

## O QUE É **BOOTSTRAP** E SUA IMPORTÂNCIA

**Bootstrap** é um framework front-end open-source que facilita o desenvolvimento de sites e aplicações web responsivas e modernas. Criado pela equipe do Twitter, ele fornece uma coleção de ferramentas CSS e JavaScript, incluindo estilos, componentes de interface e plug-ins. A importância do Bootstrap se deve a vários fatores:

- **Rapidez e Eficiência:** Acelera o processo de desenvolvimento, fornecendo estilos e componentes prontos para uso.
- **Responsividade:** Inclui um sistema de grid que permite criar layouts que se ajustam a diferentes tamanhos de tela.
- **Consistência:** Garante que todos os desenvolvedores em um projeto utilizem os mesmos estilos e componentes, mantendo a uniformidade.
- **Customização:** Permite a personalização fácil de estilos e componentes através de variáveis Sass.
- **Suporte da Comunidade:** Tem uma grande comunidade de desenvolvedores, oferecendo suporte e recursos adicionais.

### O que é um Framework?

Framework é um conjunto de bibliotecas, que abordam funcionalidades, e estruturas, para o desenvolvimento de aplicações, a fim de fornecer soluções para um mesmo domínio de problema, permitindo a reutilização do seu código. Assim, através das diversas linguagens de programação, os Frameworks são criados e identificados pelas **metodologias, propósitos, e implementações**, aos quais os diversos tipos de aplicações, na maioria **orientadas a objetos**, poderão reutilizar suas estruturas e códigos.

Um Framework é formado por um **conjunto classes** implementadas em uma determinada linguagem de programação, que fornece recursos **comuns** já **prontos, validados e testados**, os quais podem ser usados para auxiliar o desenvolvimento de software, viabilizando maior eficiência na resolução dos problemas, otimização de recursos, e detecção de erros. Ele fornece um modelo de dados, usados para resolver um problema específico, abstraindo parte do código das soluções que estamos tratando.

Na programação orientada a objetos, especificamente, um Framework é um conjunto de **classes e interfaces**, com objetivo da reutilização de arquitetura de software, é composto por objetos, pelos mapeamentos das suas responsabilidades, que através de interfaces, com um fluxo de controle definido, possibilitará a interação com aplicações, ou seja é ele quem irá guiar à solução de um domínio específico.

Um Framework possui duas partes fundamentais:

**Hot-Spots** – são as partes específicas de sistemas individuais, projetados para serem genéricos, os quais podem ser adaptados às necessidades da aplicação.

**Frozen-Spots** - definem a arquitetura geral de um sistema de software, seus componentes básicos e os relacionamentos entre eles, imutáveis em todas as instanciações do Framework.

### **Qual é a diferença entre um Framework e uma Biblioteca?**

A fim de fornecer uma melhor distinção sobre um Framework, devemos esclarecer alguns fundamentos de uma Biblioteca, ou Library, com o propósito de evitar qualquer tipo de confusão entre eles, e que também irá nos fornecer um melhor entendimento sobre um Framework.

Uma Biblioteca é uma coleção de implementações, escritas sob os termos de uma certa linguagem, fornecendo recursos, independentes, para determinado comportamento ser executado em uma aplicação. Assim, o programa invoca o comportamento fornecido por meio de um mecanismo da linguagem, possuindo assim o controle sobre os recursos fornecidos pela biblioteca.

Já um Framework é um conjunto de classes e interfaces que impõe um modelo colaborativo e um padrão de interação ao qual o nosso programa deverá se adaptar, as dependências entre as classes são embutidas, com a comunicação entre os objetos já definida. Então o Framework possui a responsabilidade para definir o comportamento da aplicação e chamar cada método, ou seja é o Framework que chama o código da nossa aplicação, e é ele quem tem o controle sobre a nossa aplicação, definindo o seu comportamento e o fluxo de controle da aplicação, esse conceito é chamado de Inversão de Controle.

### **Tipos de Frameworks.**

**Framework de Suporte** ⇒ Disponibiliza serviços para sistemas operacionais, e como nosso objetivo principal é a informação de dados técnicos para desenvolvimento de softwares, não será detalhado neste artigo.

**Framework de aplicação** ⇒ Também chamado de Framework Horizontal, encapsula conhecimentos aplicáveis a vários tipos de aplicações, ou seja, eles podem ser usados em diferentes domínios, sem qualquer dependência dos mesmos. Ele resolve apenas uma parte da solução do problema, como por exemplo framework para construção de interface GUI ou interface gráfica.

