



# Programação para Internet

---

Módulo 3 – Gestão da Informação

Introdução a Design Responsivo

Introdução ao Framework Bootstrap

Prof. Dr. Daniel A. Furtado - FACOM/UFU

Conteúdo protegido por direito autoral, nos termos da Lei nº 9 610/98

A cópia, reprodução ou apropriação deste material, total ou parcialmente, é proibida pelo autor

# Conteúdo da Aula

- Introdução ao Design Responsivo, viewport e media queries
- Introdução ao Framework Bootstrap
- Sistema de Grade do Bootstrap
- Formulários com responsividade
- Outros recursos comuns

# Design Responsivo

- Baseado na ideia de que o website deve ter a capacidade de se ajustar para permitir uma boa exibição em todos os dispositivos, seja um desktop, um notebook, um tablet ou um smartphone
- Evita a necessidade de ter diferentes versões do website para diferentes dispositivos e tamanhos de tela
- Pode envolver a utilização de vários recursos como a meta tag viewport, media queries, unidades relativas, flexbox, grid etc.

# Conceito de Viewport

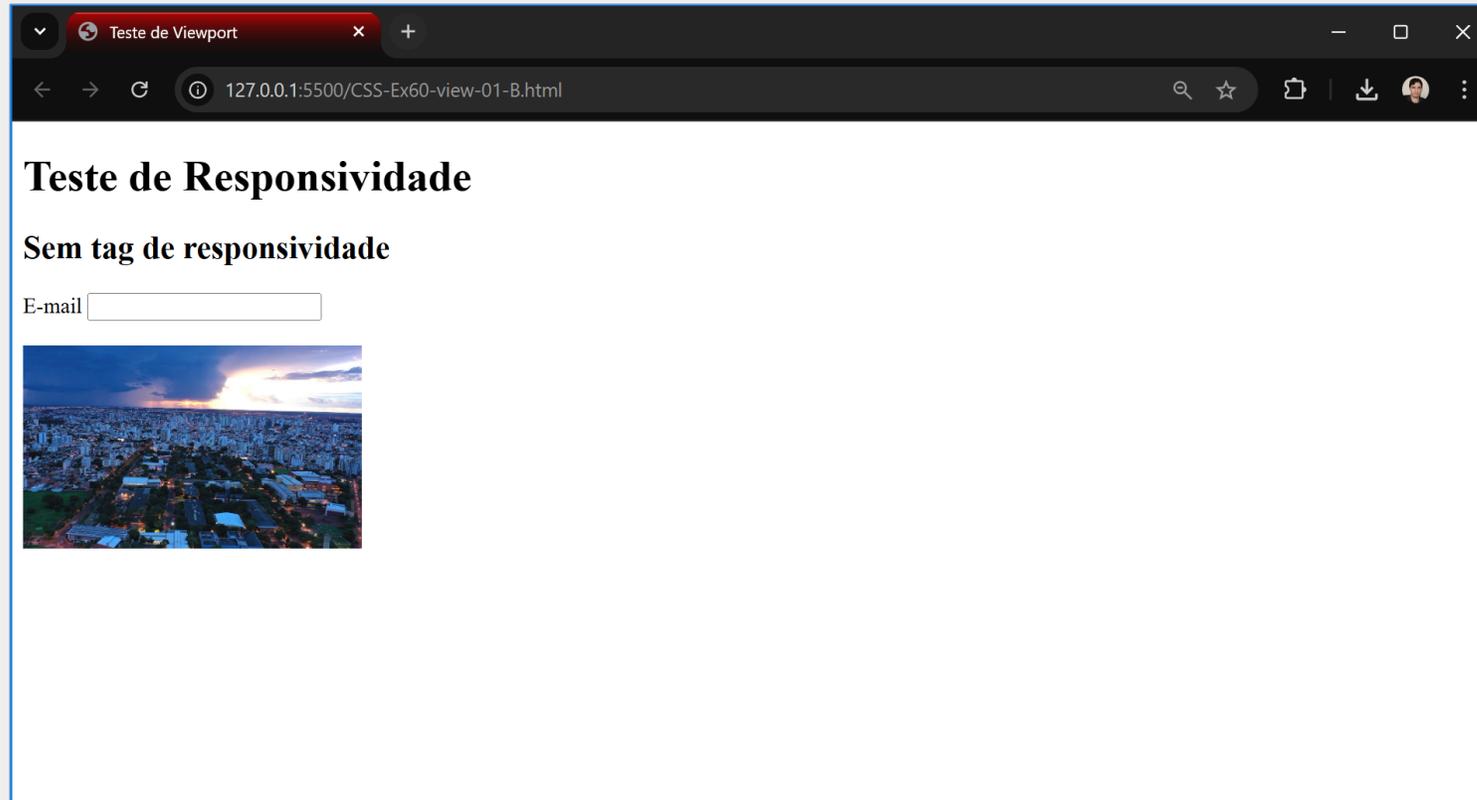
- **Viewport** é a área visível de renderização da página no navegador
- Na viewport, o dimensionamento (largura e altura) é tratado utilizando o conceito de **pixel CSS**
- **1 pixel CSS** de largura na viewport **nem sempre** corresponde a **1 pixel físico** da tela do dispositivo, pois pode ser levado em conta um fator de escala conhecido como **pixel ratio**
- A ideia do **pixel ratio** é permitir que dispositivos com alta densidade de pixels, como smartphones, tenham os elementos da página ampliados
- Para ativar o fator de escala é necessário utilizar a tag de responsividade:  

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

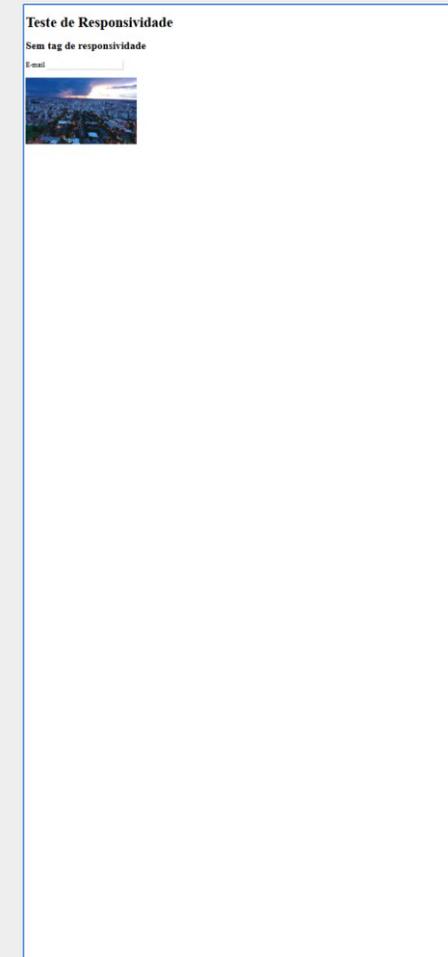
# Página de Exemplo sem Tag de Responsividade

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Teste de Viewport</title>
</head>
<body>
  <h1>Teste de Responsividade</h1>
  <h2>Sem tag de responsividade</h2>
  <label for="email">E-mail</label>
  <input type="email" id="email"><br><br>
  
