

# Programação Front End – AULA 12

Matheus Moresco

Engenharia de Software – 3 Período

2025/01

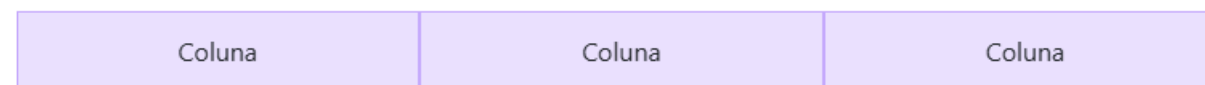
# Objetivo da Aula

- Ensinar como estruturar layouts responsivos com **Bootstrap Grid**
- Estilizar **formularios** de forma eficiente.

# O que é Grid Layout?

- Sistema de Grid do Bootstrap usa uma série de contêineres, linhas e colunas para layout e alinhamento de conteúdo. Ele é construído com flexbox e é totalmente responsivo. Abaixo está um exemplo e uma explicação detalhada de como o sistema de grade se reúne.

```
<div class="container text-center">  
  <div class="row">  
    <div class="col">  
      Column  
    </div>  
    <div class="col">  
      Column  
    </div>  
    <div class="col">  
      Column  
    </div>  
  </div>  
</div>
```



# Como funciona ?

Resumindo, veja como o sistema de grade se forma:

- Sistema de 12 colunas flexíveis
- Usa containers, rows e cols
- Classes responsivas: col-sm-, col-md-, col-lg-
- Permite criar layouts dinâmicos

# Opções de grade

- O sistema de grade do Bootstrap pode se adaptar a todos os seis breakpoints padrão e a quaisquer breakpoints que você personalizar. As seis camadas de grade padrão são as seguintes:
  - Extra pequeno (xs): <576px
  - Pequeno (sm): ≥576px
  - Médio (md): ≥768px
  - Grande (lg): ≥992px
  - Extra grande (xl): ≥1200px
  - Extra extra grande (xxl): ≥1400px

# Definindo o tamanho de colunas

- Para Grids que são as mesmas do menor dos dispositivos para o maior, use as classes `.col` `.col-*`. Especifique uma classe numerada quando precisar de uma coluna de tamanho específico; caso contrário, use `.col` para fazer o ajuste automático do tamanho da coluna.

```
<div class="container text-center">
  <div class="row">
    <div class="col">col</div>
    <div class="col">col</div>
    <div class="col">col</div>
    <div class="col">col</div>
  </div>
  <div class="row">
    <div class="col-8">col-8</div>
    <div class="col-4">col-4</div>
  </div>
</div>
```

coluna	coluna	coluna	coluna
col-8			col-4

## Definindo o tamanho de colunas

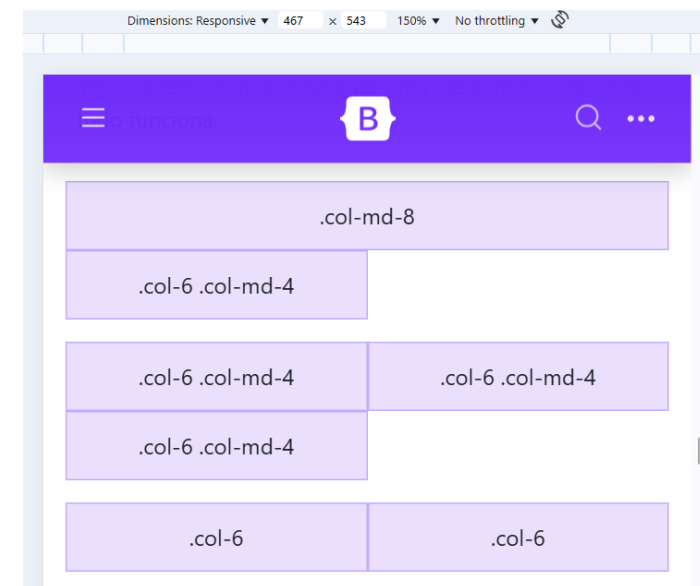
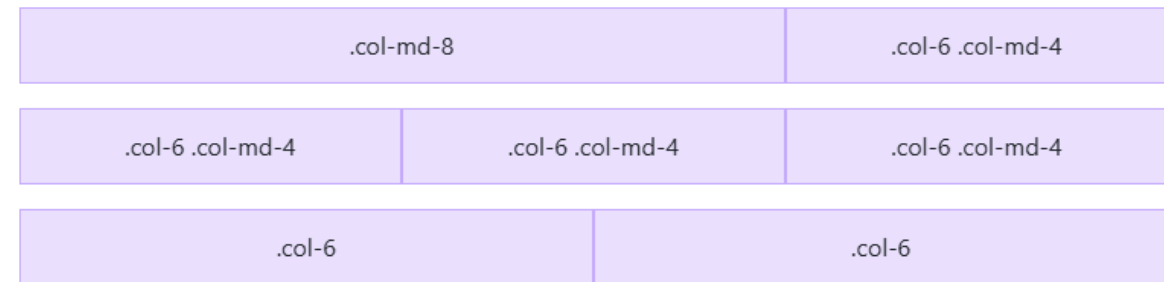
- Se quiser usar mais de um tamanho de display, use uma combinação de classes diferentes para cada tamanho, conforme necessário. Veja o exemplo a seguir para ter uma ideia melhor de como tudo funciona.

