

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BA – IFBA

PROGRAMAÇÃO WEB – BOOTSTRAP
PROFESSOR: ADDSON ARAÚJO DA COSTA

VALENÇA - BA

Sumário

1. Introdução.....	4
2. Importar o Bootstrap.....	4
3. Grid Layout.....	5
→ container e container-fluid.....	5
→ row.....	6
→ col.....	6
→ Mesclando colunas.....	6
→ Alinhamento em linhas.....	7
→ Alinhamento em colunas.....	8
4. Componentes.....	9
→ Como usar um componente.....	9
→ Mais Componentes.....	10
5. Ícones.....	11
Exercício Proposto – Site IFBA.....	12

Lista de figuras

Figura 1: Resultado do Exemplo 2.....7

Figura 2: Documentacao Bootstrap.....10

Figura 3: Font Awesome.....11

1. Introdução

A internet foi criada desde 1969, mas o HTML tem sua história iniciada em 1991 enquanto o CSS foi desenvolvido em 1994. Já o Bootstrap só teve sua primeira versão em 2011 e seu nome originalmente era Twitter Bootstrap.

O que começamos a aprender na nossa apostila de HTML foi o padrão HTML 5 que foi publicado em 2000 e em nossa apostila de CSS iniciamos a aprender o padrão CSS 2.1 que tem seu padrão recomendado desde 2011. O que aprenderemos nessa apostila é a versão 5 do Bootstrap que foi publicada em junho de 2020 e é a última publicada.

O Bootstrap é um framework open source, isto é, de código aberto. Um framework é um conjunto de componentes prontos que podem ser utilizados e, uma vez incluídos no nosso projeto, o framework que assumirá o funcionamento do componente durante a utilização.

Está incluído no Bootstrap componentes prontos como, por exemplo, o carousel que é um componente capaz de mostrar slides de fotos, classes CSS que permitem posicionar objetos sem precisar digitar código CSS através de um sistema chamado de Grid Layout, classes para definir tamanhos, cores etc.

Entregando as classes e componentes prontos, o objetivo do Bootstrap é permitir desenvolvimento de páginas de internet mais bonitas digitando menos código e sem necessitar recorrer a digitar diretamente código CSS ou Javascript, tornando mais simples o desenvolvimento e aumentando também a experiência do usuário pois irá ter acesso a uma interface rica, otimizada em eficiência e compatibilidade entre browsers.

2. Importar o Bootstrap

Antes de importar o Bootstrap, vamos incluir uma linha ao início do nosso documento HTML, é o doctype. Apesar de parecer uma tag do HTML, ela não é. Trata-se de informar ao navegador que ele irá ler um código HTML e por não ser uma tag HTML, ela não fecha, é dessa forma mesmo e deve ser a primeira linha de uma página HTML. Logo em seguida o código pode iniciar normalmente com a tag HTML.

```
<!doctype html>
```

O primeiro passo para usar Bootstrap é carregá-lo em nosso HTML. Para fazer isso, precisamos apenas adicionar as linhas de código abaixo em nosso HTML. As tags abaixo devem ser inseridas em nossa tag HEAD.

```
<meta charset="utf-8">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet"/>
```

A primeira tag META é responsável por exibir corretamente os acentos digitados no código, enquanto a segunda tag META é responsável por gerar compatibilidade entre o tamanho da tela entre telefones, tablets e computadores.

Isto porque embora um telefone tenha a mesma quantidade de pixels que um computador, a tela é bem menor, assim, cria-se uma similaridade de tamanhos informando que não irá importar o

tamanho da tela do telefone em pixels e sim o tamanho real da tela e será iniciado a exibição sem uso de zoom.

Por último, a tag LINK é responsável por carregar todas as classes e elementos de CSS disponíveis no framework.

Já a tag abaixo deve ser inserida como sendo a última tag antes de fechar o body. Isto porque ela é responsável por dar movimento e interatividade em nosso site, precisando assim que os componentes tenham primeiro sido posicionados na página ou então poderia gerar o efeito de mover objetos ainda não carregados na tela, causando quebra de layout enquanto a página carrega.