</body>
</html>
```

# Visualização da Página sem Tag de Responsividade



Visualização da página em computador desktop de 14", resolução 1920x1080

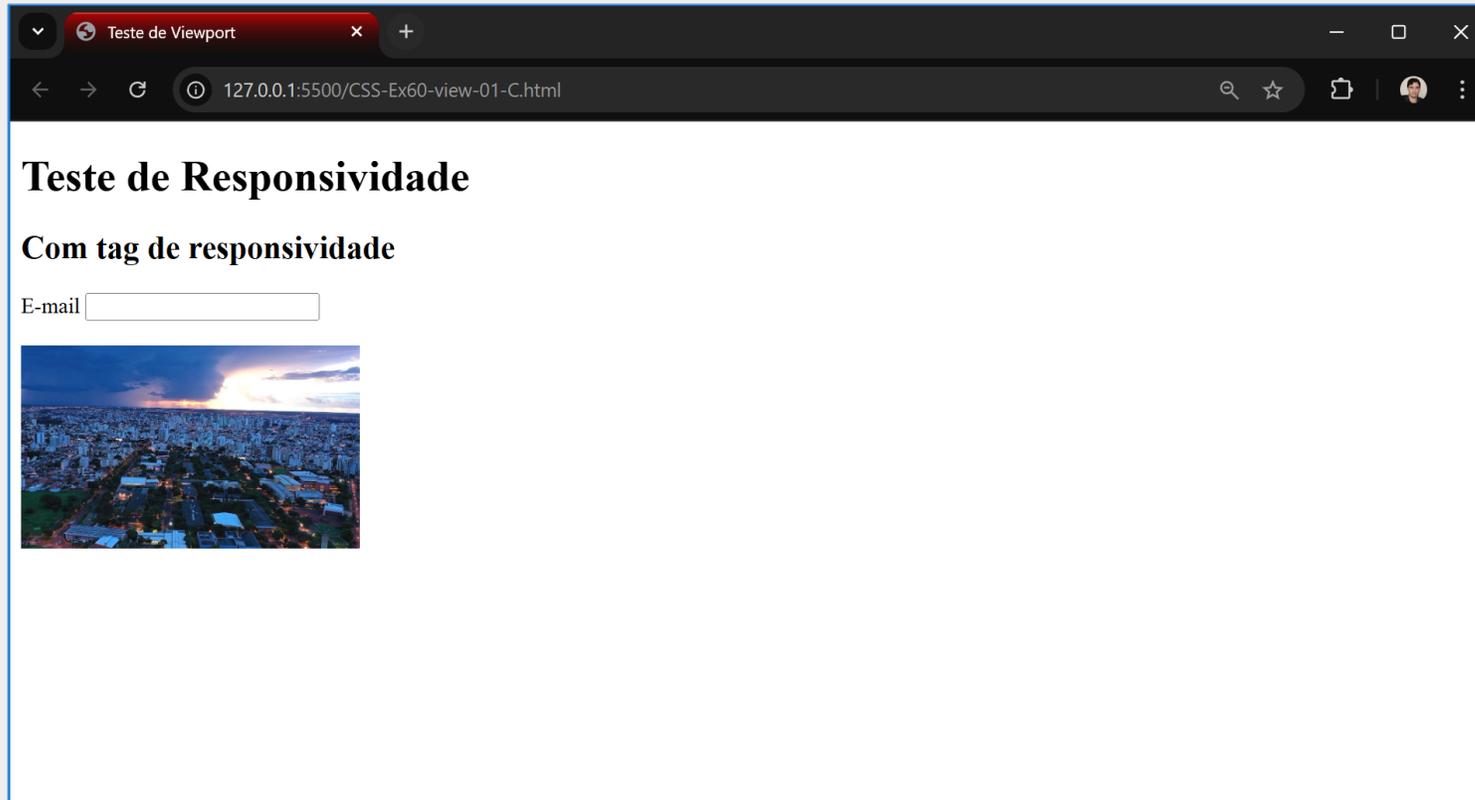


Visualização da página no Galaxy S20, resolução 1440x3200

# Página de Exemplo com Tag de Responsividade

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <title>Teste de Viewport</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
  <h1>Teste de Responsividade</h1>
  <h2>Sem tag de responsividade</h2>
  <label for="email">E-mail</label>
  <input type="email" id="email"><br><br>
  
</body>
</html>
```

# Visualização da Página com Tag de Responsividade



Visualização da página em computador desktop de 14", resolução 1920x1080

## Teste de Responsividade

Com tag de responsividade

E-mail



Visualização da página no Galaxy S20, resolução 1440x3200

# Tag de Responsividade e Pixel Ratio

- Repare que ao utilizar a tag de responsividade (exemplo anterior), o pixel ratio do dispositivo é levado em conta quando a página é exibida em dispositivos móveis
- No caso do Galaxy S20, o pixel ratio é igual a 4, o que significa que os elementos da página, de uma forma geral, são ampliados em 4x
- Por exemplo, ao definir `margin-left: 10px`, estamos definindo a margem esquerda para **10 pixels CSS**, mas a margem ocupará efetivamente a largura de  $10 \times 4 = 40$  pixels da resolução horizontal do Galaxy S20

# Quanto maior a densidade de pixels, maior o pixel ratio

Aparelho	Resolução da Tela (pixels)	Densidade de Pixels da Tela	Pixel Ratio Padrão	Res. da Viewport (pixels CSS)
Galaxy S20 6,2"	1440 x 3200	563	4	360 x 800
iPhone 14 Pro 6,1"	1179 x 2556	460	3	393 x 852
LG K10 5,3"	720 x 1280	277	2	360 x 640

De uma forma geral, os navegadores calculam o pixel ratio do dispositivo como uma aproximação da equação: *densidade de pixels da tela / 150*

**Ex:** Galaxy S20:  $563 / 150 = 3,753$  → pixel ratio no Google Chrome: 4

**Ex:** iPhone 12:  $460 / 150 = 3,06$  → pixel ratio no Google Chrome: 3

**Ex:** Google Pixel 5:  $432 / 150 = 2,8$  → pixel ratio no Google Chrome: 2,8

**OBS:** no código JavaScript é possível utilizar a propriedade `window.devicePixelRatio` para obter o pixel ratio do dispositivo que está acessando a página