**Framework de domínio** ⇒ Também chamado de Framework Vertical, ou Framework Especialista, encapsula conhecimentos aplicáveis a aplicações com um domínio particular em comum, obtidos através de um determinado contexto específico. Ele resolve boa parte da aplicação, e temos como exemplo um Framework para aplicações de controle financeiro.

## Configuração do ambiente

Para começar a usar Bootstrap, você pode optar por incluí-lo via CDN ou instalá-lo localmente.

### Via CDN:

Adicione as seguintes linhas no `<head>` do seu documento HTML:

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootst
rap.min.css" rel="stylesheet"

integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh
0JMhY6hW+ALEwIH" crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstra
p.bundle.min.js"

integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9Gkc
IdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
```

E no final:

```
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"

integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR
+Vc4jQkC+hVqc2pM8ODewa9r"
  crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/
bootstrap.min.js"

integrity="sha384-0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLw
CsOGabYfZo0T0to5eqruptLy"
  crossorigin="anonymous"></script>
```

CONSULTAR [DOCUMENTAÇÃO GUIA DE INÍCIO](#)

### Instalação Local:

1. Baixe o Bootstrap do site oficial.
2. Extraia os arquivos e mova-os para o seu projeto.
3. Inclua os arquivos CSS e JS no seu HTML:

```
<link href="path/to/bootstrap.min.css"
rel="stylesheet">
<script src="path/to/jquery.slim.min.js"></script>
<script src="path/to/popper.min.js"></script>
<script src="path/to/bootstrap.min.js"></script>
```

### Estrutura básica de um documento com Bootstrap

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>Bootstrap Example</title>
<link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.
2/css/bootstrap.min.css" rel="stylesheet">
</head>

<body>
  <div class="container">
    <h1>Hello, world!</h1>
  </div>
  <script
src="https://code.jquery.com/jquery-3.5.1.slim.min.js">
</script>
  <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/
dist/umd/popper.min.js"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2
/js/bootstrap.min.js"></script>
</body>

</html>
```

## Layout com Contêineres e Grade (Grid System)

### Contêineres

Os contêineres são usados para centralizar e dar espaço horizontal ao seu conteúdo.

- **.container**: Fornece uma largura fixa responsiva.
- **.container-fluid**: Ocupa toda a largura disponível.

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

```
    <!-- Conteúdo aqui -->
</div>
<div class="container-fluid">
    <!-- Conteúdo aqui -->
</div>
```

## Sistema de grade (Grid System)

Bootstrap usa um sistema de 12 colunas para criar layouts responsivos.

- **.row**: Cria uma linha de colunas.
- **.col**: Cria uma coluna dentro de uma linha.

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

## Colunas responsivas e breakpoints

Você pode criar colunas que se adaptam a diferentes tamanhos de tela usando breakpoints.

```
<div class="container">
  <div class="row">
    <div class="col-sm">
```

```
        Pequena
    </div>
    <div class="col-md">
        Média
    </div>
    <div class="col-lg">
        Grande
    </div>
</div>
</div>
```

## Layout com Contêineres e Grade (Grid System)

- Contêineres (container, container-fluid)
- Sistema de grade (row, col)
- Colunas responsivas e breakpoints

## Tipografia e Texto

### Classes de tipografia

- Headings: `.h1`, `.h2`, `.h3`, `.h4`, `.h5`, `.h6`
- Display headings: `.display-1`, `.display-2`, `.display-3`, `.display-4`
- Lead: `.lead`
- Small: `<small>`
- Blockquote: `.blockquote`

```
<h1 class="h1">Heading 1</h1>
<h2 class="display-2">Display Heading 2</h2>
<p class="lead">This is a lead paragraph.</p>
<small>Small text</small>
<blockquote class="blockquote">
    <p class="mb-0">A blockquote example.</p>
</blockquote>
```

### Alinhamento e transformação de texto

- **Alinhamento:** `.text-left`, `.text-center`, `.text-right`
- **Transformação:** `.text-uppercase`, `.text-lowercase`, `.text-capitalize`

```
<p class="text-center">Centered text</p>
<p class="text-uppercase">Uppercase text</p>
```

### Utilização de fontes e utilitários de texto

Você pode utilizar utilitários como `.font-weight-bold`, `.font-italic`, `.text-muted`.

```
<p class="font-weight-bold">Bold text</p>
<p class="font-italic">Italic text</p>
<p class="text-muted">Muted text</p>
```

