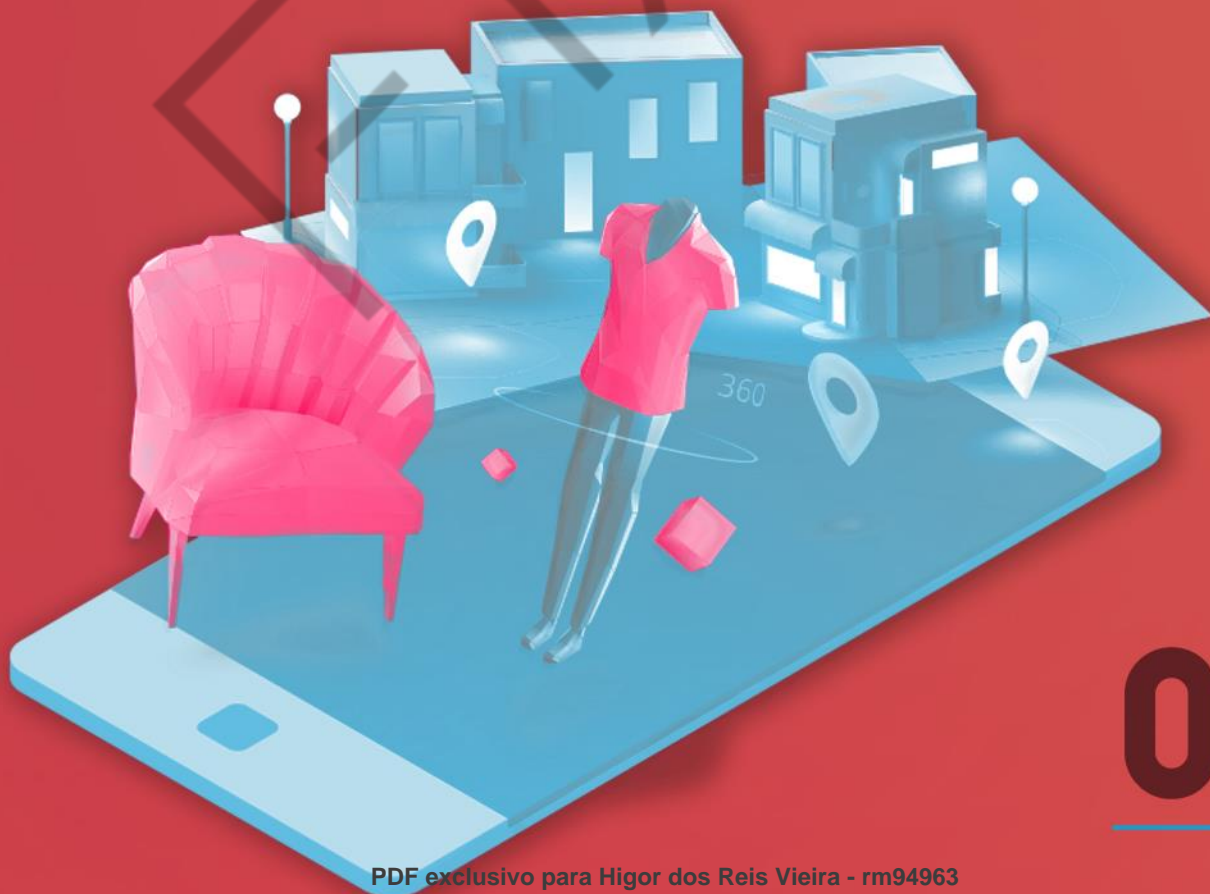


VIEW

MUITO ESTILO SEM SOFRIMENTO

ALEXANDRE CARLOS DE JESUS



05

LISTA DE FIGURAS

Figura 1 – Bootstrap	6
Figura 2 – Exemplo de responsividade do Bootstrap	8
Figura 3 – Tela de demonstração da IDE Sublime Text em seu site.....	9
Figura 4 – Tela de início da página do Bootstrap	10
Figura 5 – Tela de “Get started” do Bootstrap	10
Figura 6 – Tela de template do Bootstrap	11
Figura 7 – Tela de template no Sublime Text.....	12
Figura 8 – Estrutura HTML do arquivo de template do Bootstrap	13
Figura 9 – Exemplo 1 de página com Bootstrap utilizando referência por CDN.....	17
Figura 10 – Exemplo 2 de página com Bootstrap utilizando referência por CDN.....	18
Figura 11 – Tela dos servidores CDN do Bootstrap	19
Figura 12 – Resultado do exemplo no PC com largura de 375 x 707	22
Figura 13 – Resultado do exemplo no celular com largura de 375 x 812.....	23
Figura 14 – Exemplo das cores de estados do Bootstrap	24
Figura 15 – Exemplo do sistema de grid do Bootstrap.....	24
Figura 16 – Exemplo 2 do sistema de grid do Bootstrap.....	25
Figura 17 – Exemplo 3 do sistema de grid do Bootstrap com um layout.....	25
Figura 18 – Exemplo de container fixed	28
Figura 19 – Exemplo de container fluid.....	29
Figura 20 – Exemplo de linha (<i>row</i>).....	31
Figura 21 – Exemplo de coluna (<i>columns</i>)	34
Figura 22 – Exemplo de página com o grid completo utilizando fixed layout	37
Figura 23 – Exemplo de página com o grid completo utilizando fluid layout	39
Figura 24 – Exemplo de página com o grid completo utilizando fluid layout e redimensionada.....	40
Figura 25 – Exemplo de página com o grid completo utilizando fluid layout sendo apresentada na tela de um celular	40
Figura 26 – Exemplo de página com um formulário inline.....	42
Figura 27 – Exemplo de página com um formulário horizontal.....	44
Figura 28 – Exemplo de página com um formulário inline redimensionado que se tornou um form vertical.....	45
Figura 29 – Exemplo de página com um formulário horizontal redimensionado que se tornou um form vertical.....	46
Figura 30 – Exemplo de página com um formulário padrão ou vertical.....	47
Figura 31 – Lista de ícones do Bootstrap	49
Figura 32 – Exemplo de formulário com alert e ícone	51
Figura 33 – Ícone que foi aplicado no exemplo	51
Figura 34 – Formulário com iconografia sendo aplicada.....	53
Figura 35 – Collapse + Card - Conteúdo contraído	54
Figura 36 – Collapse + Card - Conteúdo expandido	55
Figura 37 – Tipos de alertas.....	56
Figura 38 – Badges – ênfase em elementos de span	57
Figura 39 – Cards personalizados.....	60

CÓDIGOS-FONTE

Código-fonte 1 – Exemplo de metatag	14
Código-fonte 2 – Exemplo de atributos para metatag	14
Código-fonte 3 – Exemplo do atributo viewport.....	14
Código-fonte 4 – Exemplo da metatag obrigatória para IE.....	16
Código-fonte 5 – Exemplo de referência para arquivo do Bootstrap	16
Código-fonte 6 – Exemplo de referência para arquivos JS do Bootstrap	16
Código-fonte 7 – Exemplo de referência para um servidor CDN do Bootstrap	18
Código-fonte 8 – Exemplo de arquivo HTML5.....	19
Código-fonte 9 – Página HTML5 com Bootstrap comentada	22
Código-fonte 10 – Página HTML5 com um container fixed	27
Código-fonte 11 – Página HTML5 com um container fluid	29
Código-fonte 12 – Página HTML5 com um container fluid	31
Código-fonte 13 – Página HTML5 com exemplo de coluna	33
Código-fonte 14 – Página HTML5 de exemplo completo com o grid e layout fixed ..	35
Código-fonte 15 – Arquivo styleless.css	36
Código-fonte 16 – Página HTML5 de exemplo completo com o grid e layout fluid ...	38
Código-fonte 17 – Arquivo styles.css	39
Código-fonte 18 – Formulário inline	42
Código-fonte 19 – Formulário horizontal	44
Código-fonte 20 – Formulário padrão ou vertical	47
Código-fonte 21 – Apresentação de um ícone de exclamação e a classe de alert ...	50
Código-fonte 22 – Formulário com a iconografia aplicada aos elementos	53
Código-fonte 23 – Painel Collapse	54
Código-fonte 24 – Tipos de alertas	56
Código-fonte 25 – Badges – ênfase em elementos de span	57
Código-fonte 26 – Cards personalizados	59

SUMÁRIO

1 MUITO ESTILO, SEM SOFRIMENTO	5
1.1 INTRODUÇÃO	5
1.2 SOBRE O BOOTSTRAP	5
1.3 HISTÓRIA	7
1.4 Começando com Bootstrap	8
1.4.1 Integrated Developer Enviroment – IDE	8
1.4.2 Obtendo o Bootstrap	10
1.4.3 Estrutura do template	12
1.4.4 Metatags.....	14
1.4.5 Referência de arquivos.....	16
1.4.6 Referência de arquivos remotos – CDN	17
1.4.7 HTML5.....	19
2 CRIANDO TEMPLATES.....	21
2.1 Boilerplate	21
2.2 Estados	23
2.3 Sistema De Grid	24
2.3.1 Container.....	26
2.3.2 Linhas (<i>rows</i>) e colunas (<i>columns</i>).....	30
3 FORMULÁRIOS	41
3.1 Form Inline	41
3.2 Form horizontal	43
3.3 Form vertical padrão	45
3.4 MISC	48
3.4.1 Iconografia.....	48
REFERÊNCIAS.....	61
GLOSSÁRIO	62

1 MUITO ESTILO, SEM SOFRIMENTO

1.1 INTRODUÇÃO

A linguagem CSS nos abre infinitas possibilidades para a construção de layouts atraentes. No entanto, criá-los é uma tarefa que exige muito esforço e tempo. Existe um atalho chamado Bootstrap, com o qual podemos aplicar os estilos CSS sem muito esforço. Conheça mais sobre esse atalho neste capítulo, pois isso pode beneficiar muito o Health Track!

1.2 SOBRE O BOOTSTRAP

Bootstrap é um dos frameworks para desenvolvimento web front-end mais populares do mundo. Dotado de diversos elementos e funções personalizáveis, o que mais chama a atenção é o fato de ele possibilitar a responsividade total dos projetos nos quais é aplicado, seja para um pequeno celular, seja para uma TV de 40 polegadas.

O conceito de *Mobile First* está presente neste framework, ou seja, pense sempre em desenvolver para dispositivos móveis primeiro. Quando você utiliza esse framework, ele sempre vai preparar o conteúdo inicialmente para equipamentos mobile. Caso o dispositivo que esteja carregando a página não seja mobile, ele passa para o dispositivo seguinte, um *tablet* e assim por diante até que o *viewport* seja o correto.



Figura 1 – Bootstrap
Fonte: Shutterstock (2017)

Mostrando toda a integração entre HTML, CSS e Javascript, esse framework é constantemente atualizado pela comunidade de desenvolvedores, na página do projeto no github, em <<https://getbootstrap.com/>>. Qualquer um pode contribuir para o projeto, que foi inicialmente desenvolvido em 2011 pela equipe do Twitter a fim de solucionar problemas pontuais. Com o tempo, ganhou corpo e tomou proporções que nem os próprios desenvolvedores imaginavam.

Em 2013, foi lançada a versão 3 com algumas novidades, principalmente em se tratando dos plug-ins de JQuery.

Atualmente, o Bootstrap está estável na versão 4, onde ocorreu uma grande reformulação nos padrões de arquitetura e uso dos componentes em relação à versão 3. Próximo do lançamento da versão 5, que ampliou o poder da versão anterior, mas sem trazer muitas alterações que impactem em mudanças que quebrem a compatibilidade com a versão anterior. Com a versão 5, o Bootstrap traz novamente uma biblioteca de ícones pronta para uso, dessa vez, feita e mantida pela própria equipe.

1.3 HISTÓRIA

Em 2011, o Bootstrap foi criado como uma solução interna para resolver inconsistências de desenvolvimento dentro da equipe de engenharia do Twitter. Basicamente, não havia uma definição de estrutura de código na forma escolhida pelos engenheiros do Twitter para desenvolver a plataforma.

O desenvolvimento e a engenharia da web são um trabalho especializado (muitos diriam que são uma arte), e cada engenheiro tem a sua própria maneira de codificar. Isso funciona em alguns casos, mas quando há vários engenheiros trabalhando no mesmo projeto com abordagens de codificação ligeiramente diferentes, as inconsistências são inevitáveis. Com o tempo, as inconsistências de engenharia da web podem se transformar em problemas de codificação sérios que criam incerteza e aumentam o tempo de manutenção. O Bootstrap foi uma ferramenta desenvolvida originalmente por Mark Otto e Jacob Thorton, então engenheiros do Twitter, como uma tentativa de incentivar o uso de uma única estrutura pela equipe de engenharia do Twitter, reduzindo essas inconsistências. A iniciativa do Bootstrap certamente foi bem-sucedida no Twitter, permitindo que toda a equipe trabalhasse com maior rapidez e eficiência e menos inconsistências.

Embora fosse inicialmente uma solução interna no Twitter, Mark e Jacob rapidamente perceberam que tinham descoberto algo muito maior. Em agosto de 2011, a estrutura Bootstrap foi lançada como um projeto de software livre no Github. Em alguns meses, milhares de desenvolvedores de todo o mundo contribuíram com o código e o Bootstrap se tornou o projeto de desenvolvimento de software livre mais ativo do mundo. Desde então, tornou-se cada vez mais conhecido e se transformou na “estrutura front-end mais popular para o desenvolvimento inicial de projetos móveis com capacidade de resposta na web”. (UTTERBACK, 2014, parte 1 de 2)



Figura 2 – Exemplo de responsividade do Bootstrap
Fonte: Prestashop (2014), adaptado por FIAP (2017)

1.4 Começando com Bootstrap

Para iniciarmos os estudos com o Bootstrap, são necessários alguns passos para que tudo esteja em ordem ao iniciar o desenvolvimento.

Necessitamos escolher uma IDE que nos agrade, configurar o ambiente e realizar o download das bibliotecas do framework. Além disso, é preciso criar as pastas e conhecer cada um dos arquivos que estão contidos neste framework.

Então vamos nessa, galera!

1.4.1 Integrated Developer Enviroment – IDE

IDE é a abreviação de *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado. É o programa utilizado para o desenvolvimento dos projetos.

Existem diversos tipos de IDE, dos mais simples, como NOTEPAD, até os mais avançados e completos, por exemplo, Eclipse, Visual Studio.

A escolha do IDE é muito importante pelo simples fato de que esses programas podem simplesmente dar um ganho estratosférico no desenvolvimento, como também podem ser a pior experiência na vida de um desenvolvedor. Já vi casos de empresas que escolheram mal o IDE para determinado projeto e perderem meses no prazo final, simplesmente pelo fato de que o time não se adaptou àquela ferramenta. Então escolha bem o seu, meu amigo.

Aqui, como estamos trabalhando apenas com codificação WEB, não teremos a interação de servidores e outras linguagens de scripts, podemos lançar mão de uma ferramenta mais modesta e ágil, o que acham?

Poderíamos utilizar o NOTEPAD do Windows facilmente, mas com certeza nossa produtividade não seria lá essas coisas. Apesar de o Notepad editar os arquivos HTML, CSS e JS; ele não dá suporte a algumas facilidades, por exemplo, a indentação do código para que fique inteligível.

Então vamos utilizar o Sublime Text como ferramenta de desenvolvimento. Uma ferramenta leve e existente para todas as plataformas, Windows, Linux e Mac.

Para realizar o download da ferramenta, basta acessar <https://www.sublimetext.com/>.

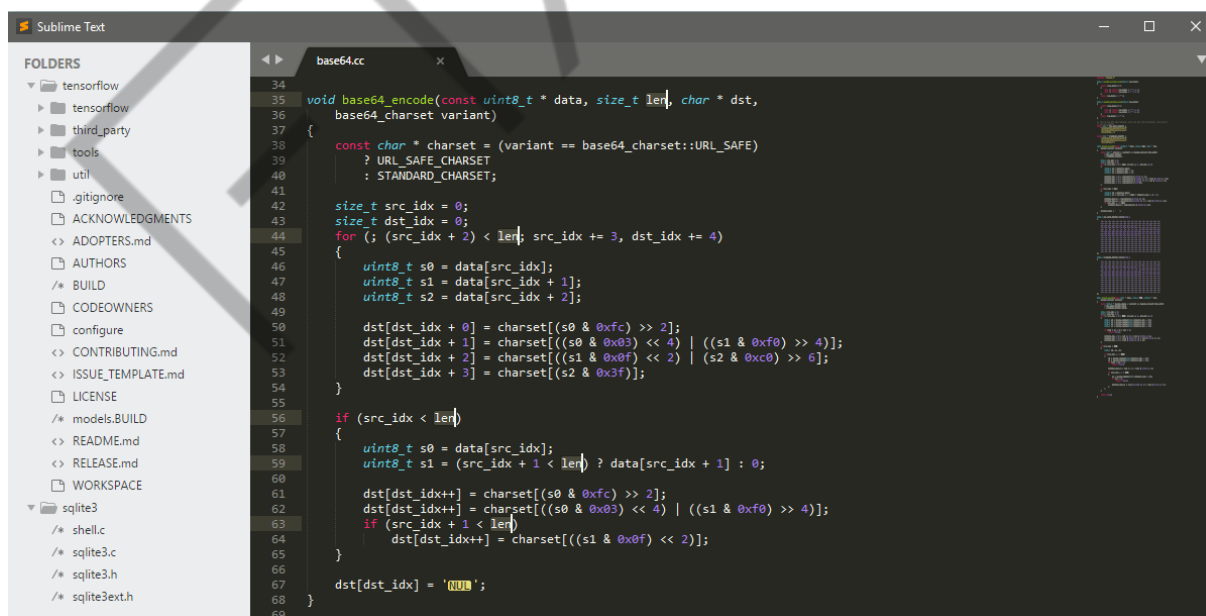


Figura 3 – Tela de demonstração da IDE Sublime Text em seu site
Fonte: Elaborado pelo autor (2021)

1.4.2 Obtendo o Bootstrap

O Bootstrap é bem fácil de obter, basta acessar o site do projeto (getbootstrap.com) e fazer uso dos arquivos do framework.