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"></script>
```

Feito os passos anteriores, estamos prontos para usar o Bootstrap. Todos os códigos abaixo deverão ser incluídos dentro da tag BODY.

3. Grid Layout

Grid Layout é uma forma de organizar elementos estabelecendo linhas e colunas, como se fosse uma tabela, porém sem usar a tag “TABLE” do HTML. Ao invés disso, usaremos DIVs. A motivação por trás de não usar a tag TABLE foi explicada na apostila anterior e se resume a falta de compatibilidade entre navegadores diferentes. Já a motivação para usar o Grid Layout é que ele permite muito mais opções do que uma tabela seria capaz de suportar.

Para entendermos o sistema de grids do Bootstrap, iniciaremos conhecendo 3 primeiras classes que podem ser comparadas ao que era oferecido pela tabela no HTML. Ao invés de usar as tags TABLE, TR e TD, vamos usar, respectivamente, as classes “container”, “row”, “col”. Note a semelhança do código abaixo com uma tabela tradicional do HTML.

<div class="container border">	<!-- <TABLE>	-->
<div class="row">	<!-- <TR>	-->
<div class="col border"> 1 </div>	<!-- <TD> 1 </TD>	-->
<div class="col border"> 2 </div>	<!-- <TD> 2 </TD>	-->
</div>	<!-- </TR>	-->
<div class="row">	<!-- <TR>	-->
<div class="col border"> 3 </div>	<!-- <TD> 3 </TD>	-->
<div class="col border"> 4 </div>	<!-- <TD> 4 </TD>	-->
<div class="col border"> 5 </div>	<!-- <TD> 5 </TD>	-->
</div>	<!-- </TR>	-->
</div>	<!-- </TABLE>	-->

Exemplo 1: Containers, linhas e colunas

Teste o Exemplo 1 acima, note que foi adicionado no exemplo uma classe ainda não explicada nessa apostila que é a classe border. Como é de se imaginar, essa classe cria uma borda ao redor do elemento.

→ container e container-fluid

O container (do inglês, significa recipiente) é um espaço para armazenar um conteúdo. Ele deve ser usado para que sejam depositados elementos dentro dele, quaisquer que sejam os elementos.

Por padrão, ao usar a classe “container” diretamente, será criado um container centralizado e que se ajusta ao tamanho da tela, mas terá no máximo 1320 pixels. Já se for usado a classe “container-fluid”, ele irá tentar ocupar toda a largura da tela, sempre.

→ **row**

A classe “row” representa uma linha. Se dentro dela contiver elementos com a classe “col”, esses elementos serão colocados lado a lado como uma linha normal de tabela. Contudo os elementos que não tiverem a classe col, não terão esse destino.

Sinta-se livre para colocar uma linha dentro de uma coluna, não é preciso criar um novo container para isto. Além disso, é possível definir o número de colunas desejadas na linha para que, caso seja ultrapassado essa quantidade, as colunas restantes continuem de baixo como se fossem uma nova linha, evitando criar muitas classes de linhas.

Para definir o número de colunas máximo na linha, use na DIV que contém a classe “row”, também uma das classes entre “row-cols-1” até a classe “row-cols-12”. O Bootstrap limita a um máximo de 12 colunas por linha, mas lembre-se que aqui o objetivo não é criar uma tabela de conteúdo, é organizar o layout da página. Teste o Exemplo 2 abaixo e teste valores diferentes para o número após “row-cols-”, teste também com um número maior de colunas.