### Cores e Contexto

#### Classes de cores de fundo

- **Cores básicas:** `.bg-primary`, `.bg-secondary`, `.bg-success`, `.bg-danger`, `.bg-warning`, `.bg-info`, `.bg-light`, `.bg-dark`

```
<div class="bg-primary text-white">Primary
background</div>
<div class="bg-success text-white">Success
background</div>
```

#### Classes de cores de texto

- **Cores básicas:** `.text-primary`, `.text-secondary`, `.text-success`, `.text-danger`, `.text-warning`, `.text-info`, `.text-light`, `.text-dark`

```
<p class="text-primary">Primary text</p>
<p class="text-danger">Danger text</p>
```

### Contexto e combinações de cores para elementos



Use combinações de cores para destacar e diferenciar elementos, mantendo a legibilidade e acessibilidade.

## Componentes Básicos

### Botões

- **Botões padrão:** `.btn`, `.btn-primary`, `.btn-secondary`, `.btn-success`, `.btn-danger`

```
<button class="btn btn-primary">Primary  
Button</button>  
<button class="btn btn-secondary">Secondary  
Button</button>
```

- **Grupos de botões:**

`.btn-group`

```
<div class="btn-group" role="group">  
  <button class="btn btn-primary">Left</button>  
  <button class="btn btn-primary">Middle</button>  
  <button class="btn btn-primary">Right</button>  
</div>
```

### Imagens

- **Imagens fluidas:** `.img-fluid`
- **Imagens em miniatura:** `.img-thumbnail`

```
  

```

### Ícones

Bootstrap Icons são uma biblioteca de ícones que pode ser usada com o Bootstrap.

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/bootst
rap-icons/1.8.1/font/bootstrap-icons.min.css">
<i class="bi bi-alarm"></i>
```

## Navegação e Barras de Navegação (Navbar)

### Navegação básica

- Navegação simples:

.nav, .nav-item, .nav-link

```
<ul class="nav">
  <li class="nav-item">
    <a class="nav-link active"
href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
</ul>
```

- Nav-pills: .nav-pills
- Nav-tabs: .nav-tabs

```
<ul class="nav nav-pills">
  <li class="nav-item">
    <a class="nav-link active"
href="#">Active</a>
  </li>
</ul>
<ul class="nav nav-tabs">
  <li class="nav-item">
```

```
        <a class="nav-link active"
href="#">Active</a>
    </li>
</ul>
```

### Barras de navegação responsivas

- **Navbar:** `.navbar`, `.navbar-expand`
- **Marca:** `.navbar-brand`
- **Botão de alternância:** `.navbar-toggler`, `.navbar-toggler-icon`
- **Colapso:** `.navbar-collapse`

```
<nav class="navbar navbar-expand-lg navbar-light
bg-light">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="#navbarNav"
    aria-controls="navbarNav"
    aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse"
id="navbarNav">
        <ul class="navbar-nav">
            <li class="nav-item active">
                <a class="nav-link" href="#">Home
<span class="sr-only">(current)</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link"
href="#">Features</a>
            </li>
        </ul>
```

```
</div>  
</nav>
```

## Componentes e Utilitários Intermediários

### Formulários Básicos

- Estrutura básica: `.form-group`, `.form-control`

```
<form>  
  <div class="form-group">  
    <label for="exampleInputEmail">Email  
address</label>  
    <input type="email" class="form-control"  
id="exampleInputEmail"  
aria-describedby="emailHelp">  
    <small id="emailHelp" class="form-text  
text-muted">We'll never share your email with  
anyone else.</small>  
  </div>  
  <div class="form-group">  
    <label  
for="exampleInputPassword1">Password</label>  
    <input type="password" class="form-control"  
id="exampleInputPassword1">  
  </div>  
</form>
```

- Campos de entrada, seleção e texto:

`<input>`, `<select>`, `<textarea>`

```

<form>
  <div class="form-group">
    <label
for="exampleFormControlSelect1">Example
select</label>
    <select class="form-control"
id="exampleFormControlSelect1">
      <option>1</option>
      <option>2</option>
      <option>3</option>
    </select>
  </div>
  <div class="form-group">
    <label
for="exampleFormControlTextarea1">Example
textarea</label>
    <textarea class="form-control"
id="exampleFormControlTextarea1"
rows="3"></textarea>
  </div>
</form>

```

- **Classes de validação de formulários**