# Definindo o tamanho de colunas

```
<div class="container text-center">
  <!-- Stack the columns on mobile by making one full-width and the other half-width -->
  <div class="row">
    <div class="col-md-8">.col-md-8</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
  <div class="row">
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns are always 50% wide, on mobile and desktop -->
  <div class="row">
    <div class="col-6">.col-6</div>
    <div class="col-6">.col-6</div>
  </div>
</div>
```





# Quebras de linha automáticas

- Use as classes responsivas `.row-cols-*` para definir rapidamente o número de colunas que melhor renderizam seu conteúdo e layout. Com `.row-cols-auto` você pode dar às colunas sua largura natural.
- Use essas classes de linhas e colunas para criar rapidamente layouts de grade básicos ou para controlar seus layouts.

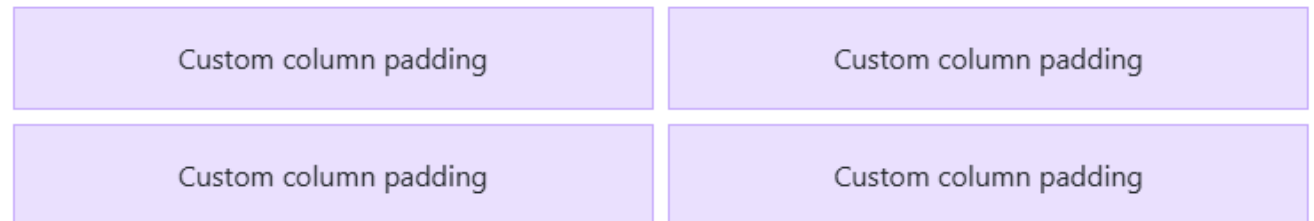
```
<div class="container text-center">  
  <div class="row row-cols-3">  
    <div class="col">Column</div>  
    <div class="col">Column</div>  
    <div class="col">Column</div>  
    <div class="col">Column</div>  
  </div>  
</div>
```

Coluna	Coluna	Coluna
Coluna		

# Espaçamentos com .g- (gutters)

- O Bootstrap oferece a classe .g- para definir o espaçamento entre as colunas e linhas dentro do grid.
- Sintaxe: g-{tamanho} → Onde {tamanho} varia de 0 a 5.

```
<div class="container text-center">
  <div class="row g-2">
    <div class="col-6">
      <div class="p-3">Custom column padding</div>
    </div>
    <div class="col-6">
      <div class="p-3">Custom column padding</div>
    </div>
    <div class="col-6">
      <div class="p-3">Custom column padding</div>
    </div>
    <div class="col-6">
      <div class="p-3">Custom column padding</div>
    </div>
  </div>
</div>
```



# Alinhamento Vertical

- A classe `.align-items-` controla o alinhamento vertical dos elementos dentro de uma row.
- Opções:
  - `.align-items-start` → Alinha os itens ao topo
  - `.align-items-center` → Alinha os itens no centro
  - `.align-items-end` → Alinha os itens na parte inferior

```
<div class="container">  
  <div class="row align-items-center" style="height: 200px;">  
    <div class="col-md-4 bg-danger text-white p-3">Coluna 1</div>  
    <div class="col-md-4 bg-warning text-white p-3">Coluna 2</div>  
    <div class="col-md-4 bg-info text-white p-3">Coluna 3</div>  
  </div>  
</div>
```