```
<div class="container">
  <div class="row row-cols-3">
    <div class="col border"> col </div>
    <div class="col border"> col </div>
    <div class="col border"> col </div>
    <div class="col border"> col </div>
  </div>
</div>
```

Exemplo 2: Definindo número de colunas por linha

A classe “col” irá dividir o espaço igualmente entre elas por padrão. Caso deseje desabilitar isto, podemos usar a classe “row-cols-auto”. Teste no Exemplo 2 substituir a classe “row-cols-3” por “row-cols-auto”.

→ **col**

A classe “col” representa uma coluna, ela serve para agrupar itens de forma semelhante a uma coluna de tabela e coloca automaticamente espaçamento para que as colunas não fiquem grudadas, mas isso é facilmente ajustado.

Uma diferença importante em usar a classe col, ao invés de usar uma coluna de uma tabela do HTML é que o número de colunas aqui é totalmente independente entre as linhas, assim, ao terminar uma linha, as demais não irão influenciar no layout, então não se preocupe com quantas colunas tem em cada linha.

→ **Mesclando colunas**

Note que o uso da classe “col” observado no Exemplo 1, irá distribuir o tamanho igualmente entre todas as colunas, mas algumas vezes desejamos que uma coluna seja maior que outra. E então, será necessário definir a largura das colunas.

Para explicar como ocorre isto, é preciso entender que o bootstrap trata a largura de colunas com base na idéia de mesclagem usada em tags table, conforme estudamos. Ele considera que cada linha tem um total fixo de 12 colunas e com base em quantas colunas mesclarmos, teremos a largura final.

Contudo ao usar a classe col, não estamos especificando a largura dessa coluna. Para especificar, precisamos usar a classe col-1, col-2, col-* onde o asterisco informa a largura em número de colunas que desejamos nesta célula.

```
<div class="row">
  <div class="col border"> col </div>
  <div class="col-6 border"> col-6 </div>
  <div class="col border"> col </div>
</div>
<div class="row">
  <div class="col border"> col </div>
  <div class="col-5 border"> col-5 </div>
  <div class="col border"> col </div>
</div>
```

Exemplo 3: Mesclando colunas

Na Figura 1 abaixo podemos ver o resultado do código presente no Exemplo 3. Por fins didáticos, na Figura 1 foi adicionado uma linha extra com 12 (doze) colunas uniformemente distribuídas usando a classe “col” numeradas de 1 (um) a 12 (doze). Um detalhe importante é que o número de colunas em uma linha não influencia nas demais, então ela foi incluída apenas para efeito demonstrativo e, por isso, está colorida diferente.

1	2	3	4	5	6	7	8	9	10	11	12
col			col-6						col		
col				col-5					col		

Figura 1: Resultado do Exemplo 2

Após a interpretação do exemplo deve-se ter notado que o uso da classe “col-6” significa que ela ocupará o espaço de 6 colunas, uma vez que ela ocupa o espaço das colunas 4 a 9, enquanto o uso da classe “col-5” significa que ela ocupará o espaço de 5 colunas, uma vez que ela ocupa quase o mesmo espaço que a col-6, mas metade a menos do espaço da coluna 4 e metade a menos do espaço da coluna 9.

O espaço que sobrou antes e após as colunas mescladas é uniformemente distribuído entre os elementos de classe “col”. Essas classes existem desde “col” até “col-12” que é o número máximo de colunas que podem ser mescladas no Grid Layout.

Caso a quantidade de colunas sendo mesclada supere o número máximo de 12 (doze) colunas, então as colunas que não couberam vão para baixo iniciando uma nova linha, ao invés de sumir da tela.

→ Alinhamento em linhas

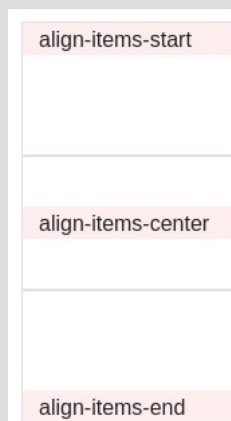
Pode-se, junto dos elementos que têm classe “row”, definir o alinhamento vertical da linha usando as classes align-items-start, align-items-center e align-items-end, para que seja feito o alinhamento vertical, respectivamente, no topo, centro ou em baixo.

Teste o Exemplo 4 abaixo. O código usado na tag style no Exemplo 4 deve ser inserida na tag head enquanto as divs devem ser incluídas na tag body. Ao lado do exemplo tem uma pré-visualização do resultado que deve ser obtido.