Ao acessar a página do Bootstrap, a seguinte tela vai aparecer para você.

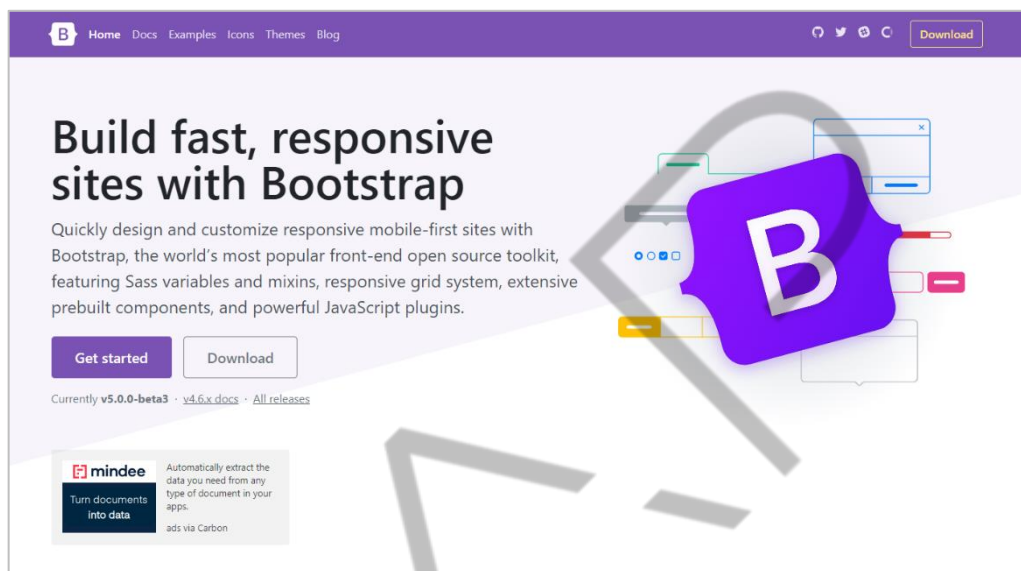


Figura 4 – Tela de início da página do Bootstrap
Fonte: Elaborado pelo autor (2021)

Basta clicar no botão “Get started”, para ser enviado à área de tutorial e documentação do toolkit.

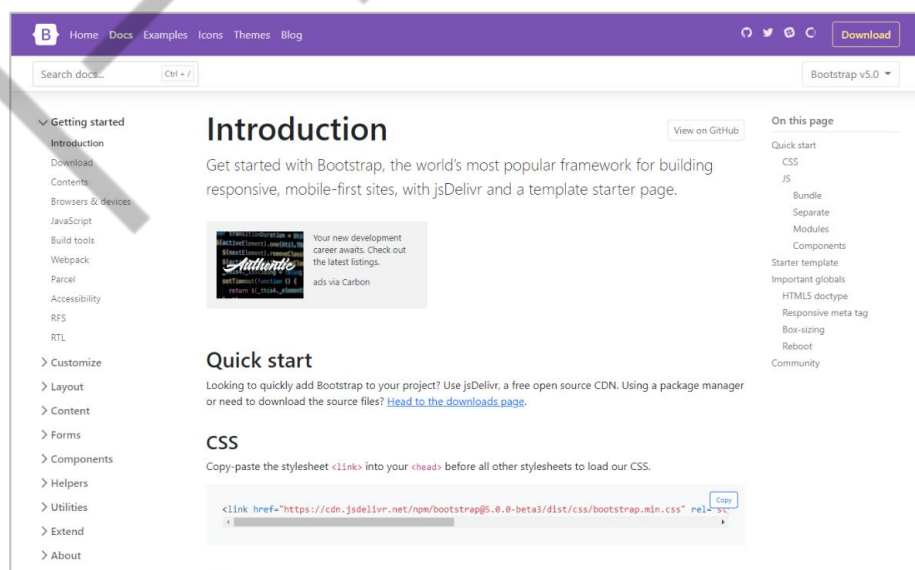


Figura 5 – Tela de “Get started” do Bootstrap
Fonte: Elaborado pelo autor (2021)

Nessa seção, existem algumas possíveis práticas para iniciarmos o nosso projeto que utilizará o Bootstrap. Usaremos a abordagem em que tanto o arquivo CSS quanto o JS responsáveis pela “mágica” do Bootstrap estarão disponíveis na nuvem, em um sistema conhecido como CDN (*Content Delivery Network* – rede de distribuição de conteúdo). Rolando a página, ou acessando o link <https://getbootstrap.com/docs/5.0/getting-started/introduction/#starter-template> , encontramos o *template* (modelo) do documento HTML inicial (index.html) que utilizaremos como base para começar os nossos projetos em Bootstrap.

Copie o conteúdo do *template*, selecionando-o todo ou utilizando o botão “Copy”.

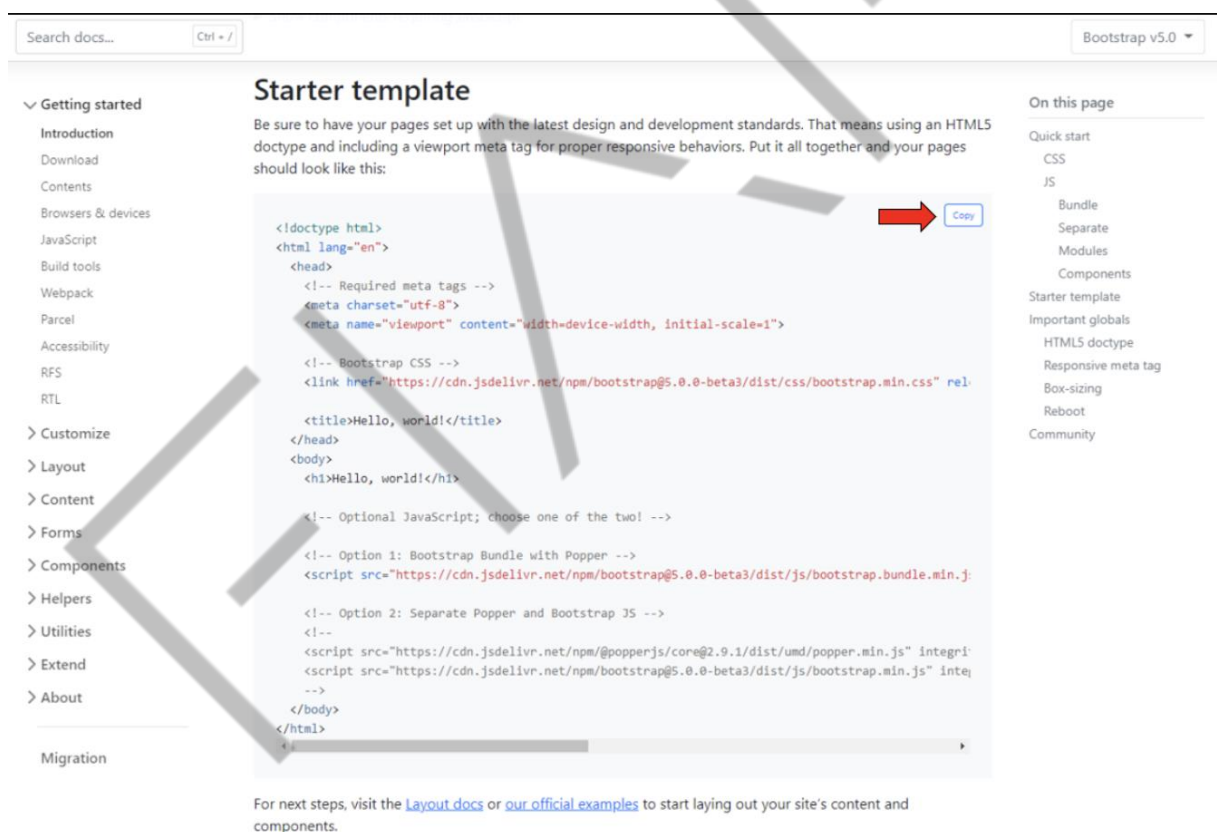
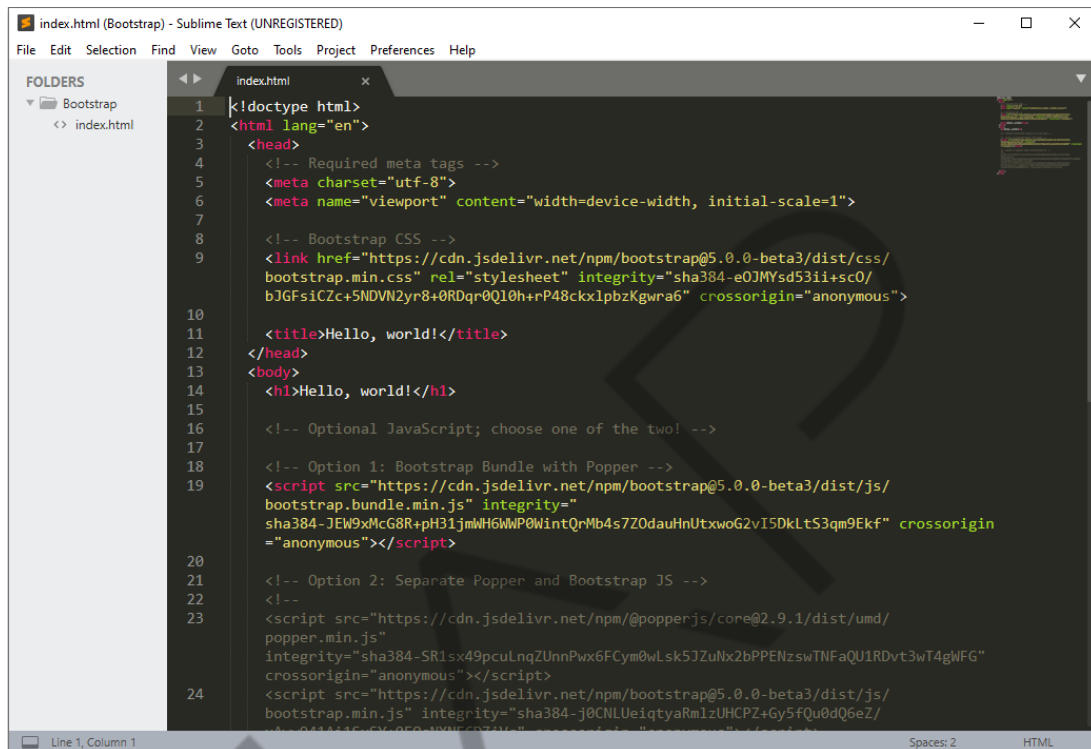


Figura 6 – Tela de template do Bootstrap
Fonte: Elaborado pelo autor (2021)

No Sublime Text, selecione a opção “File” e, logo após, em “Open Folder...”, crie ou escolha uma pasta que será utilizada como diretório raiz do projeto. Idealmente, para novos projetos, escolha uma pasta vazia. Com a pasta aberta no Sublime Text, a IDE interpretará o diretório como sendo um projeto. Clique com o

botão direito do mouse no nome do diretório, que aparece, por padrão, no canto superior esquerdo. No menu de contexto que aparecerá, selecione a opção “New File” e, neste novo arquivo que foi aberto, cole o conteúdo do *template*.



```
index.html (Bootstrap) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
  Bootstrap
    <> index.html

1 <!doctype html>
2 <html lang="en">
3 <head>
4   <!-- Required meta tags -->
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7
8   <!-- Bootstrap CSS -->
9   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/
bootstrap.min.css" rel="stylesheet" integrity="sha384-e0JMYsd53ii+sc0/
bJGfSiCZc+5NDVN2yr8+0RDqr0Q10h+rpP48ckxlpbzKgwra6" crossorigin="anonymous">
10
11   <title>Hello, world!</title>
12 </head>
13 <body>
14   <h1>Hello, world!</h1>
15
16   <!-- Optional JavaScript; choose one of the two! -->
17
18   <!-- Option 1: Bootstrap Bundle with Popper -->
19   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/
bootstrap.bundle.min.js" integrity="
sha384-JEW9xMcG8R+pH31jmWH6WP0WintQrMb4s7Z0dauHnUtxwoG2vI5DkLtS3qm9Ekf" crossorigin
="anonymous"></script>
20
21   <!-- Option 2: Separate Popper and Bootstrap JS -->
22   <!--
23   <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/
popper.min.js"
integrity="sha384-SR1sx49pcuLnqZUnnnPx68FwCm0uLsk5J3U2bPPENzswTNFaQU1RDvt3wT4gWFG"
crossorigin="anonymous"></script>
24   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/
bootstrap.min.js" integrity="sha384-j0CNLUeiqtYarmlzUHCPZ+Gy5fQu0dQ6eZ/
d...></script>
```

Figura 7 – Tela de template no Sublime Text
Fonte: Elaborado pelo autor (2021)

Salve o arquivo como “index.html” no diretório raiz do projeto.

1.4.3 Estrutura do template

Vamos analisar a estrutura do *template* que baixamos do Bootstrap.

```

<!doctype html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-e0JMYsd53ii+sc0/b7GFsiCZc+5NDVN2yr8+0RDQr0Q10h+rP48ckxlpbKgwra6" crossorigin="anonymous">

  <title>Hello, world!</title>
</head>
<body>
  <h1>Hello, world!</h1>

  <!-- Optional JavaScript; choose one of the two! -->

  <!-- Option 1: Bootstrap Bundle with Popper -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JEW9xMcG8R+pH31jmH6WHP00wintQrMb4s7Z0dauHnUtxwoG2vI5DkLts3q99Ekf" crossorigin="anonymous"></script>

  <!-- Option 2: Separate Popper and Bootstrap JS -->
  <!--
  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"
  integrity="sha384-SR1sx49pcuLnqZUnnPpx6GFCym0wLsk5JZuNx2bPPENzswTNFaQU1RDvt3wT4gWFG" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.min.js"
  integrity="sha384-j0CNLUeiqtYarmlzUHCPZ+Gy5fQu0dQ6eZ/xAww941Ai1SxSY+0EQqNI0EDziVc" crossorigin="anonymous"></script>
  -->
</body>
</html>

```

Figura 8 – Estrutura HTML do arquivo de template do Bootstrap
 Fonte: Elaborado pelo autor (2021)

Nesse modelo, além da estrutura padrão do HTML5, vemos a presença de outros elementos, como as tags META, LINK e SCRIPT.

A tag LINK está vinculando o arquivo CSS, arquivo de folha de estilos, ao HTML, esse CSS é o responsável por aplicar todos os estilos visuais predefinidos pelo Bootstrap.

Já a tag SCRIPT incorpora o código-fonte JavaScript do Bootstrap que é responsável por definir comportamentos mais elaborados para componentes inteligentes que foram criados pela equipe do Bootstrap.

O que chama a atenção são os arquivos *.css e *.js, conhecidos como bootstrap.min.css e bootstrap.bundle.min.js, mas o que isso quer dizer?

Arquivos com a partícula .min são otimizados para a melhor distribuição online, com tamanho reduzido e com apenas as informações necessárias, favorece uma rápida baixa do servidor e melhor execução nos navegadores.

O *template* traz duas opções de uso dos scripts, na primeira, já ativada (sem estar comentada), é a versão “empacotada” (bundle) de todas as ferramentas de JavaScript que serão utilizadas pelo Bootstrap. Já a segunda opção, a desativada (comentada) traz separadamente cada um dos scripts contidos no pacote. Não há uma diferenciação ou melhor opção. As duas realizam o mesmo processo de possibilitar componentes inteligentes do Bootstrap.

1.4.4 Metatags

Metatags são tags HTML que, posicionadas no cabeçalho do arquivo, podem determinar valores que vão influenciar desde a codificação dos caracteres apresentados até o formato como sua página será apresentada pelos navegadores.

Essa Metatag deve ser posicionada no cabeçalho do documento, entre as tags HEAD.

```
<head>
  <meta name="description" content="EXEMPLO - BOOTSTRAP">
</head>
```

Código-fonte 1 – Exemplo de metatag
Fonte: Elaborado pelo autor (2017)

Acima, podemos ver a sintaxe da metatag. O atributo name recebe o atributo que será carregado, o atributo content recebe o valor do atributo que foi especificado anteriormente.

Existem diversos atributos que podem ser atribuídos à metatag.

```
<head>
  <meta name="description" content="EXEMPLO - BOOTSTRAP">
  <meta name="keywords" content="HTML, CSS, XHTML, JavaScript">
  <meta name="author" content="Jorginho">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
</head>
```

Código-fonte 2 – Exemplo de atributos para metatag
Fonte: Elaborado pelo autor (2017)

Mas sempre tem um “mas”, vamos analisar um em especial.

```
<head>
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
</head>
```

Código-fonte 3 – Exemplo do atributo viewport
Fonte: Elaborado pelo autor (2017)

Meu amigo, vamos analisar a importância desse atributo para a utilização do nosso framework, ou não. Digo isso porque essa metatag não é exclusiva do Bootstrap, qualquer site deve utilizá-la, não importa o framework.

O quê? Você já tinha esquecido que estávamos estudando um framework? Cara, você precisa de um pouco de ação. Então saia, vá dar um “rolê” e volte daqui a pouco.

Pronto, lembrou? Agora, sim, vamos lá, meu amigo.

Esse atributo, o viewport, é simplesmente a chave para a responsividade das páginas entre os mais diversos dispositivos, e quando digo diversos, me refiro a todos que possuam um navegador de internet capaz de traduzir esses novos elementos HTML e realizar a mágica. Então “PELO AMOR DE DEUS”, não deixe de colocar esse sujeito no momento em que formos para o nosso HANDS-ON. Gostou? Aprendi hoje essa palavra e vou usá-la sempre agora 😊.

