



# Containers com Bootstrap

## Introdução

Iniciando a abordagem sobre os fundamentos do Bootstrap, temos os “containers”, que são essencialmente caixas utilizadas para organizar conteúdo em uma página. O Bootstrap oferece três tipos de classes de containers, semelhantes às classes de grid de colunas, que auxiliam no posicionamento e organização dos elementos na página. As três classes variam em sua capacidade de definir largura máxima, adaptar-se a diferentes limites de tamanho de tela e estabelecer uma largura total, independentemente do tamanho do dispositivo. Após entender como essas classes de container operam, elas são postas em prática em um projeto, dando forma a diferentes seções, como cabeçalhos e rodapés, substituindo estilos personalizados pelos padrões do Bootstrap, minimizando a necessidade de criar CSS na folha de estilo pessoal.

Prosseguindo com a exploração do Bootstrap, serão apresentadas as classes para estilizar formulários, permitindo torná-los mais atraentes sem a necessidade de criar CSS personalizado. A prática é aplicada na seção de contato de um site, aprimorando a estética e o alinhamento de rótulos e campos. A estilização de tabelas também será abordada, demonstrando como o Bootstrap simplifica o processo ao fornecer diversas classes pré-definidas que substituem a necessidade de CSS personalizado para cada tabela. Ao final, os estudantes devem ser capazes de aplicar esses conceitos em seus próprios projetos, demonstrando um entendimento sólido do Bootstrap e de suas capacidades na criação de interfaces de usuário responsivas e atraentes.





## Containers

Containers, em sua essência, são caixas que servem para organizar o conteúdo em uma página. O Bootstrap proporciona três tipos de classes de containers, semelhantes às classes de grids de colunas, que auxiliam no posicionamento e organização dos elementos na página.


Os três tipos de classes de containers são:

1. A classe “container”, que define uma largura máxima para a caixa.
2. A classe “container” seguida por um breakpoint, que define uma largura máxima até o limite desse breakpoint.
3. A classe “Container Fluid”, que estabelece uma largura de 100% para qualquer tamanho de tela.

Os breakpoints, mencionados no sistema de grade, são utilizados para ajustar o comportamento em diferentes tamanhos de dispositivos. Por exemplo, utilizando a classe container “SM” (para small devices), a largura máxima será de 540 pixels para dispositivos de tamanho igual ou maior que 576 pixels, e uma largura máxima de 720 pixels para dispositivos iguais ou maiores que 768 pixels, e assim por diante.

Depois de entender as classes de container, a próxima etapa é aplicá-las em um projeto. O exemplo dado inclui um cabeçalho (header), seções (sections) e um rodapé (footer). Em um exemplo comum, atribuímos um plano de fundo ao cabeçalho, alinhamos o texto ao centro e adicionamos um espaçamento (padding). Para o rodapé e as seções, uma margem de 40 pixels é atribuída.

No entanto, o objetivo agora é substituir esses estilos personalizados pelos estilos do framework Bootstrap. A folha de estilo personalizada, às vezes

chamada de “custom style”, é reservada para estilizar elementos específicos. A maioria do site será construída usando as classes do Bootstrap, de  do a minimizar a necessidade de criar CSS na folha de estilo pessoal.

Neste exemplo, o cabeçalho terá a classe “container-fluid” atribuída, o que significa que ocupará 100% do tamanho do dispositivo. Tendências atuais favorecem essa abordagem, embora a utilização de containers com tamanho fixo também seja válida dependendo do projeto. Além disso, a classe de padding do Bootstrap, “padding-2”, é usada para definir um espaçamento relativo.

Outra funcionalidade do Bootstrap é a disponibilidade de classes para atribuir planos de fundo. Neste exemplo, a classe “BG Warning” é usada para atribuir uma cor laranja ao plano de fundo. Para centralizar os elementos, a classe “Text Center” é utilizada.

Ao aplicar essas classes, observa-se que o cabeçalho agora preenche 100% da largura, os elementos estão centralizados e o plano de fundo está laranja. O próximo passo é estilizar outras seções do projeto, alternando as cores de fundo entre “background-light” e “background-warning” para diferenciar as seções.

A conclusão é que o uso das classes de container do Bootstrap é uma ferramenta poderosa para a estilização e organização dos elementos em uma página. No próximo módulo, haverá uma continuidade na exploração dessas e de outras classes do Bootstrap.

## Formulários e Botões

Agora, vamos abordar a aplicação das classes do Bootstrap para estilizar formulários, tornando-os mais atrativos sem a necessidade de criar CSS

personalizado. É possível encontrar todas as classes na documentação do projeto.



Focando no site que está sendo construído, a seção de contato contém um formulário, iniciando com um 'fieldset' (grupo de campos no HTML5). O 'fieldset' é utilizado para agrupar campos e contém a legenda 'Dados Pessoais'.