# Simulação de Dispositivo Móvel no Navegador

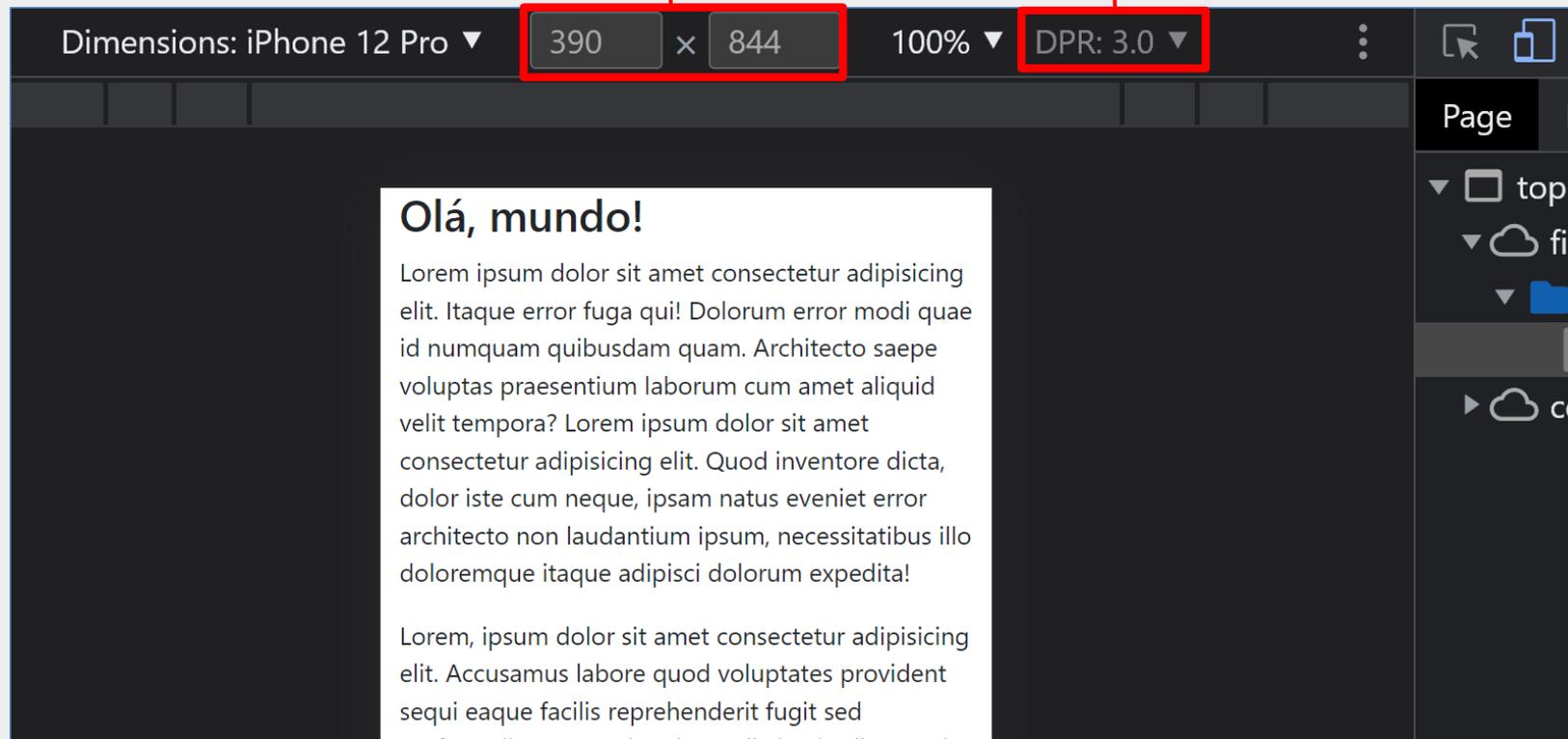
- Nos principais navegadores é possível simular a tela de dispositivos móveis
- No Google Chrome em um desktop, tecle **F12**, clique em  e depois selecione o aparelho desejado;
- Para visualizar o pixel ratio do dispositivo simulado, clique nos três pontinhos  e selecione **Add device pixel ratio**



# Simulação do Dispositivo Móvel no Navegador

Resolução da **viewport** em **pixels CSS**

**Pixel ratio** do dispositivo  
(*device pixel ratio* – DPR)



# Resumo sobre a Meta Tag Viewport

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

- Viabiliza a responsividade de acordo com o dispositivo e sua tela
- Faz com que o **pixel ratio** do dispositivo móvel seja considerado quando a página for acessada pelo dispositivo
- Portanto, o dimensionamento dos elementos passa a considerar a **densidade** de pixels da tela do dispositivo móvel
- **Resultado:** página melhor escalonada em dispositivos com alta resolução e tela pequena

# Observação sobre a Meta Tag Viewport

- Utilizar a meta tag viewport não faz com que a página se torne totalmente responsiva
- A meta tag viewport é apenas o **primeiro passo** para o design responsivo
- Normalmente ela é utilizada em conjunto com outros recursos como:
  - Media queries
  - Unidades relativas (como %)
  - Módulo Flexbox da CSS, Grid etc.

# Introdução à Media Queries

**screen** - para dispositivos com tela  
**print** - para impressão (ex. modo Ctrl-P)  
**all** - para todos os dispositivos (default)

**MediaType** é opcional  
Se omitido, será considerado **all**

**MediaCondition** é a condição a ser satisfeita para que o código CSS dentro das chaves seja considerado

```
@media MediaType AND MediaCondition {  
    /* Código CSS */  
}
```

(min-width: 400px)  
(max-width: 900px)  
(min-width: 400px) and (max-width: 900px)  
(400px <= width <= 900px)  
(orientation: portrait)  
(orientation: landscape)

Exemplos de **media condition**

min-width, max-width e orientation são **media features**

# Introdução à Media Queries

```
<style>
  body {
    width: 60%;
    padding: 2% 0;
    margin: 0 auto;
  }
  @media (max-width: 480px) {
    body {
      width: 95%;
    }
  }
</style>
```

Neste exemplo, o corpo da página aparecerá centralizado e com largura de 60% em dispositivos com telas largas.

Porém, em dispositivos com largura de viewport menor ou igual a 480px (smartphones), a segunda regra também será utilizada, permitindo que a página se expanda e ocupe 95% da largura.

max-width é uma *media feature*

# Introdução à Media Queries

```
body {  
  width: 60%;  
  padding: 2% 0;  
  margin: 0 auto;  
  line-height: 2.0;  
}  
  
@media print and (orientation: portrait) {  
  body {  
    line-height: 1.1;  
    font-size: 10pt;  
  }  
}
```

Neste exemplo, o espaçamento entre linhas e o tamanho da fonte serão reduzidos quando o documento estiver em modo de impressão na orientação **retrato** (Ctrl-P)

# Introdução à Media Queries

```
<style>
  body {
    width: 60%;
    padding: 2% 0;
    margin: 0 auto;
  }
  @media (max-width: 480px) {
    body {
      width: 100%;
    }
    header {
      display: none;
    }
  }
</style>
```

Neste exemplo, o corpo da página aparecerá centralizado e com largura de 60% em dispositivos com telas largas.

Porém em smartphones, na vertical, a página se expandirá para ocupará 100% da largura e o cabeçalho (header) não será apresentado.

# Introdução ao Framework Bootstrap

- Bootstrap é um framework de front-end muito popular
- É gratuito, de código aberto, com foco na criação de páginas responsivas
- Utiliza HTML, CSS e JavaScript
- Disponibiliza uma coleção de classes CSS
  - Melhoria de layouts, botões, formulários, etc.
- Disponibiliza uma coleção de ícones e templates
- Documentação: <https://getbootstrap.com/docs>