A viewport vai fazer com que sua página ocupe bonitinho o espaço da tela dos mais diversos tipos de dispositivo, e hoje, qualquer coisa acessa a internet, então nem preciso dizer que isso é imprescindível. Tá, mas e o valor que o acompanha, o que é aquilo?

Quando determinamos `<content="width=device-width, initial-scale=1.0">`, especificamos que:

- `width=device-width` – assumir o tamanho da tela do dispositivo, ou seja, o viewport vai tentar se encaixar na tela do dispositivo em que sua página está sendo apresentada.
- `initial-scale=1.0` – carregar nível de zoom inicial. Aqui, você determina que, assim que a página carregar, por exemplo, em um dispositivo, ela vai ter esse nível de zoom.
- `maximum-scale=1.0` – proibir qualquer nível de zoom, aqui você determina que o usuário não vai poder alterar o nível de zoom.

Proibir ZOOM – isso pode ser um problema, levando-se em consideração as pessoas que possuem dificuldade de leitura. Já imaginou o ódio que ela vai sentir quando tentar dar um zoom na tela do celular e não conseguir?

Além da viewport, existe mais uma metatag que deve ser colocada no cabeçalho de sua página. É a `http-equiv`, que vai forçar o Internet Explorer a utilizar as definições de css mais recentes. Caso não a utilize, sua página vai aparecer sem nenhum css.

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE-edge">
```

```
<meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
</head>
```

Código-fonte 4 – Exemplo da metatag obrigatória para IE
Fonte: Elaborado pelo autor (2017)

1.4.5 Referência de arquivos

Gente, estamos trabalhando com HTML, lembra? Aqui vale a mesma regra que utilizamos nos projetos normais.

Caso tenha escolhido a opção de baixar os arquivos do Bootstrap, faça a referência aos arquivos.

```
1 - <!Doctype html>  
2 -   <html>  
3 -     <head>  
4 -       <meta name="description" content="BOOTSTRAP">  
5 -       <link rel="stylesheet" href="css/bootstrap.min.css">  
6 -     </head>  
7 -     <body>  
8 -       <p>Hello Bootstrap!</p>  
9 -     </body>  
10 - </html>
```

Código-fonte 5 – Exemplo de referência para arquivo do Bootstrap
Fonte: Elaborado pelo autor (2017)

Não se esqueça de fazer as referências para os arquivos JS também, lembre-se de que sem eles, nem todos os componentes irão funcionar.

```
1 - <!Doctype html>  
2 -   <html>  
3 -     <head>  
4 -       <meta name="description" content="BOOTSTRAP">  
5 -       <link rel="stylesheet" href="css/bootstrap.min.css">  
6 -     </head>  
7 -     <body>  
8 -       <p>Hello Bootstrap!</p>  
9 -       <script src="js/bootstrap.bundle.min.js"></script>  
10 -    </body>  
11 - </html>
```

Código-fonte 6 – Exemplo de referência para arquivos JS do Bootstrap
Fonte: Elaborado pelo autor (2017)

1.4.6 Referência de arquivos remotos – CDN

Acima, realizamos a referência a arquivos locais que poderiam estar em nossas máquinas, mas como vimos anteriormente, podemos utilizar servidores remotos que hospedam arquivos, popularmente chamados de CDN (*Content Delivery Network*).

Há um pró e um contra em se utilizar os CDN.

O pró:

- Sempre Atualizado: se você optar por utilizar o CDN, isso lhe garante estar sempre com a última versão da biblioteca que você utiliza, pois está buscando na fonte.

O contra:

- Acesso Negado: imagine que pessoas de uma determinada empresa acessem seu site, que por sua vez se utiliza de CDN que está hospedado nos servidores externos. Até aí, normal, o problema começa quando a política de segurança da empresa proíbe o acesso a determinados domínios, e isso pode ocorrer com seu CDN. Aí, meu amigo, seu site vai sair disto:

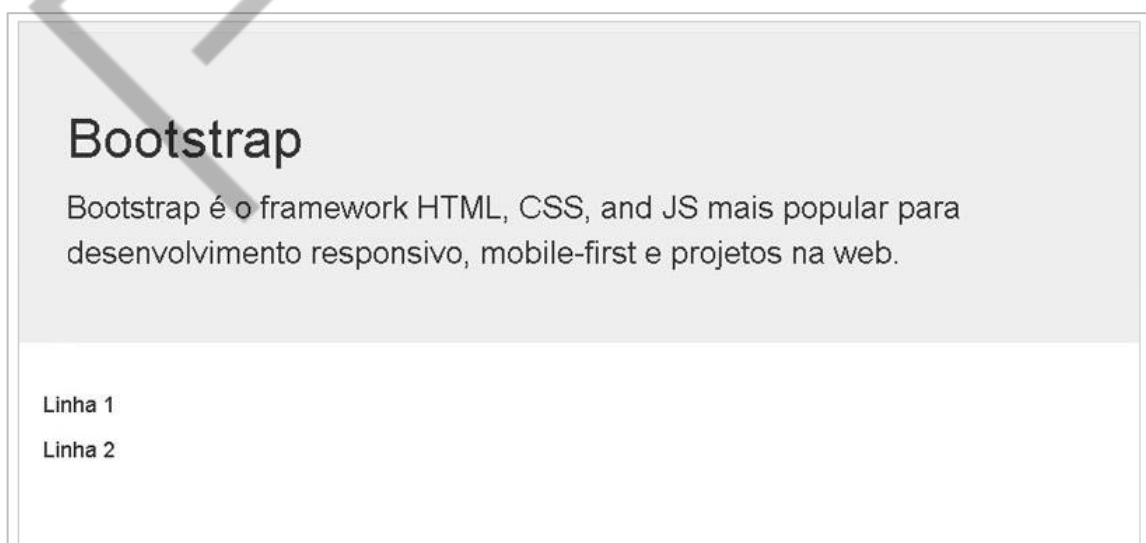


Figura 9 – Exemplo 1 de página com Bootstrap utilizando referência por CDN
Fonte: Elaborado pelo autor (2017)

Para isto:

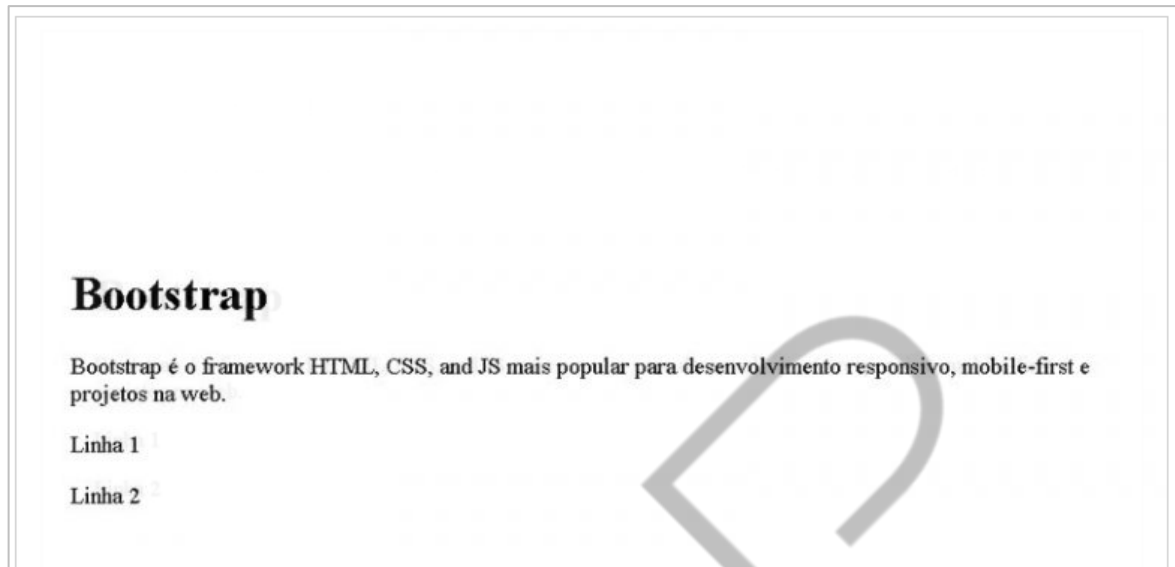


Figura 10 – Exemplo 2 de página com Bootstrap utilizando referência por CDN
Fonte: Elaborado pelo autor (2017)

Então, o segredo é pensar antes da utilização para que não haja problemas.

- Lentidão: além de acessar o seu site, o usuário vai ter que acessar o site do CDN também, para puxar as versões dos arquivos do framework e isso é mais demorado do que acessar os arquivos localmente em sua página.

Veja como você pode fazer para utilizar a referência de um CDN.

```
<head>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgw
ra6" crossorigin="anonymous">
</head>
```

Código-fonte 7 – Exemplo de referência para um servidor CDN do Bootstrap
Fonte: Elaborado pelo autor (2021)

No próprio site do Bootstrap, existe uma lista com os links para o CDN do framework.

Acesse <<https://getbootstrap.com/docs/5.0/getting-started/download/#cdn-via-jsdelivr>>.

CDN via jsDelivr

Skip the download with [jsDelivr](#) to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-..."></script>
```

If you're using our compiled JavaScript and prefer to include Popper separately, add Popper before our JS, via a CDN preferably.

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js" integrity="sha384-..."></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.min.js" integrity="sha384-..."></script>
```

Figura 11 – Tela dos servidores CDN do Bootstrap
Fonte: Elaborado pelo autor (2021)

1.4.7 HTML5

Pessoal, o framework só vai funcionar corretamente se o seu arquivo HTML tiver sido definido como HTML5.

Se não sabe como, dê uma olhada no Código-fonte “Exemplo de arquivo HTML5” abaixo.

```
1 - <!doctype html>
2 - <html>
3 -   <head>
4 -     <meta name="description" content="BOOTSTRAP">
5 -     <link rel="stylesheet" href="css/bootstrap.min.css">
6 -   </head>
7 -   <body>
8 -     <p>Hello Bootstrap!</p>
9 -     <script src="js/bootstrap.bundle.min.js"></script>
10 -   </body>
11 - </html>
```

Código-fonte 8 – Exemplo de arquivo HTML5
Fonte: Elaborado pelo autor (2017)

No código-fonte acima, temos uma página normal que, aos olhos dos leigos, passaria despercebida. Mas eu sempre digo que tem esse “mas”.

Na linha 1, temos o que chamamos de declaração `<!doctype html>`. Essa declaração vai dizer aos navegadores que esse arquivo é do tipo HTML5. Caso essa

declaração seja esquecida ou escrita de forma incorreta, fique tranquilo, a página vai carregar. Ufa!

Porém, tudo que é baseado no HTML5 será traduzido em HTML 4.1. Daí, meu camaradinho, você vai ter alguns usuários furiosos com você pelo simples fato de o seu site ficar todo torto ou os elementos simplesmente não funcionarem. Isso porque existem elementos HTML que só funcionam dentro de um arquivo HTML5. Então, fique atento.

EMENDAS

2 CRIANDO TEMPLATES

Bem, até agora, nosso trabalho se resumiu a: obter uma boa IDE para o desenvolvimento, conhecer algumas particularidades do framework, descobrir novos elementos HTML que são obrigatórios e baixar os arquivos necessários.

Agora, nós vamos aprender como juntar tudo isso em uma página HTML5 e aplicar os conceitos do framework, criar o que chamamos de *template* básico ou *boilerplate*. Seria como um arquivo modelo do framework para poupar trabalho.

2.1 Boilerplate

Chegou o momento do nosso HANDS-ON. Vamos iniciar pela criação do arquivo HTML chamado index.html e colocá-lo dentro da pasta que criamos.

Abaixo, o código utilizado neste primeiro exemplo, explicando linha a linha em comentários no próprio código.

```
<!doctype html>
<html lang="pt-br">
  <head>
    <!-- A codificação de caracteres que será utilizada: -->
    <meta charset="utf-8">
    <!-- Faz com que a página ocupe todo o espaço da tela do
dispositivo que é apresentada: -->
    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <!-- Referencia do arquivo Bootstrap CSS do toolkit: -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgw
ra6" crossorigin="anonymous">

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Referencia do arquivo Bootstrap Bundle JS (com o
Popper) do toolkit: -->
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
```