```

<style>
  .row { height: 100px; background-color: #ffeeee; }
  .container { width: 160px !important; }
</style>
<div class="container">
  <div class="row align-items-start border">
    <div class="col"> align-items-start </div>
  </div>
  <div class="row align-items-center border">
    <div class="col"> align-items-center </div>
  </div>
  <div class="row align-items-end border">
    <div class="col"> align-items-end </div>
  </div>
</div>

```



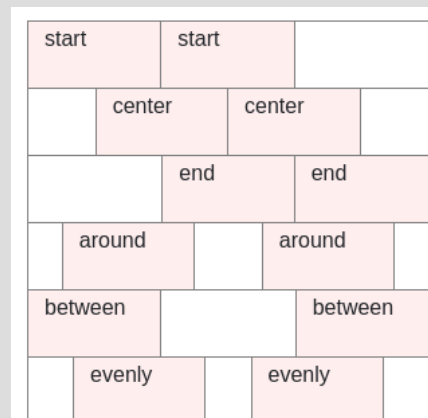
Exemplo 4: Alinhamento vertical

Além do alinhamento vertical, ainda sobre as linhas, é possível realizar um alinhamento horizontal. Para isso, deve-se usar junto com a classe “row” a classe “justify-content-start”, “justify-content-center”, “justify-content-end”, “justify-content-around”, “justify-content-between” ou “justify-content-evenly”, que significam justificar o conteúdo no início, centro, fim da linha e ao redor, no meio ou igualmente entre colunas. Teste o Exemplo 5 substituindo a classe “justify-content-start” para testar as opções.

```

<style>
  .row {
    height: 50px;
    border: solid 1px gray;
  }
  .col-4 {
    background-color: #ffeeee;
    border-left: solid 1px gray;
    border-right: solid 1px gray;
  }
  .container {
    width: 300px !important;
  }
</style>
<div class="container">
  <div class="row justify-content-start">
    <div class="col-4"> start </div>
    <div class="col-4"> start </div>
  </div>
</div>

```



Exemplo 5: Alinhamento Horizontal

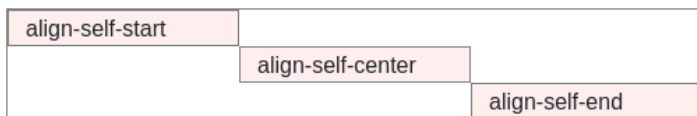
→ Alinhamento em colunas

Para alinhamento vertical em colunas, podemos usar junto com a classe col, uma entre as classes “align-self-start”, “align-self-center” ou “align-self-end” para alinhar a coluna individualmente no início, no centro ou final da linha.


```

<style>
  .row { height: 80px; }
  .col { background-color: #ffeeee; }
  .container { width: 500px; }
</style>
<div class="container">
  <div class="row border">
    <div class="col border align-self-start"> align-self-start </div>
    <div class="col border align-self-center"> align-self-center </div>
    <div class="col border align-self-end"> align-self-end </div>
  </div>
</div>

```



Exemplo 6: Alinhamento vertical de colunas

4. Componentes

Além do sistema de layout em grid, o Bootstrap entrega muito mais classes prontas desde alterar cores de texto, background, tamanhos, margens etc, mas por enquanto vamos olhar alguns componentes sofisticados que podem trazer uma riqueza visual as nossas páginas.

→ Como usar um componente

O componente abaixo é o accordion e ele mostra apenas um conteúdo de cada vez, ocultando os demais. Veja o Exemplo 7 abaixo.

