para serem colocados na mesma linha;
 - Para evitar quebra de texto use text-nowrap nos links;
 - Use d-lg-flex para não ter a quebra de linha nas larguras ≥992px.



Exercício 10: Adicionar o formulário do Exercício 6 no elemento “centro” do Exercício 9.

Requisitos do exercício:

- O visual deverá ser responsivo. A página deverá ter o seguinte layout com a largura do viewport ≥992px:



- A página deverá ter o layout mostrado ao lado com a largura do viewport <992px;
- Dicas:
 - Use w-100 para que os campos ocupem 100% da largura do elemento pai.