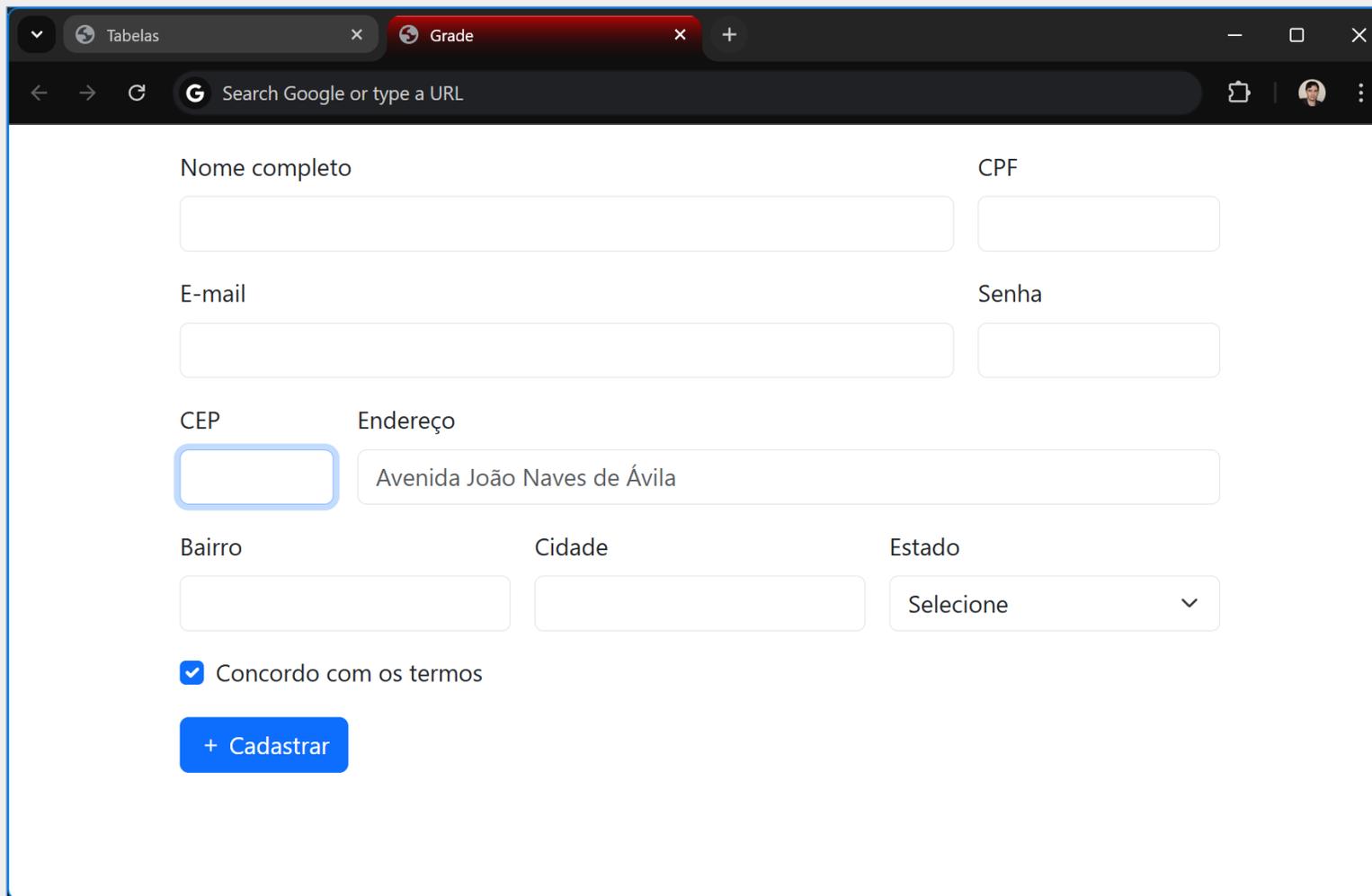
# Exemplo de Tabela Estilizada com Bootstrap



The image shows a browser window with a single tab titled 'Tabelas'. The address bar contains the text 'Search Google or type a URL'. Below the browser interface, a table titled 'Resultado dos Testes' is displayed. The table has four columns: 'Nome', 'Acertos', 'Tempo', and 'Nota'. The data rows are: Fulano (23, 00:30:58, 3.80), Beltrano (22, 00:24:08, 3.66), Ciclano (22, 00:26:14, 3.60), and Deltrano (22, 00:27:17, 3.66). The table is styled with a light gray background and thin borders.

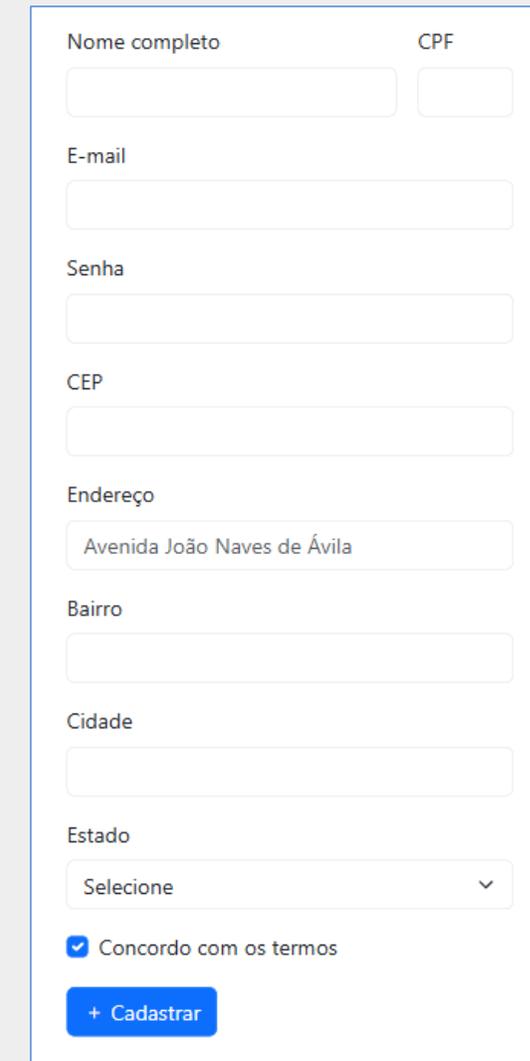
Nome	Acertos	Tempo	Nota
Fulano	23	00:30:58	3.80
Beltrano	22	00:24:08	3.66
Ciclano	22	00:26:14	3.60
Deltrano	22	00:27:17	3.66

# Exemplo de Formulário Responsivo Estilizado com Bootstrap



A screenshot of a web browser window showing a registration form. The browser has two tabs: 'Tabelas' and 'Grade'. The address bar contains 'Search Google or type a URL'. The form is displayed in a desktop layout with two columns. The fields are: 'Nome completo' and 'CPF' (top row); 'E-mail' and 'Senha' (second row); 'CEP' and 'Endereço' (third row, with 'Endereço' containing 'Avenida João Naves de Ávila'); 'Bairro', 'Cidade', and 'Estado' (bottom row, with 'Estado' as a dropdown menu). A checked checkbox 'Concordo com os termos' and a blue '+ Cadastrar' button are at the bottom.

Visualização em computador desktop / tela larga



A screenshot of the same registration form on a smartphone. The form is stacked vertically. The fields are: 'Nome completo' and 'CPF' (top row); 'E-mail' (second row); 'Senha' (third row); 'CEP' (fourth row); 'Endereço' (fifth row, containing 'Avenida João Naves de Ávila'); 'Bairro' (sixth row); 'Cidade' (seventh row); 'Estado' (eighth row, dropdown menu); a checked checkbox 'Concordo com os termos' (ninth row); and a blue '+ Cadastrar' button (bottom row).