```
beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG8R+phH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLts3qm9
Ekf" crossorigin="anonymous"></script>
</body>
</html>
```

Código-fonte 9 – Página HTML5 com Bootstrap comentada
Fonte: Elaborado pelo autor (2021)

Vamos ver como ficou o resultado desse exemplo em dois dispositivos diferentes: no PC e em uma tela de celular.

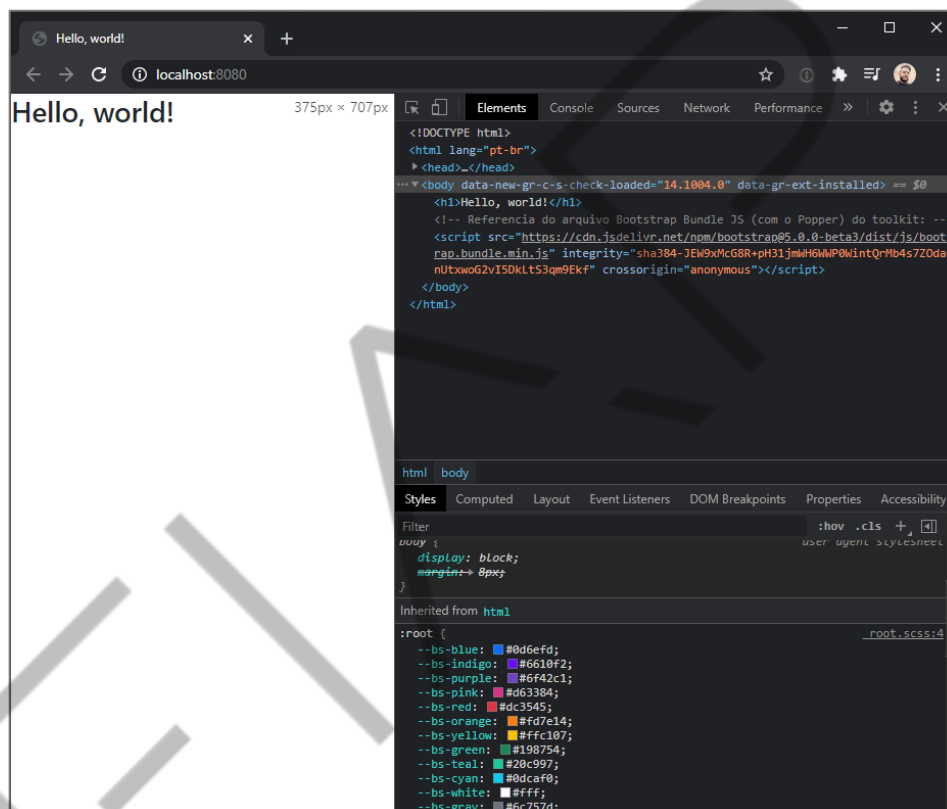


Figura 12 – Resultado do exemplo no PC com largura de 375 x 707
Fonte: Elaborado pelo autor (2021)

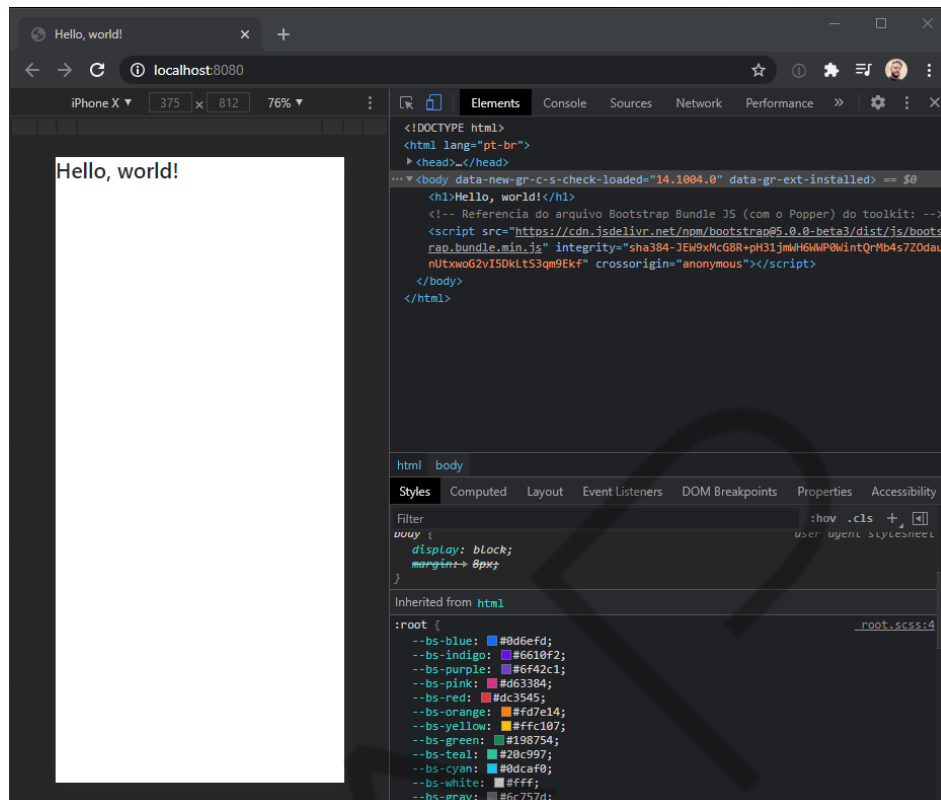


Figura 13 – Resultado do exemplo no celular com largura de 375 x 812
Fonte: Elaborado pelo autor (2021)

Podemos ver que a exibição entre os dois dispositivos é diferente, isso ocorre porque estamos utilizando a tag <meta> viewport.

A *viewport* é toda a área em que será apresentado o conteúdo.

Esse exemplo é o que chamamos de *boilerplate*, ele vai servir de base para nossos outros exemplos.

2.2 Estados

O Bootstrap divide suas ações em 6 tipos de estados diferentes, os quais são identificados por cores.

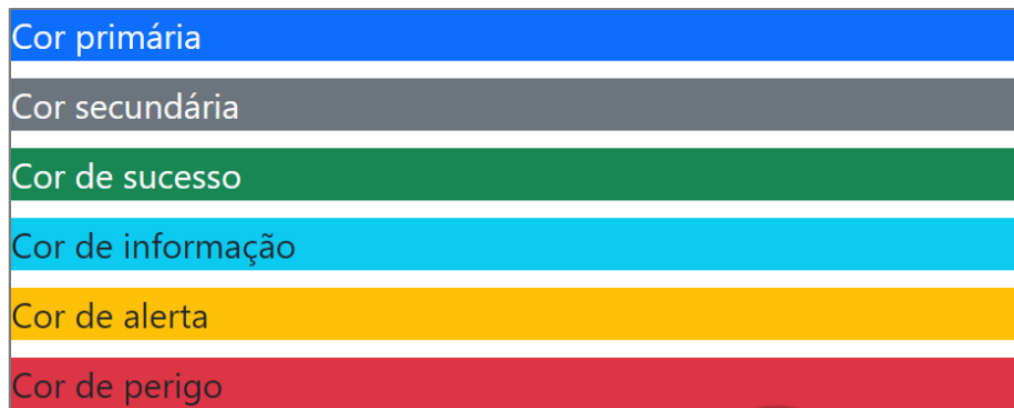


Figura 14 – Exemplo das cores de estados do Bootstrap
Fonte: Elaborado pelo autor (2021)

Então, quando estiver criando os componentes, fique atento à contextualização para que não haja enganos na aplicação dos estados.

2.3 Sistema De Grid

O sistema de grid do Bootstrap define colunas e linhas, alinhando na horizontal e na vertical todo o conteúdo na *viewport*. O sistema de grid deixa o layout com a aparência de uma tabela.

Esses blocos e linhas devem ficar dentro de um container.



Figura 15 – Exemplo do sistema de grid do Bootstrap
Fonte: Elaborado pelo autor (2017)

Como podemos ver na figura acima, o grid parece mesmo uma tabela, em que cada uma das colunas pode acomodar algum tipo de conteúdo nos espaços que foram predeterminados.

Observe que temos números dentro de cada coluna. Isso indica que cada linha tem várias ou uma coluna cuja soma deve sempre ser 12 ou menor que 12, isso ocorre porque o sistema de grid do Bootstrap é fixado em até 12 colunas. Então, daqui por diante, sempre que montarmos nossos layouts a soma da linha nunca deverá ultrapassar 12.

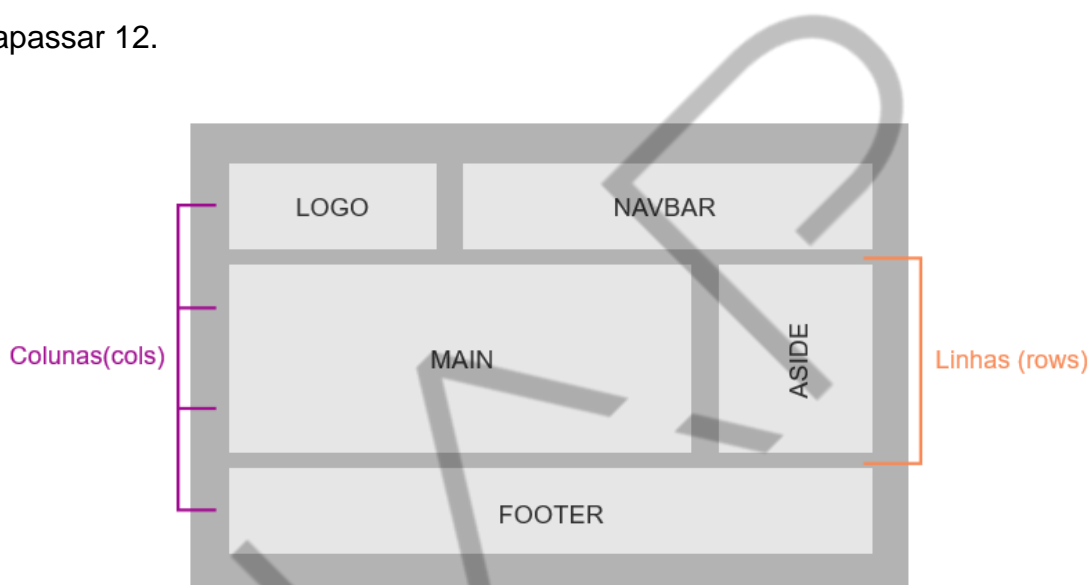


Figura 16 – Exemplo 2 do sistema de grid do Bootstrap
Fonte: Elaborado pelo autor (2017)

E, como já foi dito anteriormente, podemos colocar qualquer conteúdo dentro dessas unidades ou colunas.

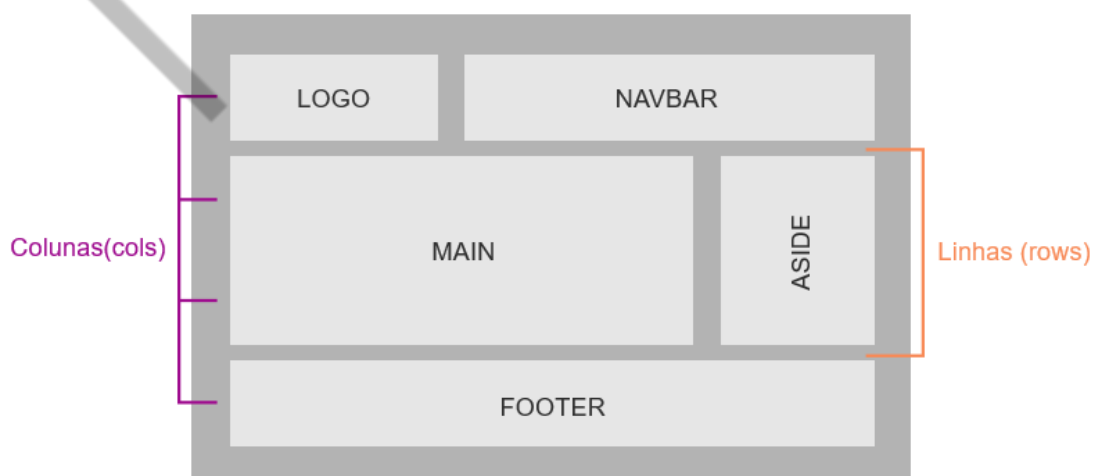


Figura 17 – Exemplo 3 do sistema de grid do Bootstrap com um layout
Fonte: Elaborado pelo autor (2017)

Então aprendemos aqui que o sistema de grid do Bootstrap é formado por linhas e colunas que devem estar dentro de um container.

A soma da coluna por linha nunca pode ultrapassar 12.

As colunas comportam os mais diversos conteúdos, vídeos, textos e imagens.

2.3.1 Container

Agora que já conhecemos o sistema de grid, vamos partir para o HANDS-ON a respeito do container.

O container é o box base para a montagem do layout com o Bootstrap. Todos os outros elementos devem estar dentro do container.

Mas o que é um container?

Um container é uma div que foi declarada com a classe específica de .container do Bootstrap e é assim que vamos trabalhar, utilizando suas classes já predefinidas, realizando alguns ajustes quando necessário.

Classes/componentes – você pode conhecer todos os componentes do Bootstrap e dicas de utilização de cada um deles neste endereço:
<<http://getbootstrap.com/components/>>.

Voltando ao container, existem duas classes de containers: o tipo fixed e fluid.

O container fixed mantém o tamanho em caso de redimensionamento da tela e fica posicionado no centro da viewport.

Vamos criar um exemplo utilizando nosso boilerplate existente e adicionando o container a ele.

```
<!doctype html>
<html lang="pt-br">
  <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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
```

```
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgw
ra6" crossorigin="anonymous">

<title>Bootstrap - Hands-On</title>

<style type="text/css">
  .container {
    background-color: var(--bs-secondary);
  }
</style>
</head>
<body>
  <div class="container text-light">

    Container fixo.

  </div>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9
Ekf" crossorigin="anonymous"></script>
  </body>
</html>
```

Código-fonte 10 – Página HTML5 com um container fixed
Fonte: Elaborado pelo autor (2017)

Veja que, além da div e da declaração do container, foi criado um bloco style no head para realizar a alteração de cor de fundo do container de modo que possamos enxergar o exemplo.

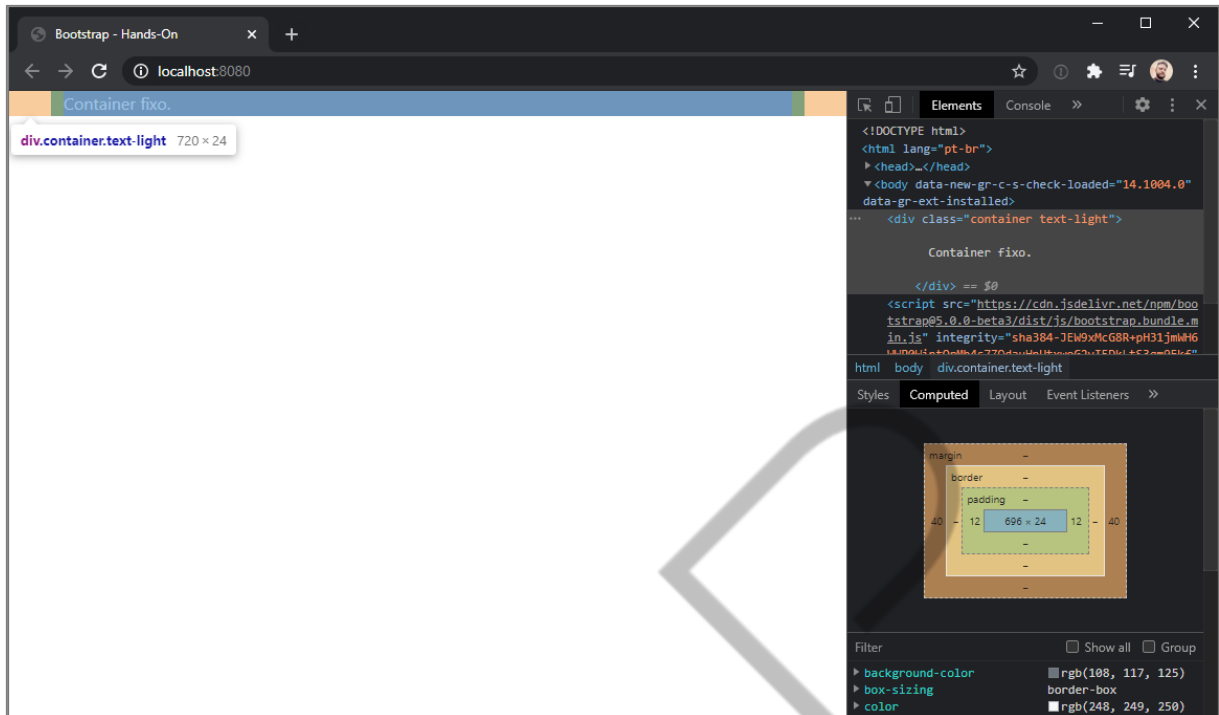


Figura 18 – Exemplo de container fixed
Fonte: Elaborado pelo autor (2021)

Podemos ver que o container fixed se posiciona ao centro, deixando espaços laterais na viewport.

Agora, vamos verificar no container-fluid. Esse container tem a característica inversa do fixed, porque ele se expande e toma todo o espaço da viewport.

Vamos ver um exemplo desse tipo de container.

```

<!doctype html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgw
ra6" crossorigin="anonymous">

    <title>Bootstrap - Hands-On</title>

    <style type="text/css">
      .container-fluid {
        background-color: var(--bs-secondary);

```

```
    }  
  </style>  
</head>  
<body>  
  <div class="container-fluid text-light">  
  
    Container fluido.  
  
  </div>  
  
  <script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-  
beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-  
JEW9xMcG8R+phH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLTS3qm9  
Ekf" crossorigin="anonymous"></script>  
  </body>  
</html>
```

Código-fonte 11 – Página HTML5 com um container fluid
Fonte: Elaborado pelo autor (2017)

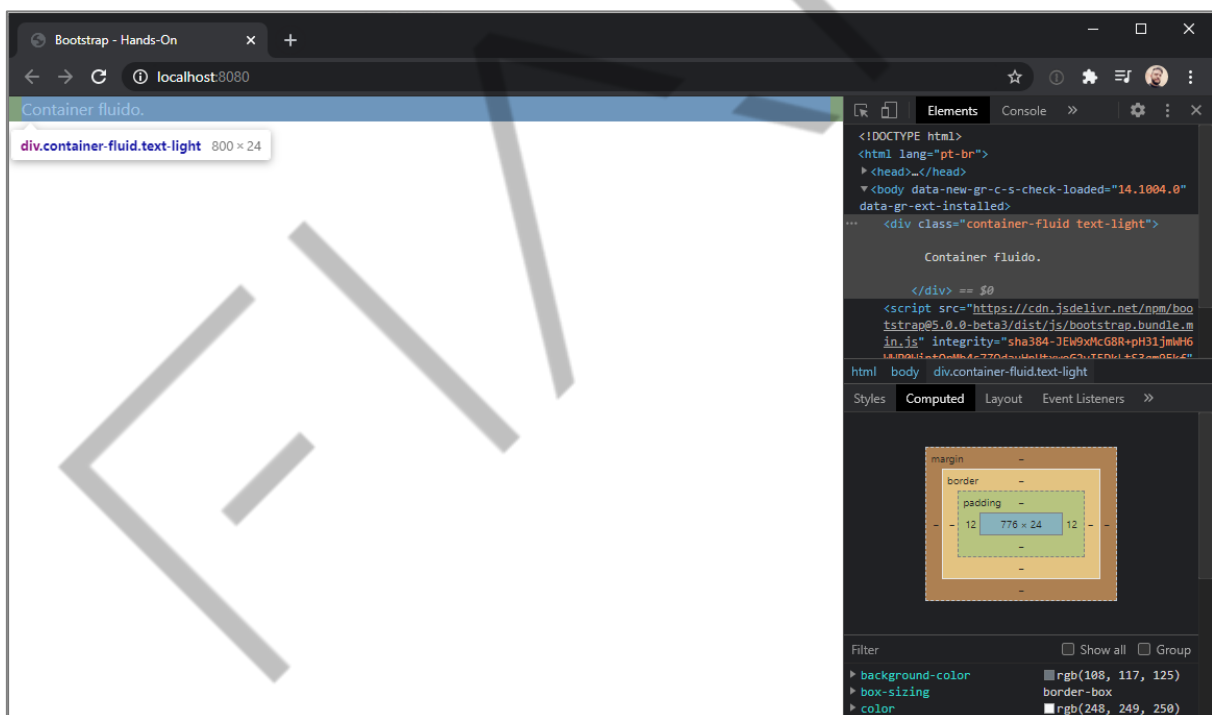


Figura 19 – Exemplo de container fluid
Fonte: Elaborado pelo autor (2021)

O container fluid ocupa todo o espaço da viewport e é o mais indicado para utilização geral.

Veja que os containers devem ser utilizados porque, com eles, a visualização da página fica bem melhor.

Alteração de componentes – gente, não é porque estamos utilizando o Bootstrap que não podemos alterar um ou outro componente. Vocês viram que alterei a background-color do container por meio de um bloco style que não é recomendado. Caso haja a necessidade de alteração, faça-a por meio de arquivos css.

Atenção – sua página deve ter pelo menos um container geral que englobe tudo. Chamamos esse container de Wrapper.

2.3.2 Linhas (rows) e colunas (columns)

Para completar o layout, necessitamos das linhas (rows) e das colunas (columns).

Você cria as linhas dentro dos containers e qualquer elemento HTML de bloco pode ser declarado com essa classe do Bootstrap.

O prefixo utilizado é row, então basta criar o elemento HTML e declarar a classe dessa forma `<div class="row">...</div>`. Pronto, temos uma linha criada.

As linhas são responsáveis pelo alinhamento horizontal do conteúdo na viewport. Elas podem ser criadas dentro de outras linhas e não existe um limite para a quantidade desse componente.