```

<div class="accordion" id="accordionExample">
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingOne">
      <button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne"> Item #1 </button>
    </h2>
    <div id="collapseOne" class="accordion-collapse collapse show" aria-labelledby="headingOne" data-bs-parent="#accordionExample">
      <div class="accordion-body"> Conteudo item 1 </div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingTwo">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo"> Item #2 </button>
    </h2>
    <div id="collapseTwo" class="accordion-collapse collapse" aria-labelledby="headingTwo" data-bs-parent="#accordionExample">
      <div class="accordion-body"> Conteudo item 2 </div>
    </div>
  </div>
</div>

```

Exemplo 7: Accordion

Nele temos uma classe accordion na primeira linha do código que representa o componente, dentro do componente temos duas divs destacadas em amarelo com classe accordion-item, que representa um item do accordion (apenas um será mostrado de cada vez) e dentro desses items está destacado em laranja as partes que devemos editar para adaptar nosso conteúdo.

O nome das classes tem significado, não perca de traduzir para entender seu significado sempre que não souber. Além disso, perceba no código acima que o ID definido em um elemento, é referenciado em trechos do código, veja as marcações em azul. Assim, ao alterar o ID de um elemento, é preciso alterar nos demais locais onde esse ID é usado, tente entender o padrão formado com os IDs.

Note que poderíamos ter mais items em nosso accordion, nesse caso aumentando o número de divs accordion-item, mas note que ela tem uma sintaxe que precisa ser seguida.

Veja também o componente funcionando e mais opções de uso do componente no site do bootstrap <<https://getbootstrap.com/docs/5.0/components/accordion/>>. O Bootstrap entrega o código do componente pronto e mostra um exemplo do componente funcionando diretamente em sua documentação no link acima, não perca de verificar.

→ Mais Componentes

O Bootstrap nos fornece os componentes: Alerts, Badge, Breadcrumb, Buttons, Button group, Card, Carousel, Close button, Collapse, Dropdowns, List group, Modal, Navs & tabs, Navbar, Pagination, Popovers, Progress, Scrollspy, Spinners, Toasts e Tooltips.

Mostrar um a um aqui na apostila não caberia uma apresentação interessante pois não daria para ver o componente funcionando e ter seu uso testado, mas não se preocupe, pois, na documentação do Bootstrap <<https://getbootstrap.com/docs/5.0/components/accordion/>>, podemos ver todos os componentes em funcionamento e também o código pronto do elemento já com o botão de copiar o código junto ao componente.

Ao lado, na Figura 2, vemos um exemplo de como é montado a documentação do Bootstrap.

Primeiro vemos o resultado gerado com o componente e abaixo o código usado para gerar a visualização exibida acima.

Junto ao código, temos um botão para copiar o código exemplo, destacado em azul.

O componente mostrado na Figura 2 são Alerts, eles são usados para exibir mensagens na tela para o usuário e podem ser coloridos de acordo com a classe usada.

As classes são de acordo com o tipo de notícia, alertas primários, secundários, de perigo e etc.

Teste todos os componentes do Bootstrap para entender seus potenciais.

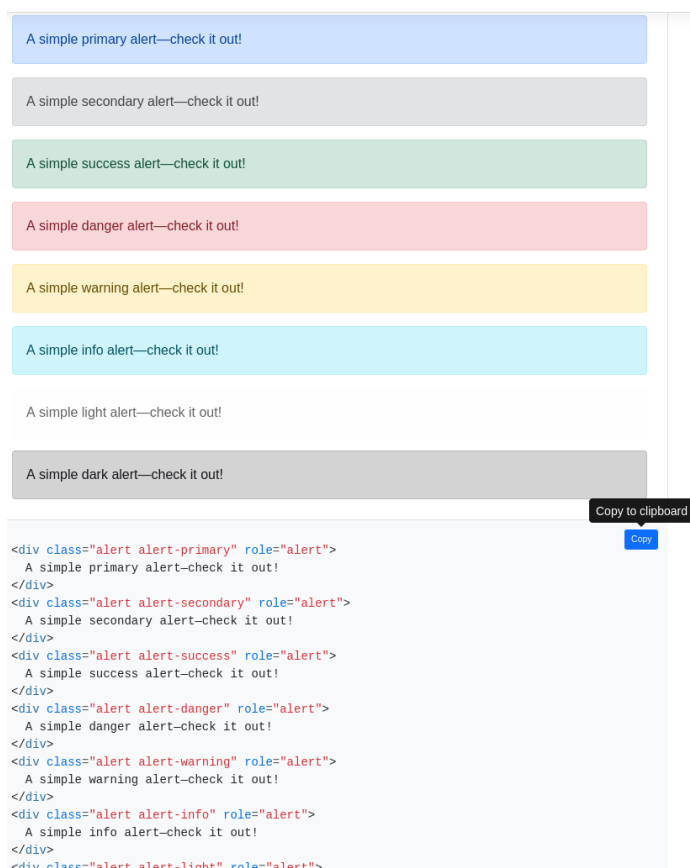


Figura 2: Documentacao Bootstrap

5. Ícones

Outro aspecto interessante no Bootstrap é que ele fornece ícones para nossos websites. Ao todo ele fornece mais de 1300 ícones open source. Para usá-los, basta acessar <https://icons.getbootstrap.com/>, escolher o ícone que quer usar e ele fornecerá o código em HTML ao lado para que seja copiado para seu código. Opte pelo código com a Tag SVG.