# Alinhamento Horizontal

- A classe `.justify-content-` controla o alinhamento horizontal dos itens dentro de uma row. Opções:
  - `.justify-content-start` → Alinha os itens à esquerda (padrão).
  - `.justify-content-center` → Centraliza os itens na linha.
  - `.justify-content-end` → Alinha os itens à direita.
  - `.justify-content-between` → Distribui os itens igualmente com espaços entre eles.
  - `.justify-content-around` → Distribui os itens com espaços iguais antes e depois.
  - `.justify-content-evenly` → Distribui os itens com espaços iguais entre eles.

```
<div class="container">
  <div class="row justify-content-around">
    <div class="col-md-3 bg-primary text-white p-3">Item 1</div>
    <div class="col-md-3 bg-secondary text-white p-3">Item 2</div>
    <div class="col-md-3 bg-success text-white p-3">Item 3</div>
  </div>
</div>
```

# Exemplo Prático

- Explorar o uso dos Grids Layouts.

# Formulários com Bootstrap

- O Bootstrap fornece também uma estilização própria pra formulários no HTML.
- Elementos principais:
  - Campos de entrada (.form-control)
  - Checkbox e radio buttons (.form-check)
  - Select (.form-select)
  - Grupos de entrada (.input-group)

# Formulários com Bootstrap

- Para criar um formulário simples, utilizamos a classe `.form-control` para estilizar os campos de entrada.
  - `.form-control` → Aplica um estilo padronizado aos inputs.
  - `.form-label` → Adiciona estilos adequados aos rótulos.

```
<div class="container mt-3">
  <form>
    <div class="mb-3">
      <label for="nome" class="form-label">
        Nome
      </label>
      <input
        type="text"
        class="form-control"
        id="nome"
        placeholder="Digite seu nome">
    </div>
    <div class="mb-3">
      <label for="email" class="form-label">
        Email
      </label>
      <input
        type="email"
        class="form-control"
        id="email"
        placeholder="Digite seu email">
    </div>
    <button type="submit" class="btn btn-primary">
      Enviar
    </button>
  </form>
</div>
```

# Formulários - Campos Específicos

- Além do input, temos campos com estilizações independentes, que podemos configurar de outras maneiras, são eles:
  - checkboxes
  - radios
  - selects
  - textarea



# Formulários - Checkbox

Caixas de seleção e rádios padrão do navegador são substituídos com a ajuda de `.form-check`, uma série de classes para ambos os tipos de entrada que melhora o layout e o comportamento de seus elementos HTML, que fornecem maior personalização e consistência entre navegadores.

```
<div class="form-check">
  <input class="form-check-input" type="checkbox" value="" id="flexCheckDefault">
  <label class="form-check-label" for="flexCheckDefault">
    Default checkbox
  </label>
</div>
<div class="form-check">
  <input class="form-check-input" type="checkbox" value="" id="flexCheckChecked" checked>
  <label class="form-check-label" for="flexCheckChecked">
    Checked checkbox
  </label>
</div>
```

- ☐ Caixa de seleção padrão
- ☒ Caixa de seleção marcada

# Formulários - Radios

- Os rádios também usam a classe form-check para determinar sua estilização, porém se diferenciam do checkbox pelo tipo do input.