```
<!doctype html>
<html lang="pt-br">
  <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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbz
Kgwa6" crossorigin="anonymous">

    <title>Bootstrap - Hands-On</title>

    <style type="text/css">
      .container-fluid {
        background-color: var(--bs-secondary);
      }
  </head>
```

```
.row {
  background-color: var(--bs-primary);
}
</style>
</head>
<body>
  <div class="container-fluid text-light">

    <h1>Container fluido.</h1>

    <div class="row">
      <h2>Essa é uma linha...</h2>
    </div>

  </div>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG8R+phH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3
qm9Ekf" crossorigin="anonymous"></script>
  </body>
</html>
```

Código-fonte 12 – Página HTML5 com um container fluid
Fonte: Elaborado pelo autor (2021)

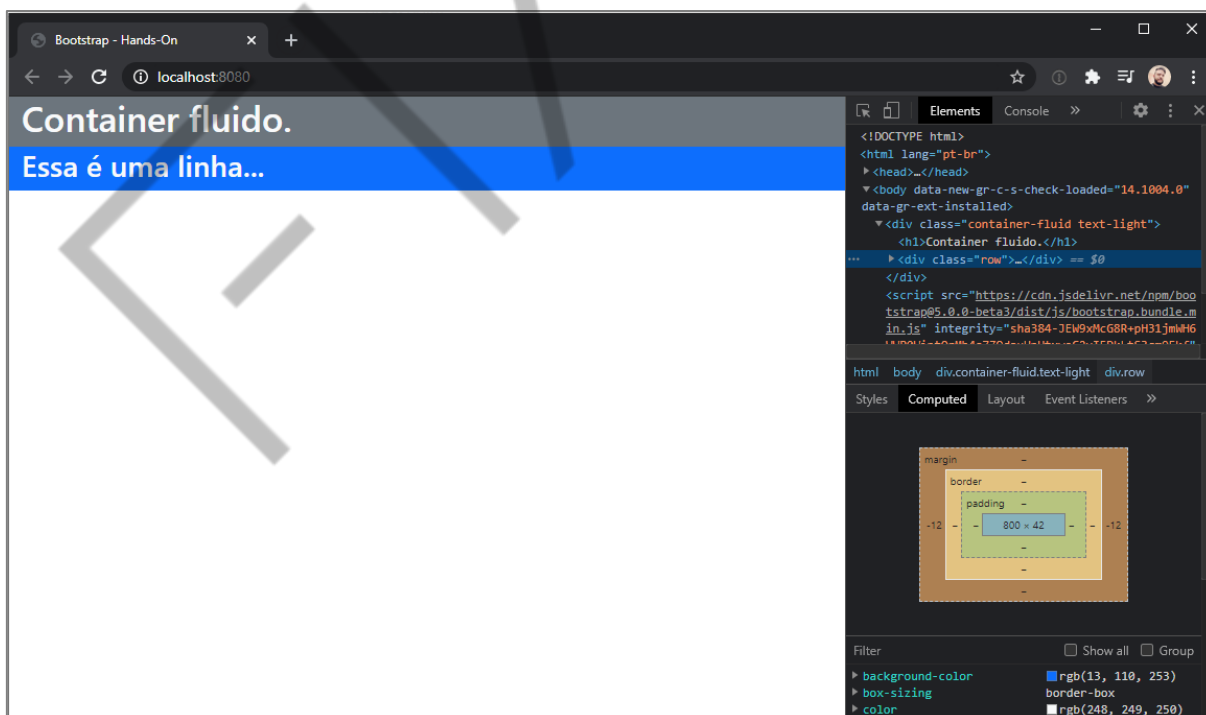


Figura 20 – Exemplo de linha (row)
Fonte: Elaborado pelo autor (2021)

Completando o grid, temos as colunas (*columns*) que são responsáveis pelo alinhamento vertical do conteúdo na viewport.

As colunas devem estar dentro das linhas (*rows*) e, assim como as linhas, podem ser declaradas com a classe específica para os mais diversos elementos de bloco HTML.

Para a declaração das colunas, o Bootstrap utiliza quatro medidas diferentes e determina uma classe para cada uma delas, levando em consideração o dispositivo que está carregando o site.

- col-xs – para telefones celulares.
- col-sm – para tablets.
- col-md – para desktops.
- col-lg – para monitores grandes, TVs.

Então, se quiséssemos declarar uma coluna dentro de uma linha, bastaria fazer isto: `<div class="col-md-4">...</div>` , `<div class="col-md-8">...</div>`. Podemos ver que a soma dos valores é igual a 12. Poderia ser menor que 12?

Sim, poderia: `<div class="col-md-2">...</div>` , `<div class="col-md-5">...</div>`

```
<!doctype html>
<html lang="pt-br">
  <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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rpP48ckxlpbz
Kgwra6" crossorigin="anonymous">

    <title>Bootstrap - Hands-On</title>

    <style type="text/css">
      .container-fluid {
        background-color: var(--bs-secondary);
      }

      .row {
```



```
        background-color: var(--bs-primary);
    }

    .col-md-4 {
        background-color: var(--bs-info);
    }

    .col-md-8 {
        background-color: var(--bs-warning);
    }
</style>
</head>
<body>
    <div class="container-fluid text-light">

        <h1>Container fluido.</h1>

        <div class="row">
            <h2>Essa é uma linha...</h2>

            <aside class="col-md-4">
                <h3>ASIDE</h3>
            </aside>

            <section class="col-md-8">
                <h3>SECTION</h3>
            </section>
        </div>

    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3
qm9Ekf" crossorigin="anonymous"></script>
    </body>
</html>
```

Código-fonte 13 – Página HTML5 com exemplo de coluna
Fonte: Elaborado pelo autor (2021)

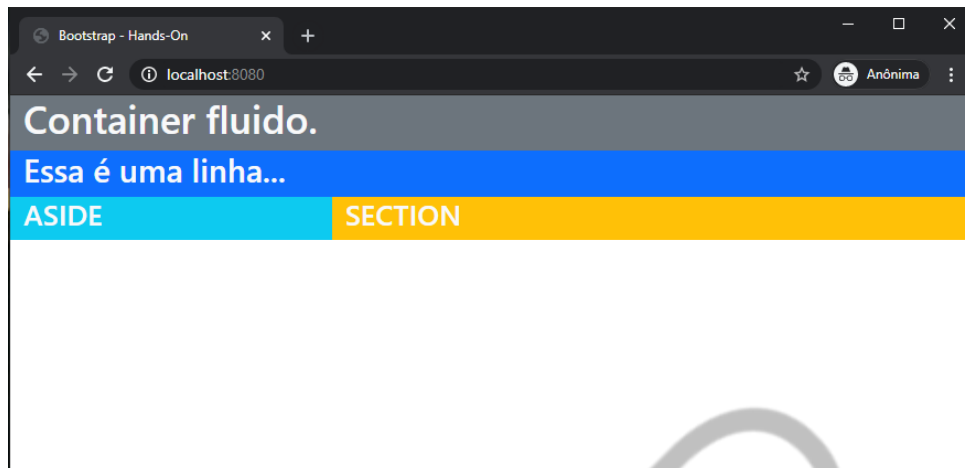


Figura 21 – Exemplo de coluna (*columns*)
Fonte: Elaborado pelo autor (2021)

Para. Para tudo!

O Bootstrap não é MobileFirst? Ele não entende sempre que a tela é de um dispositivo pequeno, e só depois vai se adaptando ao tamanho real da tela?

Por que existem essas 4 medidas ou classes aí? Hein?

Elas existem exatamente por este motivo: para que o Bootstrap possa se adaptar e utilizá-las de modo que o viewport seja apresentado da melhor maneira.

Então eu precisarei predeterminar para que tipo de dispositivo vou desenvolver?

A resposta para a pergunta acima é: depende.

Amigos, planejamento é tudo. Seria ótimo se você já predeterminasse para qual dispositivo seu site vai ser desenvolvido. Porém, hoje vivemos em tempos de responsividade. Então um site que carrega no celular precisa carregar na TV de tela grande.

Mas como faço se eu determinar a largura da coluna para um desktop e o cidadão abrir no celular?

As boas práticas pedem que se utilize a classe `col-md`, que de certa forma é a padrão e é a que melhor se adapta a outros dispositivos. E aí você vai se valer do conceito MobileFirst, lembra? O framework vai adaptar seu site inicialmente para os dispositivos mobile e vai subindo, até chegar ao tamanho que está sendo apresentado.

Vamos ver agora um exemplo de um layout padrão muito utilizado hoje em dia, é o layout-fixed. Vou mostrar como montar esse layout e aplicar algum css nele também.

```
<!doctype html>
<html lang="pt-br">
  <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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rpP48ckxlpbzKgw
ra6" crossorigin="anonymous">

    <link href="styles.css" rel="stylesheet">

    <title>Bootstrap - Hands-On</title>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <header>Cabeçalho</header>
      </div>

      <div class="row">
        <main>Principal</main>
      </div>

      <div class="row">
        <footer>Rodapé</footer>
      </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9
Ekf" crossorigin="anonymous"></script>
  </body>
</html>
```

Código-fonte 14 – Página HTML5 de exemplo completo com o grid e layout fixed

Fonte: Elaborado pelo autor (2017)

```
.container {
  padding-left: 10px;
  padding-right: 10px;
  background-color: gray;
}

.row {
  margin-bottom: 20px;
}

header, main, footer {
  background-color: aqua;
  border: 1px solid #c0c0c0;
  font-size: 50px;
  padding: 10px 0;
  text-align: center;
}

main {
  background-color: white;
}

@media screen and (min-width: 1920px) {
  main {
    height: 80vh;
  }
}

@media screen and (min-width: 980px) {
  main {
    height: 60vh;
  }
}
```

Código-fonte 15 – Arquivo styleless.css
Fonte: Elaborado pelo autor (2021)

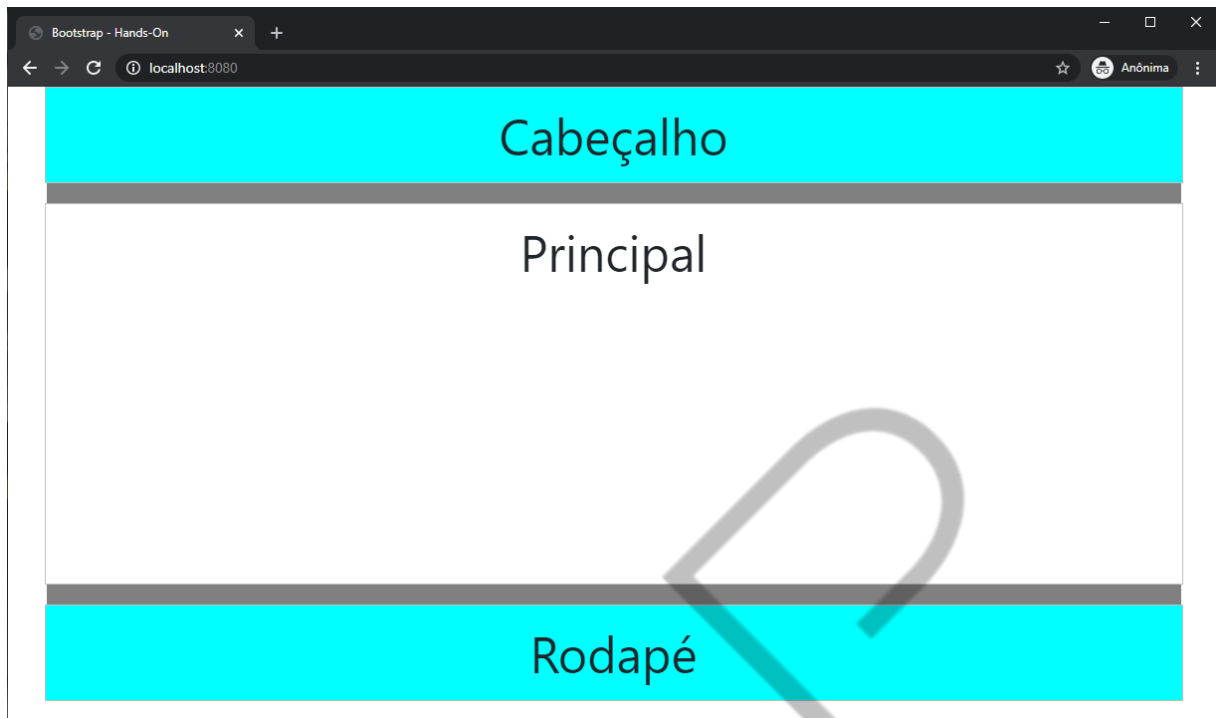


Figura 22 – Exemplo de página com o grid completo utilizando fixed layout
Fonte: Elaborado pelo autor (2021)

Podemos identificar o fixed layout pelo fato de o conteúdo permanecer no centro da viewport.

```
<!doctype html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgw
ra6" crossorigin="anonymous">

    <link href="styles.css" rel="stylesheet">

    <title>Bootstrap - Hands-On</title>
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <header>Cabeçalho</header>
      </div>
```

```
<div class="row">
  <main>Principal</main>
</div>

<div class="row">
  <footer>Rodapé</footer>
</div>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9
Ekf" crossorigin="anonymous"></script>
</body>
</html>
```

Código-fonte 16 – Página HTML5 de exemplo completo com o grid e layout fluid
Fonte: Elaborado pelo autor (2021)

```
.container, .container-fluid {
  background-color: gray;
}

.row {
  margin-bottom: 20px;
  box-sizing: border-box;
}

header, main, footer {
  background-color: aqua;
  border: 1px solid #c0c0c0;
  font-size: 50px;
  padding: 10px 0;
  text-align: center;
}

main {
  background-color: white;
}

@media screen and (max-width: 1920px) {
  main {
    height: 70vh;
  }
}

@media screen and (max-width: 980px) {
  main {
    height: 60vh;
  }
}
```

```
}  
}  
  
@media screen and (max-width: 520px) {  
  main {  
    height: 50vh;  
  }  
}  
  
@media screen and (max-width: 360px) {  
  main {  
    background-color: var(--bs-warning);  
  }  
}
```

Código-fonte 17 – Arquivo styles.css
Fonte: Elaborado pelo autor (2021)

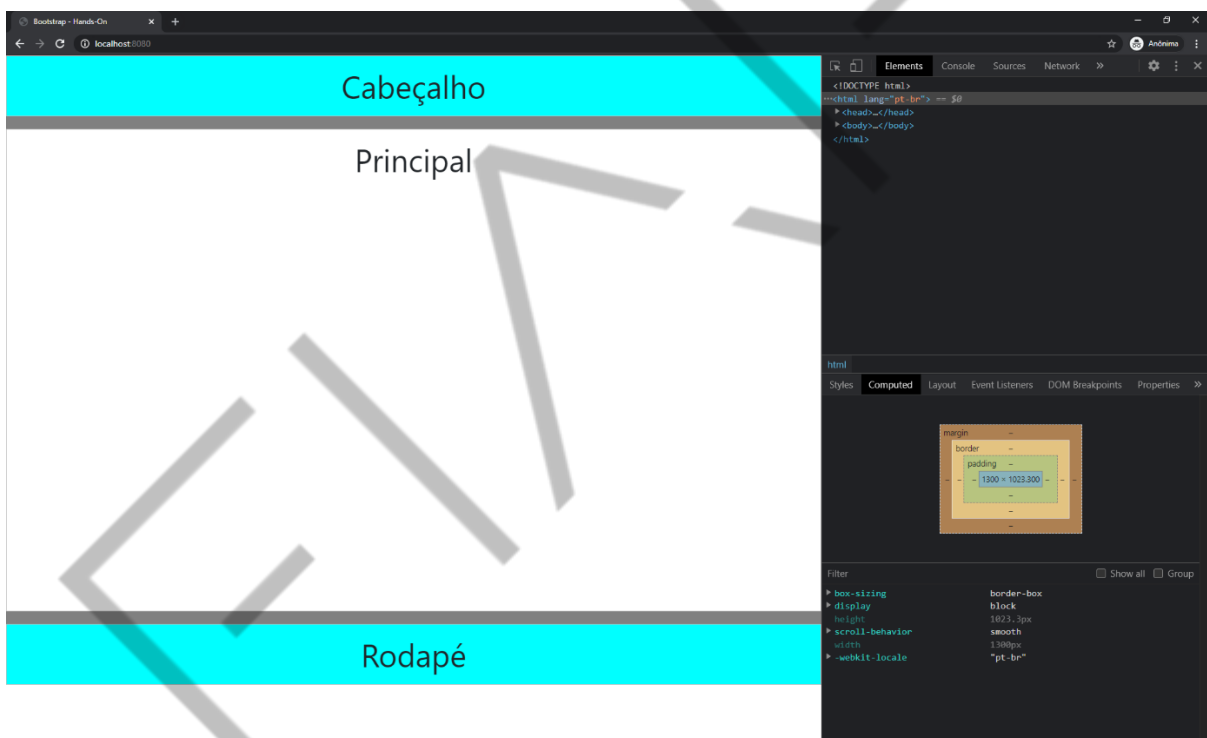


Figura 23 – Exemplo de página com o grid completo utilizando fluid layout
Fonte: Elaborado pelo autor (2021)

E aqui, utilizando o layout padrão fluid, que ocupa todo o espaço da viewport.

Os dois exemplos apresentados estão 100% responsivos. Se redimensionarmos a janela do site ou o abrirmos em um tablet ou celular, teremos o resultado a seguir.

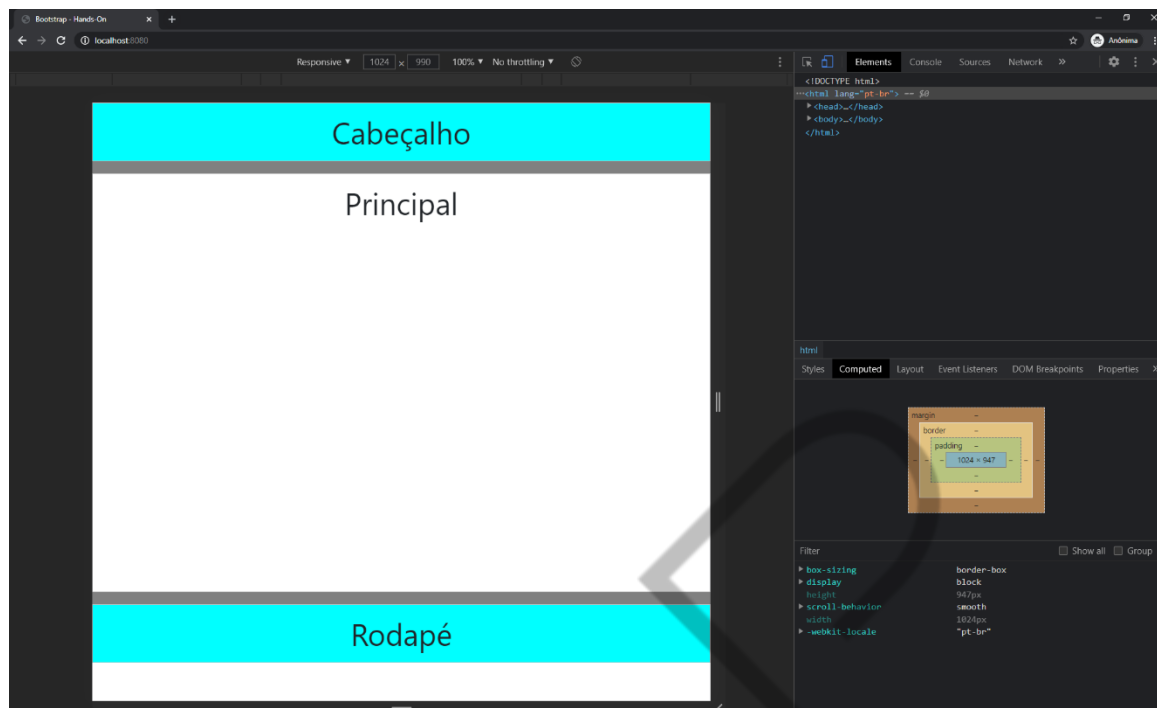


Figura 24 – Exemplo de página com o grid completo utilizando fluid layout e redimensionada
Fonte: Elaborado pelo autor (2021)

Vamos simular a visualização na tela de um celular. É possível realizar essa simulação nos principais navegadores, pressionando a Tecla F12 e clicando no botão de redimensionamento.

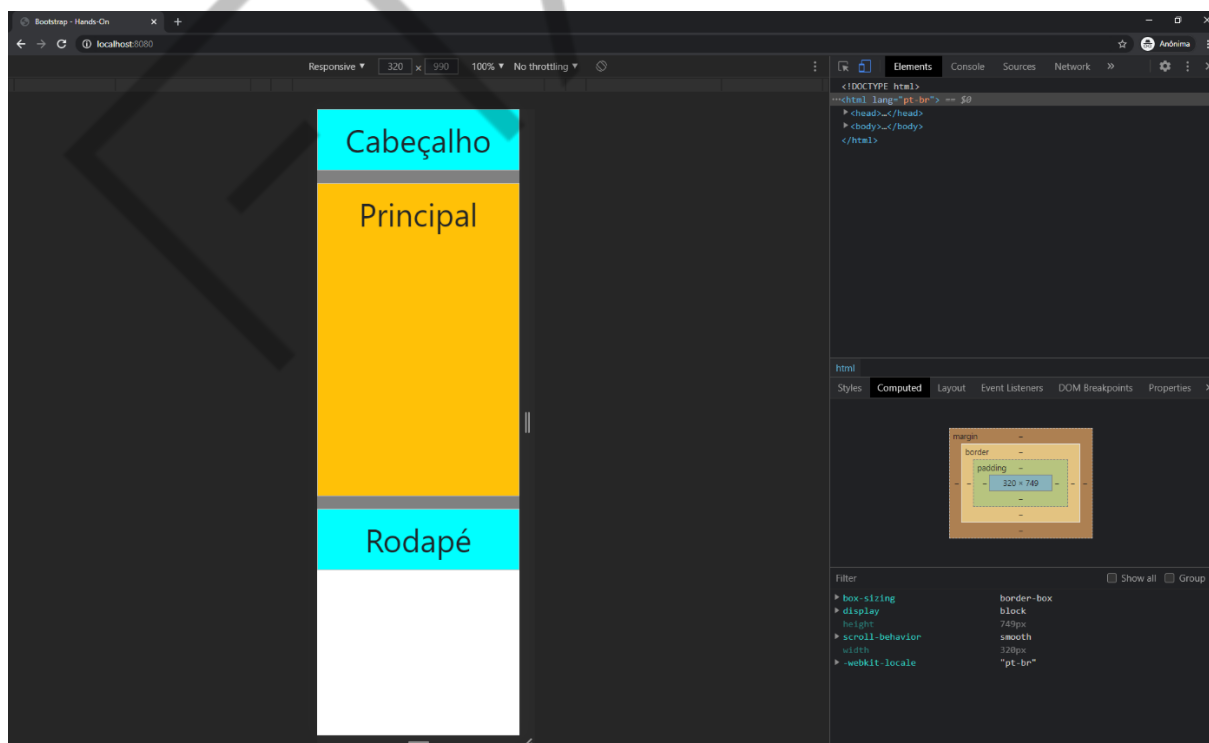


Figura 25 – Exemplo de página com o grid completo utilizando fluid layout sendo apresentada na tela de um celular
Fonte: Elaborado pelo autor (2021)

3 FORMULÁRIOS

Os formulários são elementos HTML que utilizam diversos tipos de componentes do Bootstrap e, de quebra, se aplica o grid layout nele.

Não se esqueça de sempre utilizar o container para envolver os elementos.

Vamos criar um formulário de login básico para demonstrar algumas classes.

Como já foi visto, podemos também posicionar na horizontal ou na vertical. Porém, vale ressaltar que ainda estamos falando de responsividade, então, caso a janela seja redimensionada ou o dispositivo seja diferente daquele para o qual a página foi desenvolvida, ela vai se adequar ao viewport apresentado.

3.1 Form Inline