Visualização no smartphone

# O que preciso para utilizar o Bootstrap 5?

1. Tag de responsividade (`<meta name="viewport" ...`)
2. Arquivo CSS do Bootstrap
3. Arquivo JavaScript do Bootstrap e biblioteca Popper

# Página Hello World com o Bootstrap

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <!-- 1: Tag de responsividade -->
    <meta name="viewport" content="width=device-width, initial-scale=2">
    <title>Hello Bootstrap</title>

    <!-- 2: Link para arquivo CSS do Bootstrap -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"

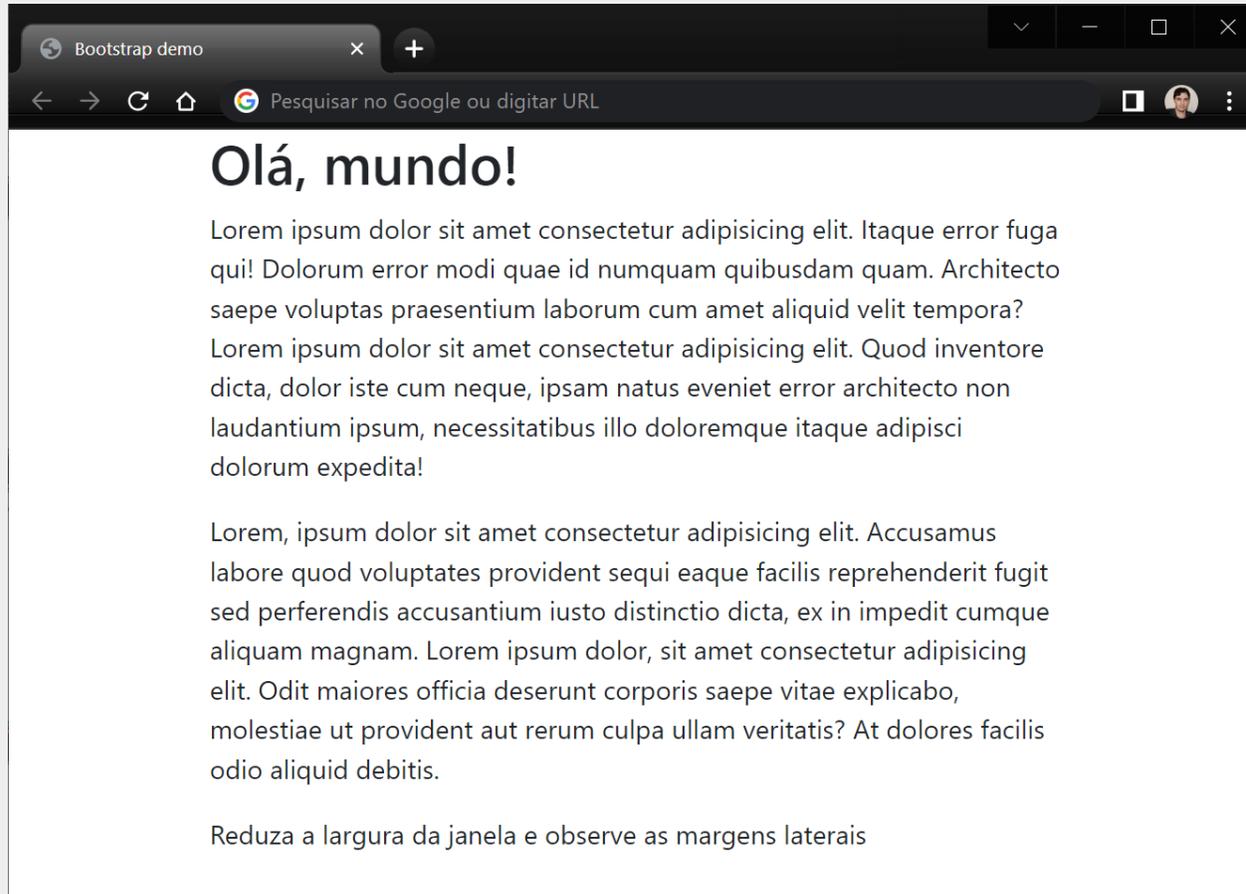
  </head>
  <body>
    <!-- Neste exemplo o conteúdo da página é inserido dentro de um elemento
    utilizando a classe 'container' do Bootstrap. Isso permite que o conteúdo
    da página apareça centralizado quando ela for acessada de um desktop -->
    <div class="container">
      <h1>Olá, mundo!</h1>
    </div>

    <!-- 3: Link para arquivo JavaScript do Bootstrap e biblioteca Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
  </body>
</html>
```

Obtenha o template para a primeira página em:

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

# Página Hello World com o Bootstrap



Visualização em telas largas, com margens nas laterais e conteúdo centralizado



Visualização em telas estreitas, ocupando 100% da largura (smartphones)

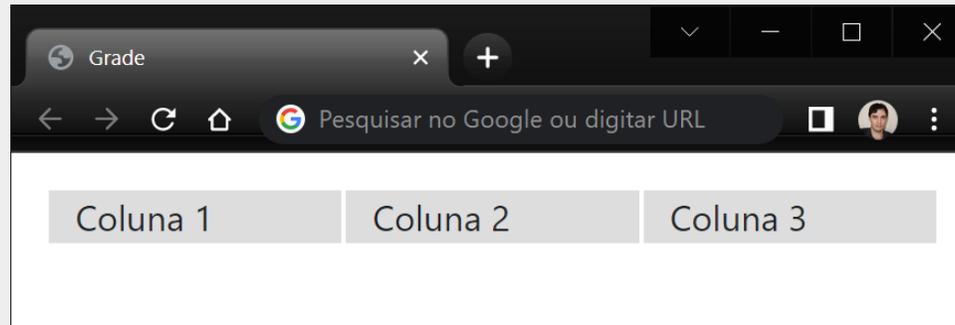
# Sistema de Grade do Bootstrap

- Permite exibir conteúdo em linhas e colunas, mas com foco em **responsividade**
- Baseado no módulo **flexbox** da CSS e em **media queries**
- Linhas são containers para colunas
- Colunas são flexíveis
  - Até 12 "espaços" em cada linha
  - Podem ser combinadas
- Ideia central
  - Telas largas → colunas na horizontal
  - Telas estreitas → colunas empilhadas na vertical

# Sistema de Grade

- Classe CSS `row`
  - Permite adicionar uma nova linha na grade
- Classe CSS `col` (ou variação)
  - Permite adicionar uma nova coluna na linha
- Classes adicionadas a containers, como o `<div>`

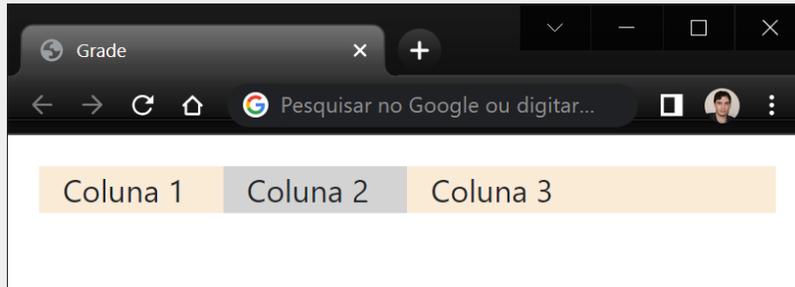
# Exemplo - Colunas de Mesma Largura



```
<div class="row">  
  <div class="col">Coluna 1</div>  
  <div class="col">Coluna 2</div>  
  <div class="col">Coluna 3</div>  
</div>
```

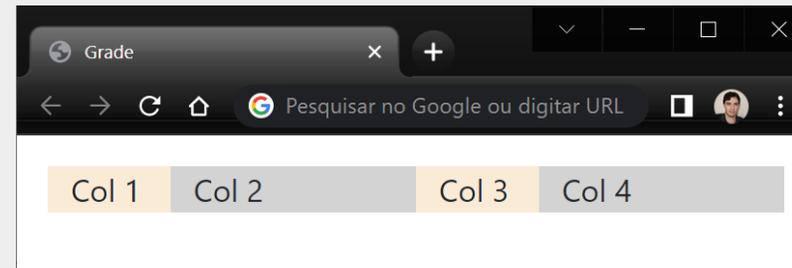
- Grade com uma linha e três colunas de mesma largura
- Aparecerão na horizontal em todos os aparelhos (sem quebra)

# Exemplo - Colunas de Larguras Individuais