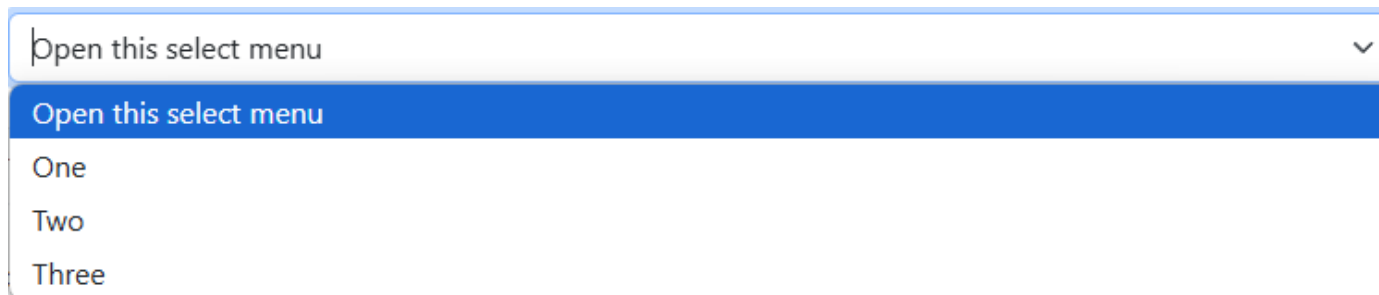
```
<div class="form-check">
  <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault1">
  <label class="form-check-label" for="flexRadioDefault1">
    Default radio
  </label>
</div>
<div class="form-check">
  <input class="form-check-input" type="radio" name="flexRadioDefault" id="flexRadioDefault2" checked="">
  <label class="form-check-label" for="flexRadioDefault2">
    Default checked radio
  </label>
</div>
```

- ☐ Rádio padrão
- ☒ Rádio marcado padrão

## Formulários - Select

- Menus personalizados <select> precisam apenas da classe personalizada .form-select para acionar os estilos personalizados. Estilos personalizados são limitados à <select>aparência inicial do e não podem modificar o <option> devido a limitações do navegador.

```
<select class="form-select" aria-label="Default select example">  
  <option selected>Open this select menu</option>  
  <option value="1">One</option>  
  <option value="2">Two</option>  
  <option value="3">Three</option>  
</select>
```



Open this select menu

Open this select menu

One

Two

Three

# Inputs com Ícones e Botões (.input-group)

- Podemos adicionar ícones e botões dentro dos inputs usando .input-group.
- Coloque um add-on ou botão em cada lado de uma entrada. Você também pode colocar um em ambos os lados de uma entrada. Lembre-se de colocar <label> fora do grupo de entrada.

```
<div class="input-group flex-nowrap">  
  <span class="input-group-text" id="addon-wrapping">@</span>  
  <input type="text" class="form-control" placeholder="Username" aria-label="Username" aria-describedby="addon-wrapping">  
</div>
```



@ Username

# Layout Responsivo com Grid

- Podemos criar formulários de com inputs de tamanhos personalizados usando colunas .row e .col-.

```
<form class="row g-3">
  <div class="col-md-6">
    <label class="form-label">Primeiro Nome</label>
    <input type="text" class="form-control">
  </div>
  <div class="col-md-6">
    <label class="form-label">Sobrenome</label>
    <input type="text" class="form-control">
  </div>
  <div class="col-md-12">
    <label class="form-label">Email</label>
    <input type="email" class="form-control">
  </div>
  <div class="col-12">
    <button type="submit" class="btn btn-success">
      Cadastrar
    </button>
  </div>
</form>
```

Primeiro Nome

Sobrenome

Email

Cadastrar

# Validação de Formulários

- Podemos utilizar classes de validação (.was-validated) para destacar campos inválidos.

```
<form class="was-validated">
  <div>
    <label class="form-label">Nome</label>
    <input type="text" class="form-control" required>
    <div class="invalid-feedback">Por favor, preencha este campo.</div>
  </div>
  <div>
    <label class="form-label">Email</label>
    <input type="email" class="form-control" required>
    <div class="invalid-feedback">Informe um email válido.</div>
  </div>
  <button type="submit" class="btn btn-primary">Enviar</button>
</form>
```

Nome

Por favor, preencha este campo.

Email

Informe um email válido.

Enviar

## Exemplo Prático

- Crie um formulário de cadastro de usuário contendo:
  - Nome, Email e Senha.
  - Select para escolher o tipo de usuário.
  - Checkbox para aceitar os termos.
  - Botão estilizado para envio.

## Exercício

1. Crie um formulário de cadastro para receber as informações de uma pessoa que está tentando acessar seu Currículo. Use Bootstrap para criar o formulário.
2. O formulário deve conter os campos dos seguinte tipos:
  - a) Nome e Sobrenome- text
  - b) E-mail - email
  - c) Empresa - text
  - d) Motivo da visualização – select (Quero contratar, curiosidade, saber da sua vida, outro)
  - e) Gostou do currículo? – radio
  - f) Sugestões – textarea
  - g) Fique a vontade para inserir mais campos
3. Use o Grid Layout para ajustar os campos do formulário