```
<!doctype html>
<html lang="pt-br">

<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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
  crossorigin="anonymous" />

  <title>Bootstrap - Hands-On</title>
</head>

<body>
  <div class="container">
    <h2>Formulário Inline</h2>

    <form class="row row-cols-lg-auto">
      <div class="col-12">
        <label for="email">Email:</label>
        <input type="email" class="form-control" id="email"
placeholder="Digite o email" />
      </div>

      <div class="col-12">
```

```
<label for="pwd">Senha:</label>
<input type="password" class="form-control" id="pwd"
placeholder="Digite a senha" />
</div>

<button type="submit" class="btn btn-
primary">Enviar</button>
</form>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
JEW9xMcG8R+phH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
crossorigin="anonymous"></script>
</body>
</html>
```

Código-fonte 18 – Formulário inline
Fonte: Elaborado pelo autor (2021)

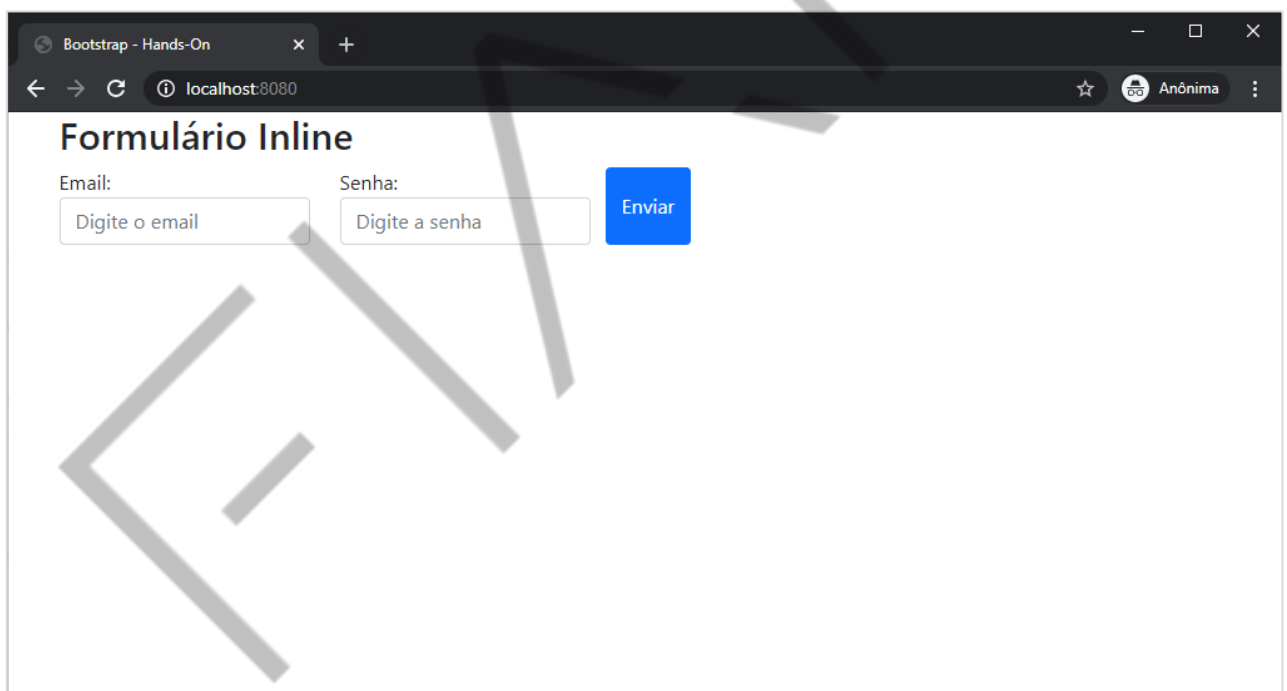


Figura 26 – Exemplo de página com um formulário inline
Fonte: Elaborado pelo autor (2021)

Aqui temos um formulário comum, em que foram aplicadas as classes do Bootstrap. A classe container já vimos em sistema de grids, agora temos outras que basicamente vão fazer o mesmo papel de controle dos elementos no grid.

A classe row-cols-lg-auto indica que o formulário em questão deve posicionar os seus elementos em linha, com o auxílio das classes col-*. Para os inputs,

determinamos a classe `form-control`, lembrando que o elemento com essa classe vai se submeter às regras do formulário.

3.2 Form horizontal

Fala a verdade: é bonito ver a responsividade trabalhando, né?

Agora, vamos ver um exemplo de form horizontal, aplicaremos algumas classes de controle nos labels e inputs.

```
<!doctype html>
<html lang="pt-br">

<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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
    crossorigin="anonymous" />

  <title>Bootstrap - Hands-On</title>
</head>

<body>
  <div class="container">
    <h2>Formulário Horizontal</h2>

    <form class="form-horizontal">
      <div class="row mb-3">
        <label class="control-label col-sm-2"
for="email">Email:</label>
        <div class="col-sm-10">
          <input type="email" class="form-control" id="email"
placeholder="Digite o email" />
        </div>
      </div>

      <div class="row mb-3">
        <label class="control-label col-sm-2"
for="pwd">Senha:</label>
        <div class="col-sm-10">
          <input type="password" class="form-control" id="pwd"
placeholder="Digite a senha" />
        </div>
      </div>
    </form>
  </div>
</body>
</html>
```

```
</div>

<div class="row mb-3">
  <div class="col-sm-10 offset-sm-2">
    <button type="submit" class="btn btn-primary">
      Enviar
    </button>
  </div>
</div>
</form>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
  crossorigin="anonymous"></script>
</body>
</html>
```

Código-fonte 19 – Formulário horizontal
Fonte: Elaborado pelo autor (2021)

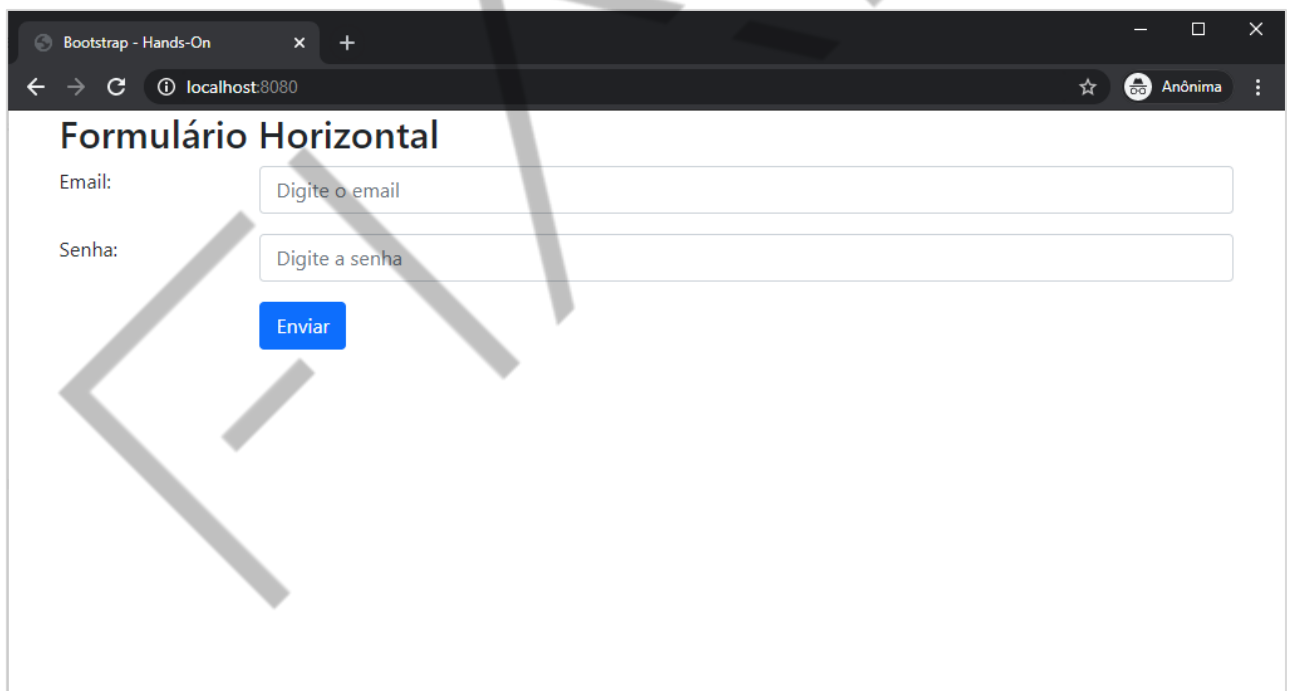


Figura 27 – Exemplo de página com um formulário horizontal
Fonte: Elaborado pelo autor (2021)

Temos nesse exemplo algumas classes de posicionamento importantes. A primeira classe declarada foi a form-horizontal, como no exemplo anterior, form-group e form-control também. Agora foram adicionadas as classes control-label, col-sm-

offset-x e a já conhecida col-sm-xx, em que xx é o tamanho da coluna que você quer dar para o elemento em questão.

A classe control-label é utilizada para aplicar efeitos do form-group ao label. Já em offset-sm-x, x é o número de colunas que você deseja deslocar a coluna para a direita.

3.3 Form vertical padrão

É engraçado falar isso, mas se você redimensionar uma página com um formulário inline ou horizontal, ele vai se tornar um formulário vertical. Vamos ver.

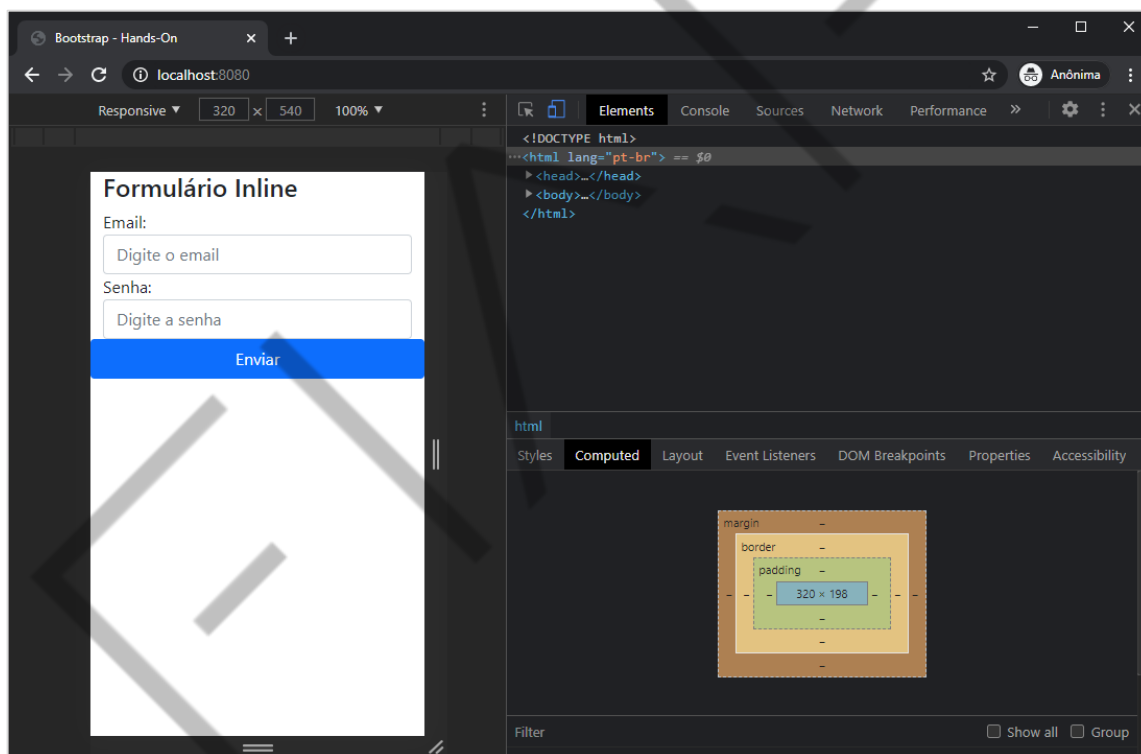


Figura 28 – Exemplo de página com um formulário inline redimensionado que se tornou um form vertical
Fonte: Elaborado pelo autor (2021)

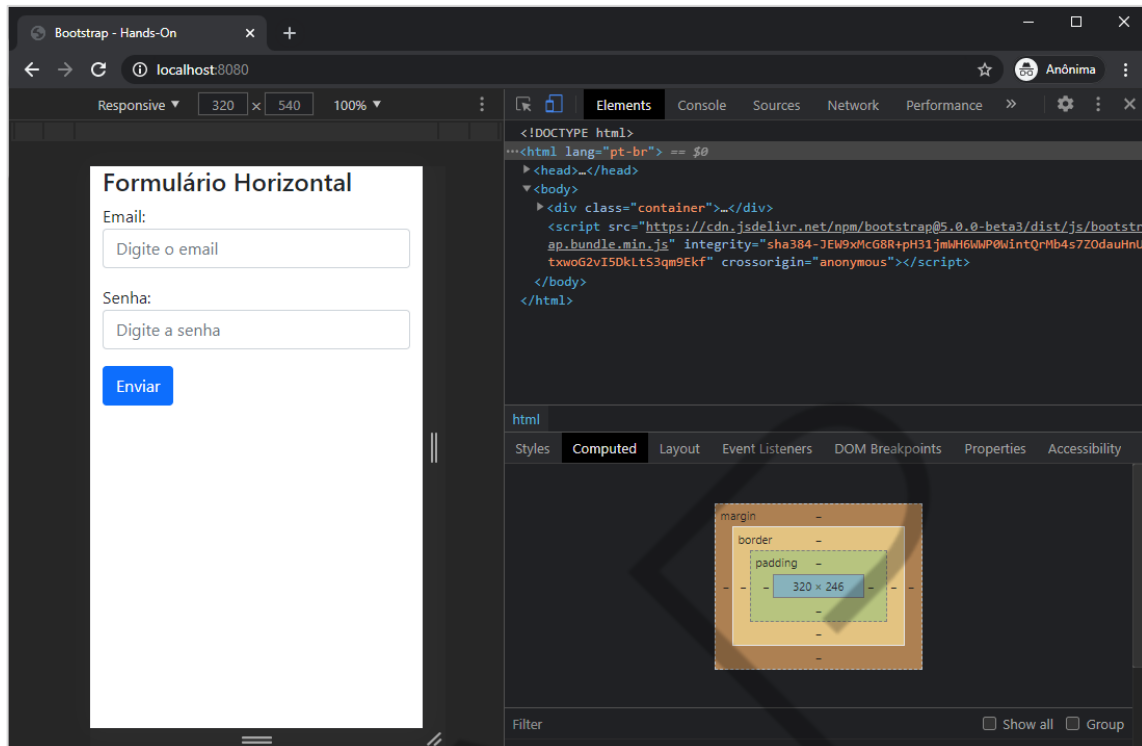


Figura 29 – Exemplo de página com um formulário horizontal redimensionado que se tornou um form vertical
Fonte: Elaborado pelo autor (2021)

Vamos ver no código de um form vertical.