Os rótulos 'labels' estão desalinhados. Para resolver isso, classes serão atribuídas para dispor os rótulos verticalmente. A estratégia envolve a criação de uma 'div' para cada campo, permitindo que os campos se alinhem verticalmente. A classe 'MB-3', que adiciona margem inferior (bottom), será adicionada a esta 'div'. O rótulo 'label' é movido para dentro da 'div', e o 'input' é movido para fora do 'label'. Atributos são adicionados ao 'label': a classe do Bootstrap 'form-label' e o atributo 'for', que identifica a qual campo se refere o rótulo. A classe do Bootstrap para estilizar o 'input', 'form-control', é adicionada ao 'input'. Esse procedimento melhora a estética do campo.

Esse processo é repetido para todos os campos. O campo de e-mail recebe uma 'div' com a classe 'MB-3', com seu 'label' e 'input' separados. Classes do Bootstrap 'form-label' e 'form-control' são adicionadas ao 'label' e 'input', respectivamente. O 'label' recebe um atributo 'for' para identificar que se refere ao campo de e-mail. O 'input' recebe um 'id' com o mesmo nome que o 'for' para que seja reconhecido.

O campo 'data de ida' é tratado da mesma maneira. Os atributos 'for' e 'id' são atribuídos como 'data-ida' e o campo 'data de volta' tem seus atributos 'for' e 'id' atribuídos como 'data-volta'. Outros campos foram eliminados para simplificar, pois serão utilizados numa futura página de checkout.

Em seguida, o segundo 'fieldset' é abordado. Classes são adicionadas para os campos de 'checkbox'. A 'div' para cada 'checkbox' recebe a classe do Bootstrap 'form-check'. As classes 'form-check-input' e 'form-check-label' são

adicionadas ao 'input' e 'label', respectivamente. Os 'labels' recebem o atributo 'for' para identificar a qual campo se referem, e os 'inputs' recebem 'ids' correspondentes. Este procedimento é aplicado para todos os 'checkboxes'.

Por fim, a seção de 'forma de pagamento' é retirada, pois haverá uma página de checkout futuramente. A classe do Bootstrap 'btn' é adicionada ao botão, que por sua vez, estiliza o botão. A segunda classe 'btn-secondary' é adicionada ao botão, que atribui a cor cinza ao botão.

Com esses passos, o formulário de contato torna-se muito mais apresentável.

## Tabelas

A estilização de tabelas pode ser uma tarefa complexa, porém, o Bootstrap facilita este processo ao disponibilizar diversas classes pré-definidas que podem ser utilizadas. Assim, não é necessário criar um CSS personalizado para cada tabela. Existe uma variedade de classes disponíveis na documentação do Bootstrap, todas acessíveis através do campo de busca. Para o estudo de hoje, algumas dessas classes serão utilizadas em uma tabela já existente.

Analisando a tabela atual, nota-se que foi criado um CSS próprio para estilizá-la. Esse CSS atribui uma largura de 100% para a tabela, determina a cor das fontes das colunas e utiliza pseudo-classes para alternar a cor de fundo de cada linha. Agora, as classes do Bootstrap serão aplicadas para substituir esse CSS personalizado.

Primeiro, adiciona-se a classe 'table' no elemento table. Em seguida, define-se uma cor para a tabela utilizando a classe 'table-warning'. Para alternar as cores das linhas, utiliza-se a classe 'table-striped'. Adicionando a classe 'table-hover', quando o mouse é passado sobre as linhas da tabela, a cor dessas linhas é alterada, facilitando a navegação. A classe 'table-bordered' adiciona

uma borda sutil à tabela. E, para alterar a cor da borda para amarelo, utiliza-se 'border-warning'.



Uma vez utilizadas as classes para estilizar a tabela, outras classes podem ser aplicadas para estilizar linhas e colunas. Aplicando a classe 'table-secondary' no elemento thead, um background cinza é atribuído ao cabeçalho. Já a classe 'table-group-divider' no tbody cria uma linha divisória entre o cabeçalho e o corpo da tabela.

Para alinhar o texto ao centro, aplica-se a classe 'text-center' à section. Adicionalmente, para aumentar o espaço entre as seções, aplica-se a classe 'p' com um número que representa o tamanho do padding desejado.

Por fim, realiza-se um ajuste no rodapé, alinhando o texto ao centro com a classe 'Text Center' e adicionando uma quebra de linha antes do botão "Salvar". Com esses ajustes, o site apresenta uma aparência mais agradável e organizada.

Assim, conclui-se este módulo sobre a estilização de tabelas com Bootstrap. No próximo módulo, serão explorados tópicos avançados do Bootstrap para melhorar ainda mais a estética do site.

## Conteúdo Bônus

Se você deseja melhorar ainda mais suas habilidades e conhecimento sobre o Bootstrap, este conteúdo bônus vai te ajudar bastante. Procure na w3schools por Bootstrap 5. A w3schools é conhecida por ser referência em html, css, js, php, sql e também é quando o assunto é bootstrap. O site explica o framework de maneira clara e compreensível, tornando-o um recurso ideal tanto para iniciantes quanto para programadores experientes que desejam se aprofundar no Bootstrap.

## Referências Bibliográficas



**Bootstrap Versions.** Disponível em:

<[https://www.w3schools.com/bootstrap/bootstrap\\_ver.asp](https://www.w3schools.com/bootstrap/bootstrap_ver.asp)>. Acesso em 17 jul. 2023.

**Ir para exercício**