```

<form class="needs-validation" novalidate>
  <div class="form-group">
    <label for="validationCustom01">First
name</label>
    <input type="text" class="form-control"
id="validationCustom01" required>
    <div class="valid-feedback">Looks good!</div>
  </div>
</form>

```

## Formulários Avançados

- Layouts de formulários:

`.form-inline, .form-row`

```
<form class="form-inline">
  <div class="form-group mb-2">
    <label for="staticEmail2"
class="sr-only">Email</label>
    <input type="text" readonly
class="form-control-plaintext" id="staticEmail2"
value="email@example.com">
  </div>
  <div class="form-group mx-sm-3 mb-2">
    <label for="inputPassword2"
class="sr-only">Password</label>
    <input type="password" class="form-control"
id="inputPassword2" placeholder="Password">
  </div>
  <button type="submit" class="btn btn-primary
mb-2">Confirm identity</button>
</form>
```

- Componentes de formulários:

`.input-group, .custom-control`

```
<div class="input-group mb-3">
  <div class="input-group-prepend">
    <span class="input-group-text"
id="basic-addon1">@</span>
  </div>
  <input type="text" class="form-control"
placeholder="Username" aria-label="Username">
</div>
```

```
aria-describedby="basic-addon1">
</div>
```

- **Feedback de validação**

```
<form class="needs-validation" novalidate>
  <div class="form-row">
    <div class="col-md-4 mb-3">
      <label for="validationCustom01">First
name</label>
      <input type="text" class="form-control"
id="validationCustom01" required>
      <div class="valid-feedback">
        Looks good!
      </div>
    </div>
  </div>
</form>
```

## Cartões e List Groups

- **Cartões:** `.card`, `.card-header`, `.card-body`, `.card-footer`

```
<div class="card" style="width: 18rem;">
  <div class="card-header">Featured</div>
  <div class="card-body">
    <h5 class="card-title">Special title
treatment</h5>
    <p class="card-text">With supporting text
below as a natural lead-in to additional
content.</p>
    <a href="#" class="btn btn-primary">Go
somewhere</a>
  </div>
```

```
<div class="card-footer text-muted">2 days  
ago</div>  
</div>
```

- **List Groups:** `.list-group`, `.list-group-item`

```
<ul class="list-group">  
  <li class="list-group-item">Cras justo odio</li>  
  <li class="list-group-item">Dapibus ac facilisis  
  in</li>  
</ul>
```

- **Combinação de cartões e list groups**

```
<div class="card" style="width: 18rem;">  
  <ul class="list-group list-group-flush">  
    <li class="list-group-item">Cras justo  
odio</li>  
    <li class="list-group-item">Dapibus ac  
facilisis in</li>  
  </ul>  
</div>
```

## Modais e Janelas de Diálogo

- **Estrutura básica de um modal:** `.modal`, `.modal-dialog`, `.modal-content`

```
<div class="modal" tabindex="-1" role="dialog">  
  <div class="modal-dialog" role="document">  
    <div class="modal-content">  
      <div class="modal-header">
```



```

        <h5 class="modal-title">Modal
title</h5>
        <button type="button" class="close"
data-dismiss="modal" aria-label="Close">
            <span
aria-hidden="true">&times;</span>
        </button>
    </div>
    <div class="modal-body">
        <p>Modal body text goes here.</p>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn
btn-primary">Save changes</button>
        <button type="button" class="btn
btn-secondary" data-dismiss="modal">Close</button>
    </div>
</div>
</div>

```

- **Triggers para abrir modais:** botões, links

```

<button type="button" class="btn btn-primary"
data-toggle="modal" data-target="#exampleModal">
    Launch demo modal
</button>

```

- **Modais com conteúdo dinâmico**

```

<script>
    $('#exampleModal').on('show.bs.modal', function
(event) {
        var button = $(event.relatedTarget)
        var recipient = button.data('whatever')
        var modal = $(this)
        modal.find('.modal-title').text('New
message to ' + recipient)
        modal.find('.modal-body
input').val(recipient)
    })
</script>

```

## Alertas e Badges

- **Alertas:** `.alert`, `.alert-dismissible`

```

<div class="alert alert-warning alert-dismissible
fade show" role="alert">
    <strong>Holy guacamole!</strong> You should check
in on some of those fields below.
    <button type="button" class="close"
data-dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&times;</span>
    </button>
</div>

```

- **Badges:** `.badge`, `.badge-pill`

```

<span class="badge badge-primary">Primary</span>

```

```
<span class="badge badge-pill  
  badge-secondary">Secondary</span>
```

- Fechamento automático de alertas