```
<div class="row">  
  <div class="col-3">Col 1</div>  
  <div class="col-3">Col 2</div>  
  <div class="col-6">Col 3</div>  
</div>
```

Duas primeiras colunas de mesma largura  
Terceira coluna com o dobro das outras



```
<div class="row">  
  <div class="col-2">Col 1</div>  
  <div class="col-4">Col 2</div>  
  <div class="col-2">Col 3</div>  
  <div class="col-4">Col 4</div>  
</div>
```

Colunas 2 e 4 com dobro da largura de 1 e 3

Nos dois exemplos as colunas aparecerão na horizontal (sem quebra) em todos os aparelhos

# Breakpoints (grid tiers)

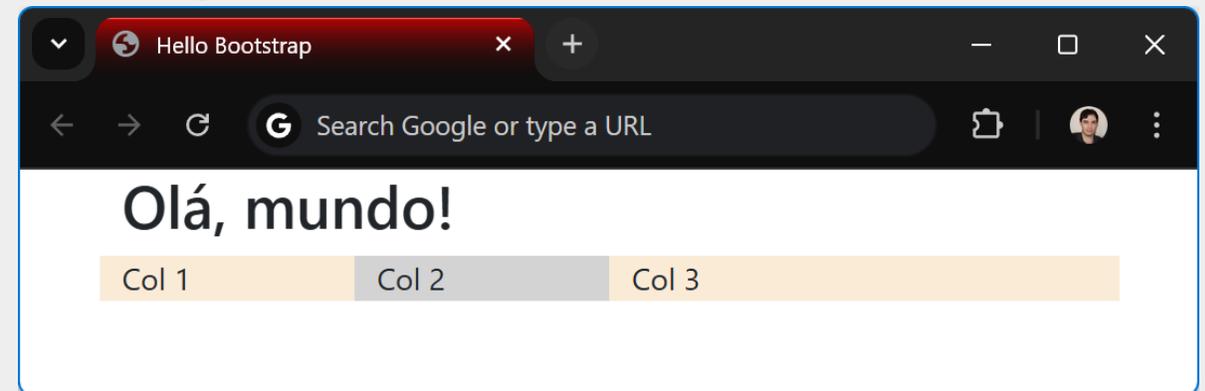
- Permitem criar grades responsivas (conceito *mobile first*)
- Definem quando o layout será ajustado nos dispositivos
- As colunas aparecem **empilhadas na vertical** em viewports com **largura inferior** ao breakpoint (ponto de quebra)
- As colunas aparecem na **horizontal** em viewports com largura **igual ou superior** ao breakpoint

# Breakpoints – Exemplo Inicial

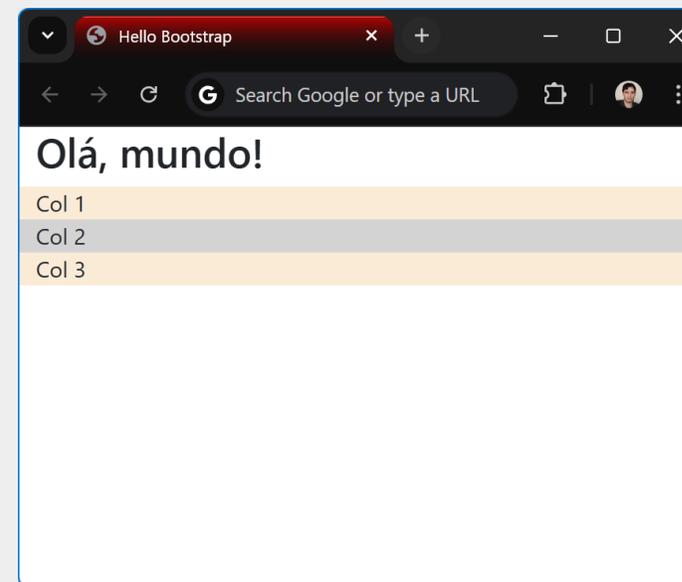
```
<body>
  <div class="container">
    <h1>Olá, mundo!</h1>
    <div class="row">
      <div class="col-sm-3">Col 1</div>
      <div class="col-sm-3">Col 2</div>
      <div class="col-sm-6">Col 3</div>
    </div>
  </div>
</body>
```

Neste exemplo temos uma grade flexível com 3 colunas. A utilização das letras **sm** nas classes de coluna definem o breakpoint **small**, fazendo com que as colunas apareçam 'quebradas' na vertical em telas estreitas (celulares na posição retrato, [viewport < 576](#)). Nos demais aparelhos as colunas aparecerão na horizontal (tablets, notebooks, desktops e celulares na posição paisagem).

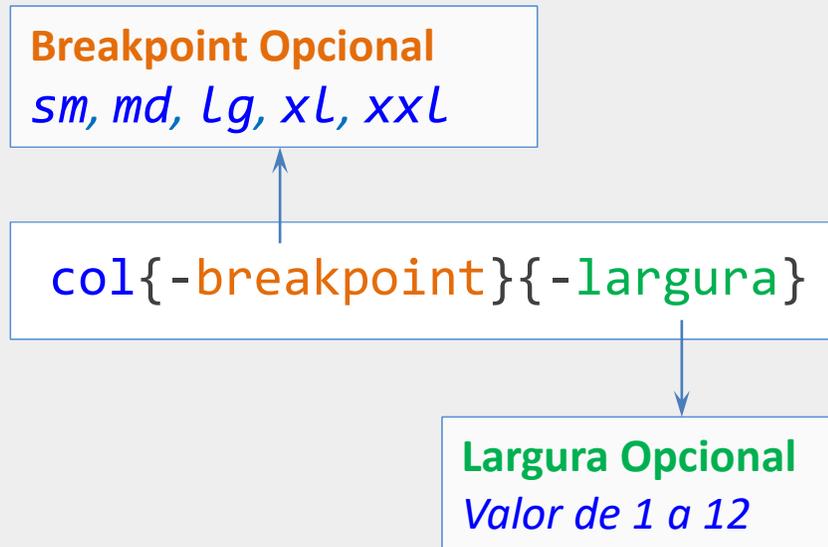
Tela "larga"



Tela "estreita"



# Variações Possíveis para a Classe col



- Com as variações é possível criar grades **flexíveis** (responsivas)
- Grades flexíveis podem se apresentar na **horizontal** ou na **vertical**
- Exemplos de variações: col-2, col-sm-6, col-lg-4

# Breakpoints (grid tiers)

Breakpoint	xs	sm	md	lg	xl	xxl
Denominação	extra small	small	medium	large	extra large	extra extra large
Larg. viewport	> 0 px	≥ 576px	≥ 768px	≥ 992px	≥ 1200px	≥1400px
Prefixo da classe	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
Exemplos	col col-2 col-3 ...	col-sm col-sm-2 col-sm-3 ...	col-md col-md-2 col-md-3 ...	col-lg col-lg-2 col-lg-3 ...	col-xl col-xl-2 col-xl-3 ...	col-xxl col-xxl-2 col-xxl-3 ...

Vejas exemplo prático em <https://youtu.be/dSU7NI-1uaw?t=1615>

# Gutters

- Espaçamentos na grade entre colunas ou linhas
- Classes no Bootstrap 5
  - **gx-\*** → gutter horizontal (espaçamento entre as colunas)
  - **gy-\*** → gutter vertical (espaçamento entre as linhas)
  - **g-\*** → gutter horizontal e vertical
  - **\*** pode ser valores de 0 a 5

# Formulários

Classes do Bootstrap para aprimoramento de campos **textuais, rótulos** e ***select***

Classe Bootstrap	Descrição
<code>.form-control</code>	Para campos textuais - input e textarea
<code>.form-label</code>	Para rótulos (label) desses campos
<code>.form-select</code>	Para o elemento <code>&lt;select&gt;</code>

Exemplo de uso