```
<svg width="16" height="16" fill="currentColor" class="bi bi-list" viewBox="0 0 16 16" xmlns="http://www.w3.org/2000/svg">
  <path fill-rule="evenodd" d="M2.5 11.5A.5.5 0 0 1 3 11h10a.5.5 0 0 1 0 1H3a.5.5 0 0 1-.5-.5zm0-4A.5.5 0 0 1 3 7h10a.5.5 0 0 1 0 1H3a.5.5 0 0 1-.5-.5zm0-4A.5.5 0 0 1 3 3h10a.5.5 0 0 1 0 1H3a.5.5 0 0 1-.5-.5z"/>
</svg>
```

Exemplo 8: Ícones

O código do Exemplo 8 é um ícone de um Menu Hamburguer, são três barrinhas paralelas que simbolizam uma lista de itens. Note que o tamanho e cor do ícone podem ser definidos no início da tag SVG nos atributos width, height e fill onde os tamanhos devem ser em pixels e a cor pode ser da mesma forma que definido em CSS.

Uma alternativa aos ícones do Bootstrap, é o uso dos ícones do Font Awesome. O Font Awesome oferece mais de 1600 ícones gratuitamente e 7800 ícones na sua versão paga, é uma das melhores fontes de ícones.

Para usar o Font Awesome, basta incluir a linha de código abaixo para importar o pacote de ícones. Por se tratar de importar arquivos de fontes, ele deve ser colocado dentro da tag head. O Font Awesome limita o uso gratuito a sites com até 10 mil visitas por mês, assim se seu site for desenvolvido para um cliente, vale a pena comprar o pacote de ícones, mas para uso durante o curso, pode ser usado a versão gratuita.

Para incluir o uso do font awesome no seu site, é preciso se cadastrar, fazer login e na aba “Your Kits” é preciso criar um kit para usar. Após criar o kit, será dado pelo font awesome uma linha de código semelhante ao Exemplo 9, para ser anexado ao seu site. Essa linha de código deve ser incluída ao final do body como uma das últimas linhas antes de fechar o body.

```
<script src="https://kit.fontawesome.com/99631d37a6.js" crossorigin="anonymous"></script>
```

Exemplo 9: Font Awesome

De forma semelhante aos ícones disponíveis no Bootstrap, ao acessar <https://fontawesome.com/icons/>, teremos a relação de todos os ícones disponíveis no FontAwesome e ao clicar em um dos ícones, teremos a exibição mostrada na Figura 3 onde, ao clicar sobre um trecho de código HTML localizado acima do ícone, será copiado o código HTML necessário para incluir o ícone em sua página HTML.

Para personalizar cor e tamanho do ícone, os ícones vindos do Bootstrap e FontAwesome são

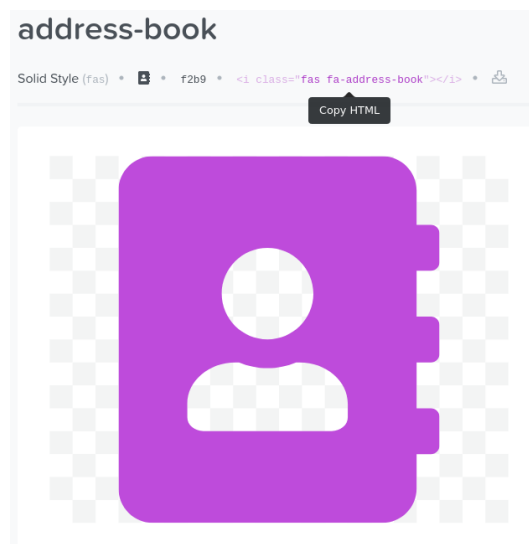


Figura 3: Font Awesome

considerados pelo HTML como sendo texto, assim, se aumentarmos o tamanho da fonte no ícone ou alterarmos a cor do texto, eles obedecerão.