```
<script>  
  $(".alert").alert()  
</script>
```

## Carrosséis e Sliders

- Estrutura básica de um carrossel: `.carousel`, `.carousel-inner`, `.carousel-item`

```
<div id="carouselExampleIndicators" class="carousel  
  slide" data-ride="carousel">  
  <ol class="carousel-indicators">  
    <li data-target="#carouselExampleIndicators"  
      data-slide-to="0" class="active"></li>  
    <li data-target="#carouselExampleIndicators"  
      data-slide-to="1"></li>  
  </ol>  
  <div class="carousel-inner">  
    <div class="carousel-item active">  
        
    </div>  
    <div class="carousel-item">  
        
    </div>  
  </div>  
</div>
```

- Controles e indicadores

```
<a class="carousel-control-prev"
  href="#carouselExampleIndicators" role="button"
  data-slide="prev">
  <span class="carousel-control-prev-icon"
    aria-hidden="true"></span>
  <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next"
  href="#carouselExampleIndicators" role="button"
  data-slide="next">
  <span class="carousel-control-next-icon"
    aria-hidden="true"></span>
  <span class="sr-only">Next</span>
</a>
```

- Sliders automáticos e manuais

```
$('#carouselExampleIndicators').carousel({
  interval: 2000
})
```

## Customização e Extensões

### Customização Básica com CSS

- Substituição de variáveis padrão do Bootstrap
- Customização de cores, espaçamento e tipografia
- Utilização de um arquivo CSS customizado

```
:root {
  --primary: #007bff;
  --secondary: #6c757d;
}
```

```
.custom-css {  
    padding: 20px;  
    background-color: var(--primary);  
}
```

## Plugins JavaScript do Bootstrap

- Introdução aos plugins JavaScript
- Utilização de plugins como Tooltip, Popover, Scrollspy

```
$(function () {  
    $('[data-toggle="tooltip"]').tooltip();  
    $('[data-toggle="popover"]').popover();  
});
```

- Inicialização de plugins via JavaScript e atributos data

```
<button type="button" class="btn btn-secondary"  
data-toggle="tooltip" data-placement="top"  
title="Tooltip on top">  
    Tooltip on top  
</button>
```

## Layouts Responsivos Avançados

- Técnicas avançadas de layout responsivo
- Utilização de classes de utilitários (spacing, display, flex)
- Criação de layouts complexos e responsivos

```
<div class="d-flex justify-content-between">  
    <div class="p-2">Flex item 1</div>  
    <div class="p-2">Flex item 2</div>  
    <div class="p-2">Flex item 3</div>
```

```
</div>
```

## Performance e Otimização

- Melhores práticas para performance com Bootstrap
- Redução do tamanho do CSS (purge CSS, minificação)
- Carregamento assíncrono de recursos

```
<link rel="stylesheet" href="bootstrap.min.css">  
<script defer src="bootstrap.bundle.min.js"></script>
```

## Substituindo Tags genéricas por Semânticas

substituir as `<div>` por tags semânticas do HTML5 é uma boa prática, pois melhora a acessibilidade e a estrutura do documento, tornando-o mais fácil de entender tanto para humanos quanto para mecanismos de busca. O Bootstrap é compatível com tags semânticas, e as classes podem ser aplicadas a qualquer elemento HTML. Aqui estão algumas recomendações para usar tags semânticas com classes do Bootstrap:

## Exemplos de Tags Semânticas Substituindo `<div>`

### Header

```
<header class="container">  
  <h1>Meu Site</h1>  
</header>
```

### Nav

```
<nav class="navbar navbar-expand-lg navbar-light  
bg-light">  
  <a class="navbar-brand" href="#">Navbar</a>  
</nav>
```

### Main

```
<main class="container">
  <div class="row">
    <article class="col-md-8">
      <h2>Artigo Principal</h2>
      <p>Conteúdo do artigo...</p>
    </article>
    <aside class="col-md-4">
      <h2>Barra Lateral</h2>
      <p>Conteúdo da barra lateral...</p>
    </aside>
  </div>
</main>
```

## Footer

```
<footer class="footer mt-auto py-3 bg-light">
  <div class="container">
    <span class="text-muted">Meu rodapé</span>
  </div>
</footer>
```

## Benefícios do Uso de Tags Semânticas

1. **Acessibilidade:** Leitores de tela e outras tecnologias assistivas entendem melhor a estrutura da página.
2. **SEO:** Mecanismos de busca podem indexar o conteúdo de maneira mais eficaz.
3. **Manutenção:** O código é mais fácil de ler e manter.