```
<div>  
  <label for="nome" class="form-label">Nome completo</label>  
  <input type="text" class="form-control" id="nome">  
</div>
```

Resultado

CPF	RG
<input type="text"/>	<input type="text"/>

# Formulários

Classes do Bootstrap para aprimoramento de campos **radio**, **checkbox** e rótulos

Classe Bootstrap	Descrição
<code>.form-check-input</code>	Para campos tipo radio e checkbox
<code>.form-check-label</code>	Para rótulos (label) desses campos
<code>.form-check</code>	Para container adicional agrupando o rótulo e o campo
<code>.form-switch</code>	Exibição estilo chave liga/desliga 

Exemplo de uso

```
<div class="form-check">  
  <input type="checkbox" class="form-check-input" id="termos">  
  <label for="termos" class="form-check-label">Concordo</label>  
</div>
```

Resultado

Concordo com os termos

Se utilizada, a classe `form-switch` deve ser acrescentada depois de `form-check`

# Formulários Responsivos

Layout complexos com sistema de grade e **responsividade**

CPF	RG
<input type="text"/>	<input type="text"/>

```
<div class="row">
  <div class="col-sm">
    <label for="cpf" class="form-label"> CPF </label>
    <input type="text" class="form-control" id="cpf">
  </div>
  <div class="col-sm">
    <label for="rg" class="form-label"> RG </label>
    <input type="text" class="form-control" id="rg">
  </div>
</div>
```

Veja exemplo prático em <https://youtu.be/dSU7NI-1uaw?t=2181>

# Campos com Rótulos Flutuantes

Bootstrap 5 possibilita campos com rótulos flutuantes

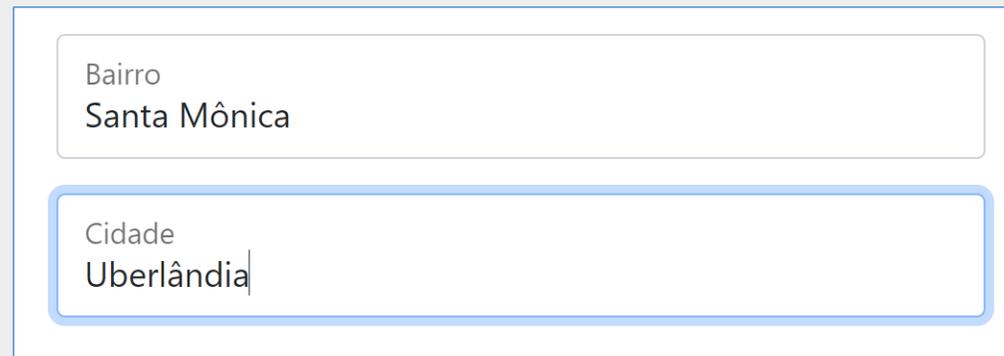
The image displays two examples of Bootstrap 5 form fields with floating labels. The left example shows two empty form fields. The top field is labeled 'Bairro' and the bottom field is labeled 'Cidade'. The right example shows the same two form fields, but with text input. The top field contains 'Santa Mônica' and the bottom field contains 'Uberlândia'. The labels 'Bairro' and 'Cidade' are positioned above the input boxes and are partially obscured by the text inside, demonstrating the floating label effect.

- O container do campo/rótulo deve usar a classe **form-floating**
- O **label** deve ser inserido **depois** do **input**
- É **obrigatório** o atributo **placeholder** com um valor qualquer diferente de vazio
- O **conteúdo do label** é utilizado como rótulo flutuante (**e não o placeholder**)

# Campos com Rótulos Flutuantes – Exemplo

```
<div class="form-floating">
  <input type="text" class="form-control" id="bairro" placeholder="">
  <label for="bairro"> Bairro </label>
</div>

<div class="form-floating">
  <input type="text" class="form-control" id="cidade" placeholder="">
  <label for="cidade"> Cidade </label>
</div>
```

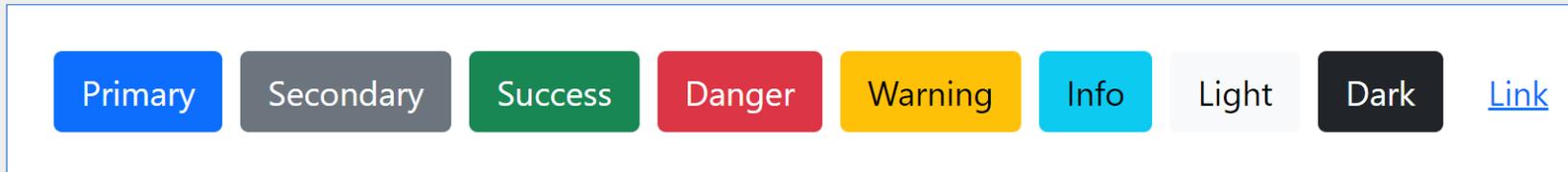


Bairro  
Santa Mônica

Cidade  
Uberlândia

# Botões Contextuais com Background

## Estilos de botões



## Classes CSS

<code>.btn</code>	<code>.btn-warning</code>
<code>.btn-primary</code>	<code>.btn-info</code>
<code>.btn-secondary</code>	<code>.btn-light</code>
<code>.btn-success</code>	<code>.btn-dark</code>
<code>.btn-danger</code>	<code>.btn-link</code>

## Exemplo

```
<button type="button" class="btn btn-primary">Primary</button>
```

# Tabelas

Classe Bootstrap	Significado
<code>.table</code>	Classe principal. Tabela com divisões horizontais
<code>.table-striped</code>	Adiciona cores alternadas às linhas
<code>.table-bordered</code>	Adiciona bordas em todas as células
<code>.table-hover</code>	Adiciona cor à linha quando apontada pelo mouse

```
<table class="table table-striped table-hover">
```

# Alerts

Operação realizada com sucesso!

```
<div class="alert alert-success" role="alert">  
  Operação realizada com sucesso!  
</div>
```

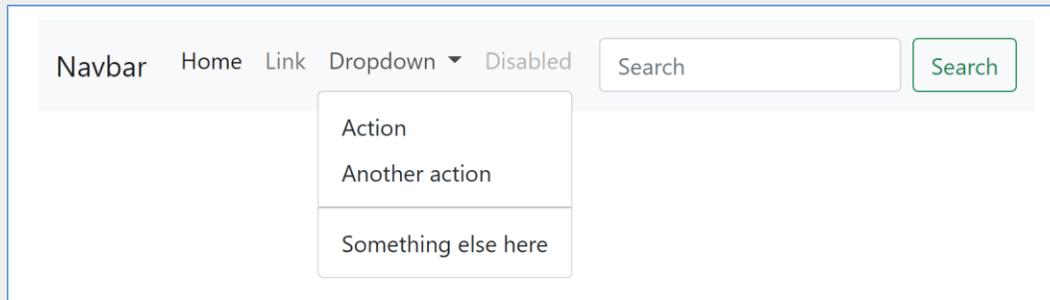
**Atenção:** a operação não poderá ser desfeita!