```

<!doctype html>
<html lang="pt-br">

<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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
    crossorigin="anonymous" />

  <title>Bootstrap - Hands-On</title>
</head>

<body>
  <div class="container">
    <h2>Form Padrão ou Vertical</h2>

    <form>
      <div>

```

```
<label for="email">Email:</label>
<input type="email" class="form-control" id="email">
</div>
<div>
  <label for="pwd">Senha:</label>
  <input type="password" class="form-control" id="pwd">
</div>
<div>
  <label><input type="checkbox"> Lembre-me</label>
</div>
<button type="submit" class="btn btn-
primary">Enviar</button>
</form>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
  crossorigin="anonymous"></script>
</body>
</html>
```

Código-fonte 20 – Formulário padrão ou vertical
Fonte: Elaborado pelo autor (2021)

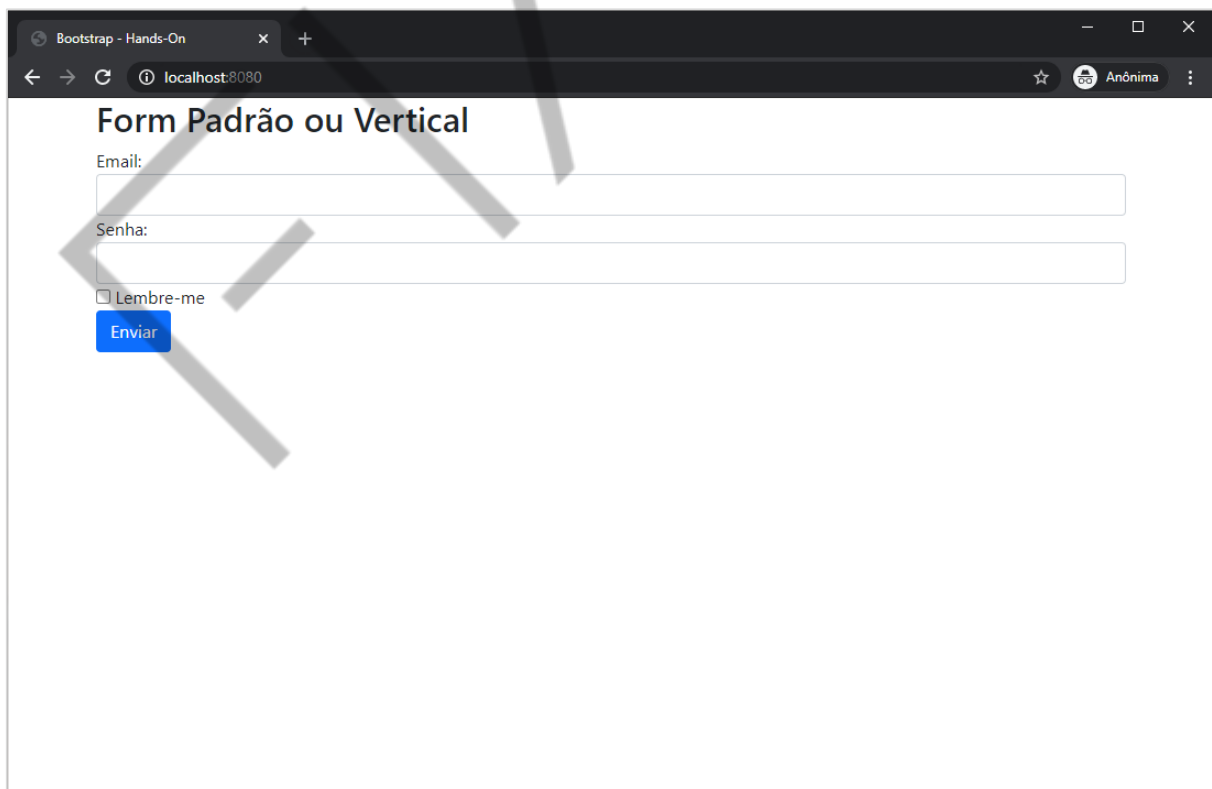


Figura 30 – Exemplo de página com um formulário padrão ou vertical
Fonte: Elaborado pelo autor (2021)

O controle dos formulários é bastante simples. Bastando, após a configuração do layout, uma boa dose de criatividade para poder aplicar o CSS.

3.4 MISC

Vou trazer para vocês agora uma das utilizações do Bootstrap que, na minha opinião, se destaca bastante. É a iconografia labels de avisos e botões.

Para demonstrar algumas dessas classes, vou utilizar nosso último exemplo dos forms.

3.4.1 Iconografia

Hoje, não existe comunicação em lugar nenhum sem ícones, seja no mundo real, seja no mundo virtual. O Bootstrap nos fornece uma lista de ícones próprios bem grandes.

A lista completa de ícones você pode conferir na página de componentes do Bootstrap, que pode ser acessada aqui: <<https://icons.getbootstrap.com>>.

Ícons

Start typing to filter...

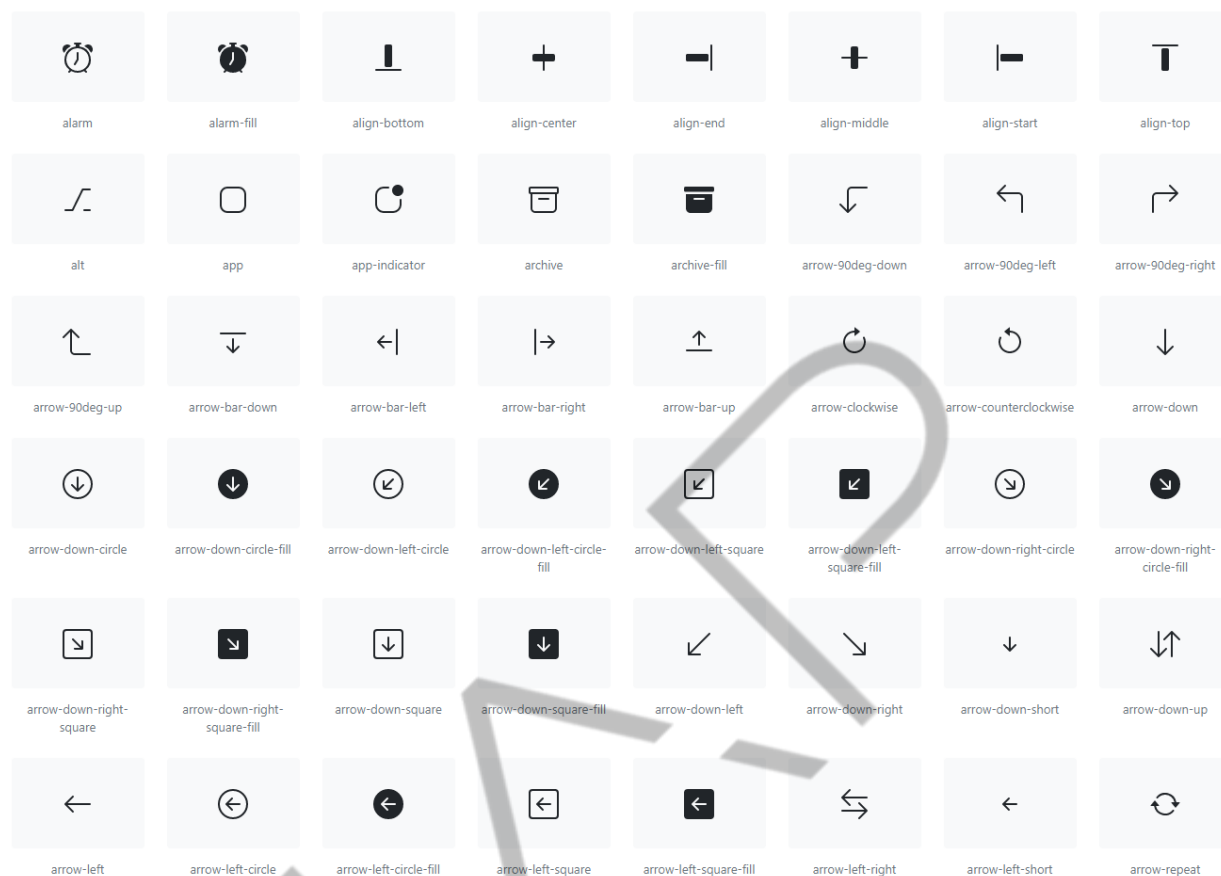


Figura 31 – Lista de ícones do Bootstrap

Fonte: Elaborado pelo autor (2021)

O ideal é que os ícones sejam apresentados com algum tipo de legenda de alerta para dar sentido ao que está sendo apresentado.

Aqui, apresentamos um ícone e um alert do tipo danger, para chamar a atenção do usuário.

```
<!doctype html>
<html lang="pt-br">

<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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
    crossorigin="anonymous" />

  <!-- Inclusão ao suporte de ícones do Bootstrap -->
```

```
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.4.1/font/bootstrap-icons.css">

<title>Bootstrap - Hands-On</title>
</head>

<body>
  <div class="container">
    <h2>Login</h2>

    <div class="alert alert-danger" role="alert">
      <i class="bi bi-exclamation-triangle"></i>
      Entre com um email válido.
    </div>

    <form>
      <div>
        <label for="email">Email:</label>
        <input type="email" class="form-control" id="email">
      </div>

      <div>
        <label for="pwd">Senha:</label>
        <input type="password" class="form-control" id="pwd">
      </div>

      <div>
        <label><input type="checkbox">Lembrar</label>
      </div>

      <button type="submit" class="btn btn-primary">Enviar</button>
    </form>

  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
JEW9xMcG8R+phH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
    crossorigin="anonymous"></script>
</body>
</html>
```

Código-fonte 21 – Apresentação de um ícone de exclamação e a classe de alert
Fonte: Elaborado pelo autor (2021)

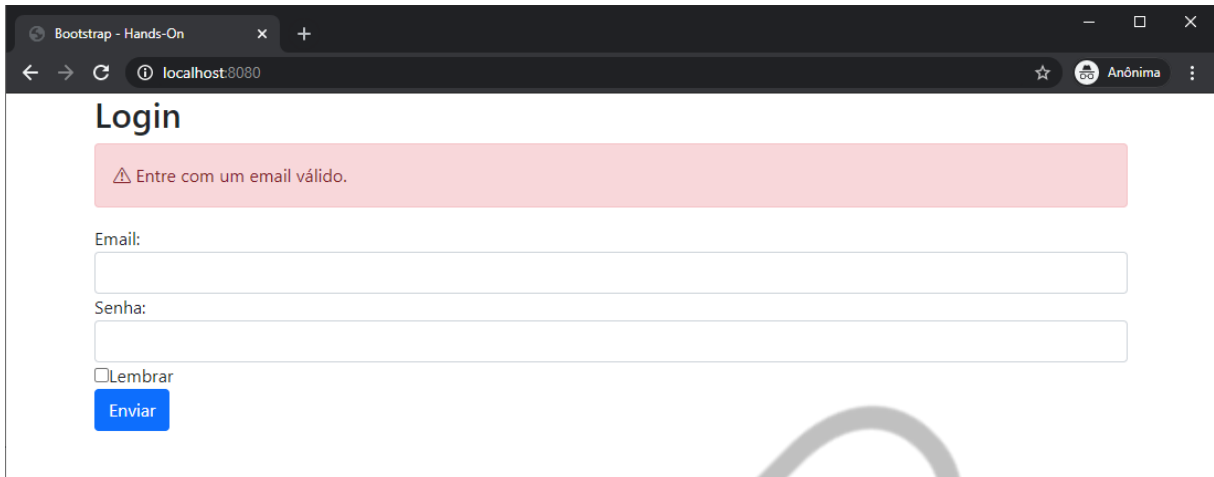
A screenshot of a web browser window titled "Bootstrap - Hands-On" showing a login form. The browser's address bar displays "localhost:8080". The form has a title "Login". Below the title is a red alert box with a warning icon and the text "Entre com um email válido.". Below the alert are two input fields: "Email:" and "Senha:". There is a checkbox labeled "Lembrar" and a blue button labeled "Enviar".

Figura 32 – Exemplo de formulário com alert e ícone
Fonte: Elaborado pelo autor (2021)

Para a utilização dos novos ícones do Bootstrap é necessário incluir o link com a folha de estilo de ícones:

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-  
icons@1.4.1/font/bootstrap-icons.css">
```

Cada ícone é representado pelo sufixo “bi-” e classe “bi” (Bootstrap Icon).

Exemplo: `<i class="bi bi-check-circle"></i>` vai apresentar o ícone de OK.



Figura 33 – Ícone que foi aplicado no exemplo
Fonte: Elaborado pelo autor (2021)

Da mesma forma que apresentamos ícones dentro de um alert, podemos fazer isso em um input do tipo button, text, email e password.

```
<!doctype html>
<html lang="pt-br">

<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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
    crossorigin="anonymous" />

  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.4.1/font/bootstrap-icons.css">

  <title>Bootstrap - Hands-On</title>
</head>

<body>
  <div class="container">
    <h2>Login</h2>

    <div class="alert alert-success" role="alert">
      <i class="bi bi-check-circle"></i>
      Login realizado com sucesso.
    </div>

    <form>
      <div>
        <label for="email">Email:</label>
        <input type="email" class="form-control" id="email">
      </div>

      <div>
        <label for="pwd">Senha:</label>
        <input type="password" class="form-control" id="pwd">
      </div>

      <div>
        <label><input type="checkbox">Lembrar</label>
      </div>

      <button type="submit" class="btn btn-
primary">Enviar</button>
    </form>

  </div>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdaHnUtxwoG2vI5DkLtS3qm9Ekf"
  crossorigin="anonymous"></script>
</body>
</html>
```

Código-fonte 22 – Formulário com a iconografia aplicada aos elementos
Fonte: Elaborado pelo autor (2021)

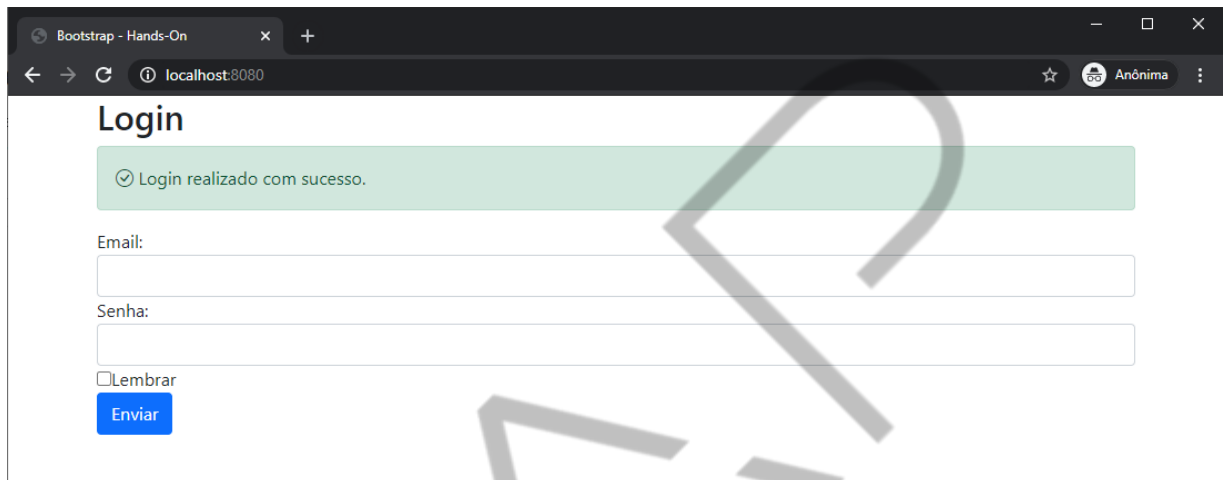


Figura 34 – Formulário com iconografia sendo aplicada
Fonte: Elaborado pelo autor (2021)