```
<div class="alert alert-warning alert-dismissible" role="alert">  
  <strong>Atenção:</strong> a operação não poderá ser desfeita!  
  <button type="button" class="btn-close" data-dismiss="alert"></button>  
</div>
```

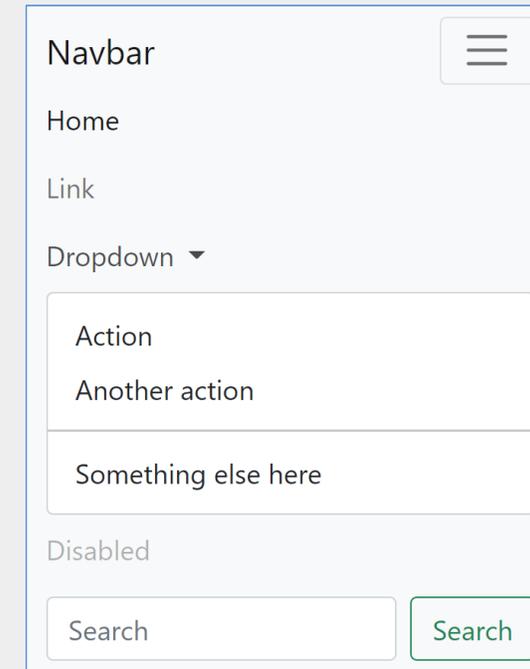
# Navbar Responsiva

## Barra de navegação em dispositivos com tela larga



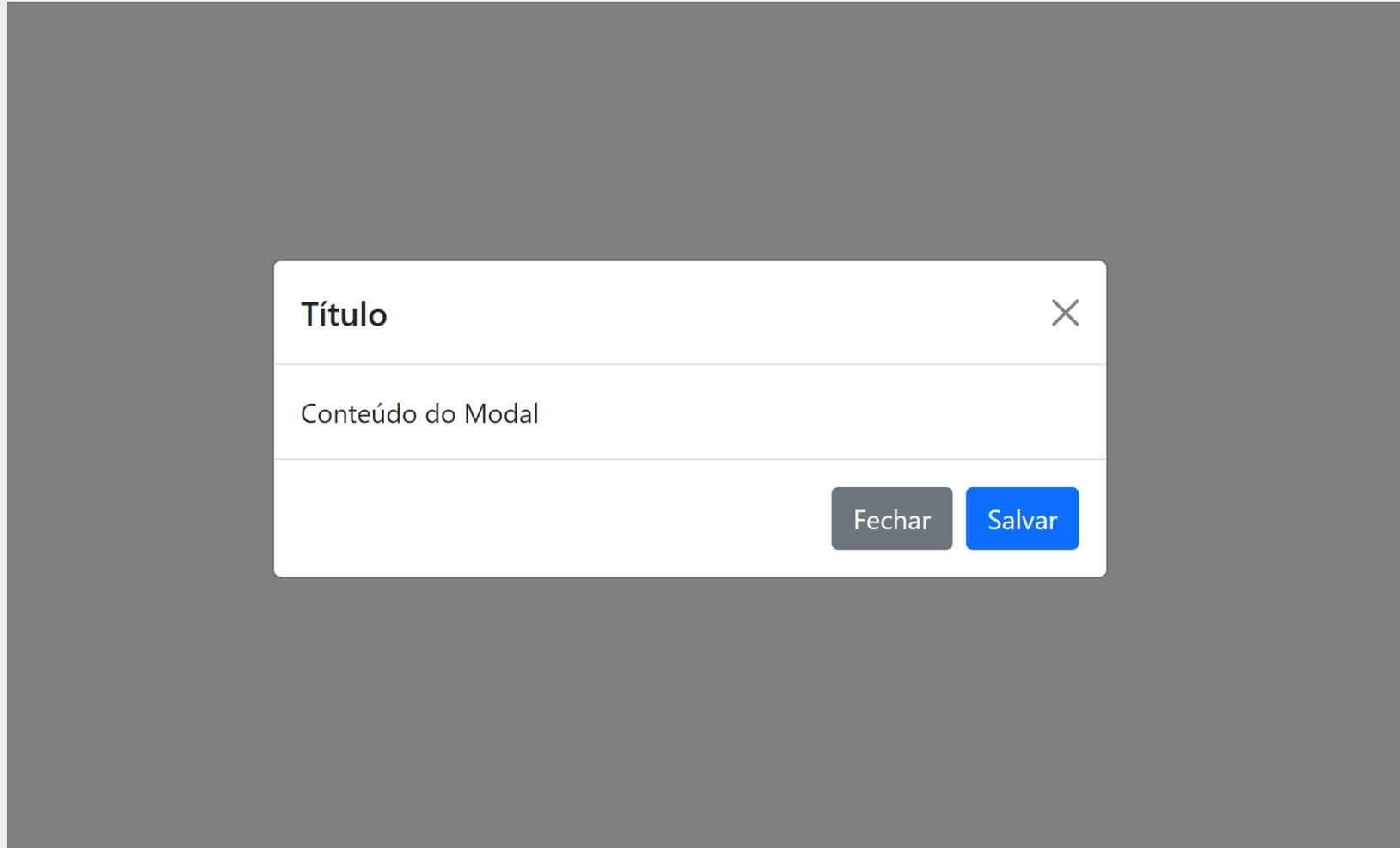
```
<nav class="navbar navbar-expand-sm">  
  
...  
  
</nav>
```

## Barra com botão em tela estreita



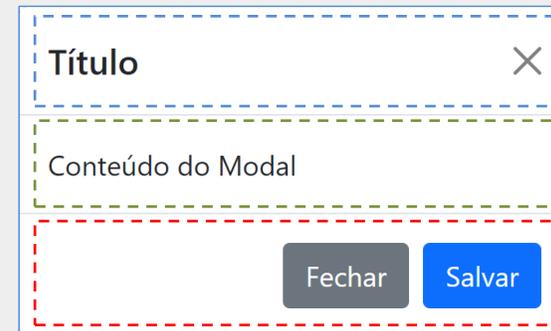
<https://getbootstrap.com/docs/5.3/components/navbar/>

# Modal



# Modal

```
<div class="modal fade" id="meuModal">  
  <div class="modal-dialog">  
    <div class="modal-content">  
  
      <div class="modal-header">  
        <h5 class="modal-title">Título</h5>  
        <button data-bs-dismiss="modal"></button>  
      </div>  
  
      <div class="modal-body">  
        Conteúdo do Modal  
      </div>  
  
      <div class="modal-footer">  
        <button data-bs-dismiss="modal">Fechar</button>  
        <button>Salvar</button>  
      </div>  
  
    </div>  
  </div>  
</div>
```



# Modal - Observações

- Atributos da `div` principal da janela modal
  - `class="modal"`
  - `id="meuModal"`
- Atributos do botão para abrir a janela modal
  - `data-bs-target="#meuModal"`
  - `data-bs-toggle="modal"`
- Atributos do botão interno para fechar a janela modal
  - `data-bs-dismiss="modal"`

# Referências

- <https://getbootstrap.com/docs>
- [https://developer.mozilla.org/en-US/docs/Mozilla/Mobile/Viewport\\_meta\\_tag](https://developer.mozilla.org/en-US/docs/Mozilla/Mobile/Viewport_meta_tag)
- <https://icons.getbootstrap.com>
- <https://fonts.google.com/icons>
- [https://developers.google.com/fonts/docs/material\\_icons](https://developers.google.com/fonts/docs/material_icons)