Para finalizar, vou apresentar algumas classes de destaque bastante utilizadas, começando pela COLLAPSE em conjunto com a CARD.

```
<!doctype html>
<html lang="pt-br">

<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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
  crossorigin="anonymous" />

  <title>Bootstrap - Hands-On</title>
</head>

<body>
  <div class="container">
    <h2>Collpase + Card</h2>
```

```

<p>
  <a class="btn btn-primary" data-bs-toggle="collapse"
href="#collapseExemplo" role="button" aria-expanded="false" aria-
controls="collapseExemplo">
    Expandir / contrair conteúdo
  </a>
</p>
<div class="collapse" id="collapseExemplo">
  <div class="card card-body">
    Conteúdo que vai ser exibido após a expansão do Card.
  </div>
</div>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
  crossorigin="anonymous"></script>
</body>
</html>

```

Código-fonte 23 – Painei Collapse
Fonte: Elaborado pelo autor (2021)

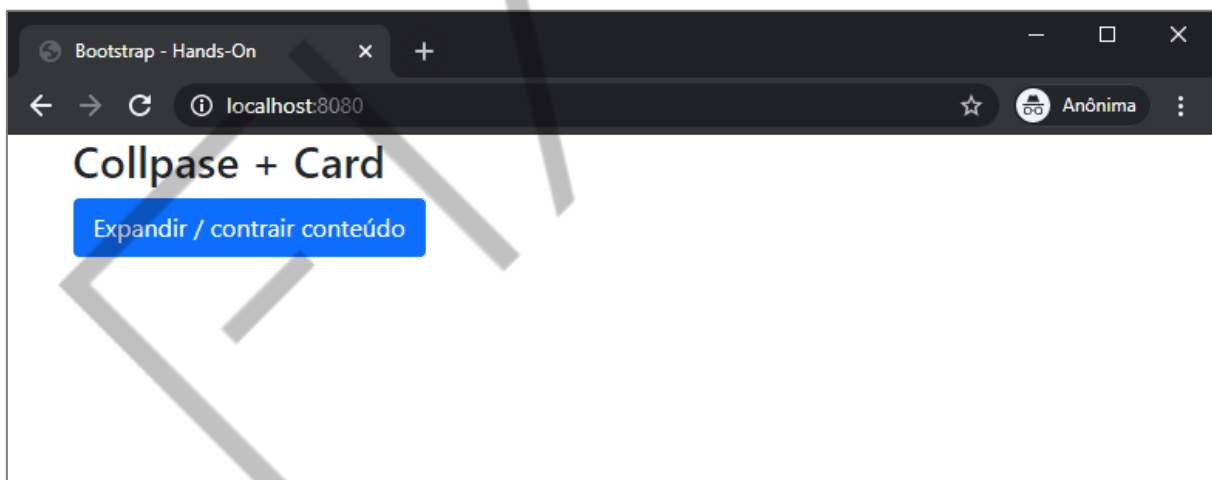


Figura 35 – Collapse + Card - Conteúdo contraído
Fonte: Elaborado pelo autor (2021)

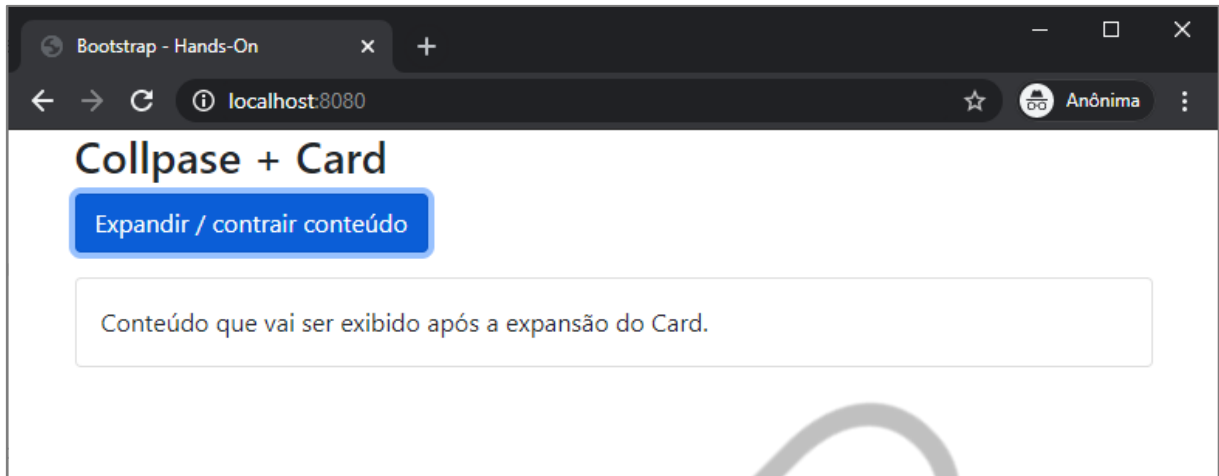


Figura 36 – Collapse + Card - Conteúdo expandido
Fonte: Elaborado pelo autor (2021)

Algumas classes podem ser utilizadas para apresentar informações importantes que necessitam ser destacadas.

```
<!doctype html>
<html lang="pt-br">

<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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgw
ra6" crossorigin="anonymous" />

  <title>Bootstrap - Hands-On</title>
</head>

<body>
  <div class="container">
    <h2>Alertas</h2>

    <div class="alert alert-success"
role="alert">Sucesso</div>
    <div class="alert alert-info"
role="alert">Informação</div>
    <div class="alert alert-warning"
role="alert">Atenção</div>
    <div class="alert alert-danger" role="alert">Erro /
Alerta</div>
  </div>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9
Ekf"
  crossorigin="anonymous"></script>
</body>
</html>
```

Código-fonte 24 – Tipos de alertas
Fonte: Elaborado pelo autor (2021)

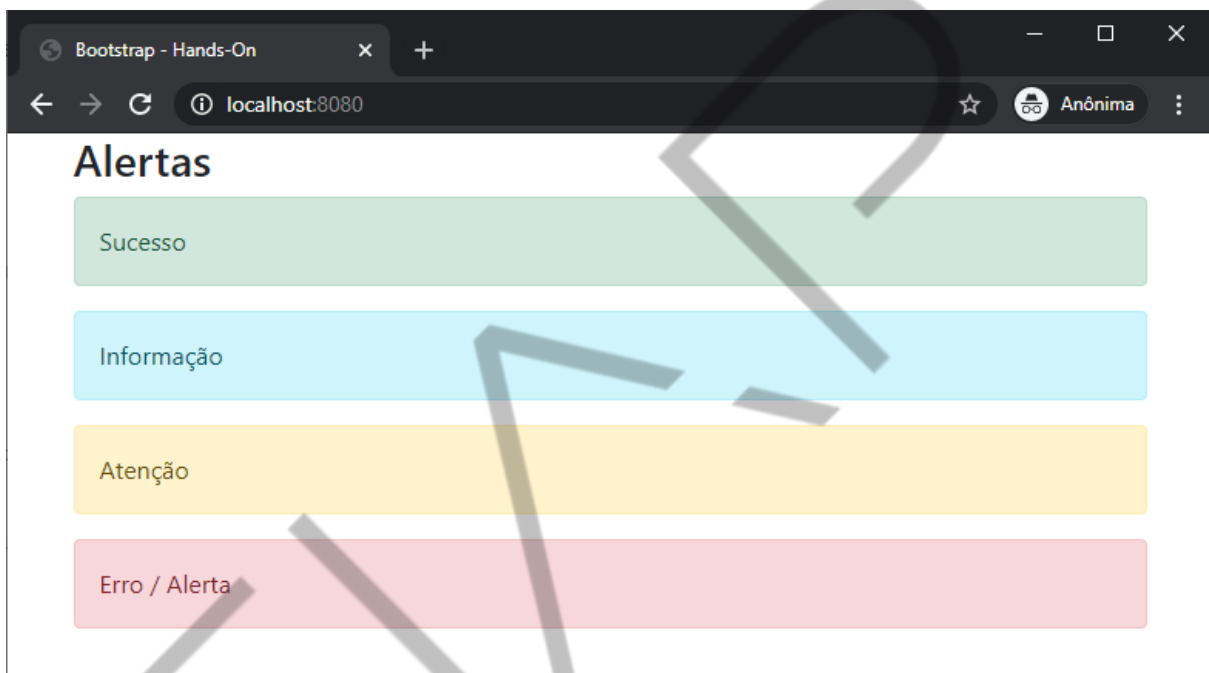


Figura 37 – Tipos de alertas
Fonte: Elaborado pelo autor (2021)

Alerts são utilizados constantemente como feedback ao usuário. Outro conjunto de classes úteis são as “badge” mais a “bg-{cor}” ({cor} é o padrão de cor do Bootstrap que deseja utilizar), que geram ênfase em algum conteúdo.

```
<!doctype html>
<html lang="pt-br">

<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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
```



```

    integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous" />

<title>Bootstrap - Hands-On</title>
</head>

<body>
  <div class="container">
    <h2>Badges</h2>

    <ul class="list-group">
      <li class="list-group-item">
        <span class="badge bg-primary">14</span>
        Produto 1
      </li>
      <li class="list-group-item">
        <span class="badge bg-secondary">5</span>
        Produto 2
      </li>
      <li class="list-group-item">
        <span class="badge bg-info">45</span>
        Produto 3
      </li>
    </ul>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
    crossorigin="anonymous"></script>
</body>
</html>

```

Código-fonte 25 – Badges – ênfase em elementos de span
 Fonte: Elaborado pelo autor (2021)

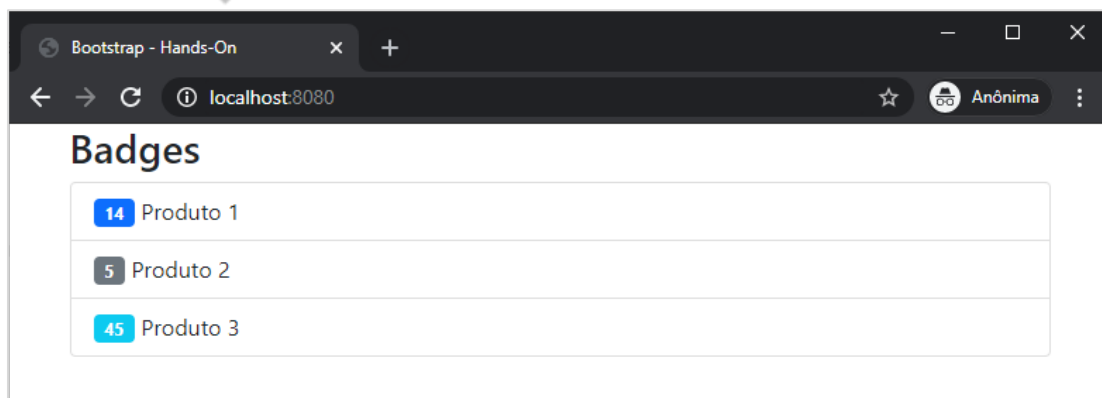


Figura 38 – Badges – ênfase em elementos de span
 Fonte: Elaborado pelo autor (2021)

Os Badges são usados para dar ênfase a elementos que precisam ser destacados isoladamente.

```
<!doctype html>
<html lang="pt-br">

<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-
beta3/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgw
ra6" crossorigin="anonymous" />

  <title>Bootstrap - Hands-On</title>
</head>

<body>
  <div class="container">
    <h2>Cards</h2>

    <div class="card mb-3">
      <div class="card-header">Padrão</div>
      <div class="card-body">
        Conteúdo...
      </div>
    </div>

    <div class="card mb-3">
      <div class="card-header bg-primary text-
light">Informações Básicas</div>
      <div class="card-body">
        Conteúdo...
      </div>
    </div>

    <div class="card mb-3">
      <div class="card-header bg-success text-
light">Sucesso</div>
      <div class="card-body">
        Conteúdo...
      </div>
    </div>

    <div class="card mb-3">
```

```
<div class="card-header bg-info">Informações</div>
<div class="card-body">
  Conteúdo...
</div>
</div>

<div class="card mb-3">
  <div class="card-header bg-warning">Atenção</div>
  <div class="card-body">
    Conteúdo...
  </div>
</div>

<div class="card">
  <div class="card-header bg-danger text-light">Alerta /
Erro</div>
  <div class="card-body">
    Conteúdo...
  </div>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9
Ekf"
  crossorigin="anonymous"></script>
</body>
</html>
```

Código-fonte 26 – Cards personalizados
Fonte: Elaborado pelo autor (2021)

O uso de cards em conjunto com as classes “bg-{cor}”, “text-{cor}” e as classes “card-header” e “card-body” se tornam muito úteis na apresentação de informações. As informações organizadas por meio de cartões são mais fáceis de entendimento e visualização pelos usuários.

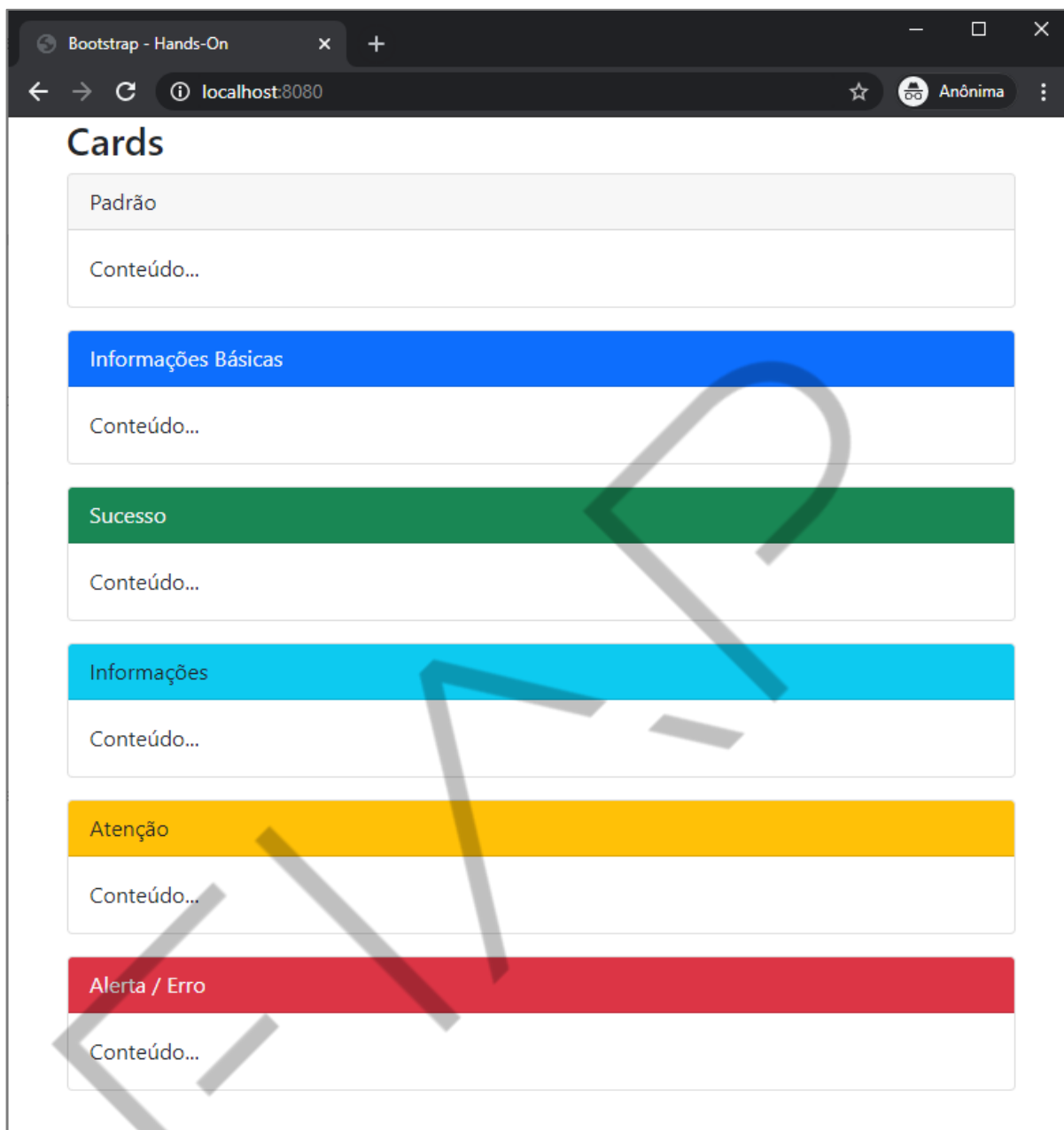


Figura 39 – Cards personalizados
Fonte: Elaborado pelo autor (2021)

REFERÊNCIAS

BOOSTRAP. **CSS Global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system.** 2016. Disponível em: <<http://getbootstrap.com/css/>>. Acesso em: 13 abr. 2021.

ENVATOTUTS+. **Dica rápida: não esqueça da meta tag de viewport.** 2013. Disponível em: <<https://webdesign.tutsplus.com/pt/articles/quick-tip-dont-forget-the-viewport-meta-tag--webdesign-5972>>. Acesso em: 13 abr. 2021.

UTTERBACK, B. **O que é o Bootstrap? – Verdades e mitos:** Parte 1 de 2. 2014. Disponível em: <<https://www.prestashop.com/blog/pt/2014/03/06/o-que-e-o-bootstrap-verdades-e-mitos-parte-1-de-2/>>. Acesso em: 13 abr. 2021.

GLOSSÁRIO

Framework	Conjunto abstrato de tecnologias utilizadas no desenvolvimento de projetos.
Responsividade	É a ação de o navegador se adaptar ao seu site sem a necessidade de criar diversas folhas de estilos.
Viewport	É o espaço em que será apresentado o conteúdo do site.
Indentação	É o que se aplica aos códigos-fontes para que fiquem legíveis.
Inteligível	Algo que seja claro, fácil de entender e interpretar.
Boilerplate	Exemplo básico de página HTML5, utilizando o framework Bootstrap.