

Criação Rápida de *Sites Responsivos* com o **Bootstrap**



Desenvolva *sites* e aplicações Web de forma simples, funcional e sofisticada

Conheça o Bootstrap: o projeto Web mais popular do GitHub

4 Projetos completos explicados passo a passo

Introdução ao Sass e à *flexbox*

EDIÇÃO

FCA – Editora de Informática, Lda.
Av. Praia da Vitória, 14 A – 1000-247 Lisboa
Tel: +351 213 511 448
fca@fca.pt
www.fca.pt

DISTRIBUIÇÃO

Lidel – Edições Técnicas, Lda.
Rua D. Estefânia, 183, R/C Dto. – 1049-057 Lisboa
Tel: +351 213 511 448
lidel@lidel.pt
www.lidel.pt

LIVRARIA

Av. Praia da Vitória, 14 A – 1000-247 Lisboa
Tel: +351 213 511 448 * Fax: +351 213 522 684
livraria@lidel.pt

Copyright © 2017, FCA – Editora de Informática, Lda.
ISBN edição impressa: 978-972-722-867-6
1.ª edição impressa: maio 2017

Impressão e acabamento: Tipografia Lousanense, Lda. – Lousã
Depósito Legal n.º 426368/17
Capa: José M. Ferrão – *Look-Ahead*

Marcas Registadas de FCA – Editora de Informática, Lda. –



Todos os nossos livros passam por um rigoroso controlo de qualidade, no entanto aconselhamos a consulta periódica do nosso *site* (www.fca.pt) para fazer o *download* de eventuais correções.

Não nos responsabilizamos por desatualizações das hiperligações presentes nesta obra, que foram verificadas à data de publicação da mesma.

Os nomes comerciais referenciados neste livro têm patente registada.



Reservados todos os direitos. Esta publicação não pode ser reproduzida, nem transmitida, no todo ou em parte, por qualquer processo eletrónico, mecânico, fotocópia, digitalização, gravação, sistema de armazenamento e disponibilização de informação, *síto Web*, *blogue* ou outros, sem prévia autorização escrita da Editora, exceto o permitido pelo CDADC, em termos de cópia privada pela AGE COP – Associação para a Gestão da Cópia Privada, através do pagamento das respetivas taxas.

ÍNDICE GERAL

0. INTRODUÇÃO	1
0.1. Um pouco de história	1
0.2. A guerra dos <i>browsers</i>	2
0.3. <i>Browsers mobile</i>	4
0.4. <i>Responsive Web Design</i>	4
0.5. <i>Frameworks Web</i>	6
0.6. Estrutura do livro.....	8
1. BOOTSTRAP	11
1.1. Introdução	11
1.2. Por que usar o Bootstrap?	12
1.3. Ferramentas de desenvolvimento	13
1.4. <i>Download</i>	14
1.4.1. Acesso local	14
1.4.2. Acesso via CDN	15
1.4.3. Local ou CDN?.....	16
1.5. A minha primeira página Bootstrap.....	17
2. CSS	21
2.1. Sistema de <i>grid</i>	21
2.1.1. Classe <i>container</i>	21
2.1.2. <i>Breakpoints</i>	22
2.1.3. Linhas e colunas	23
2.1.4. Combinação de classes	26
2.1.5. Quebra de colunas	26
2.1.6. Deslocamento de colunas	27
2.1.7. Colunas aninhadas	27
2.1.8. Ordenação de colunas	28
2.1.9. Classes responsivas	28
2.2. Tipografia.....	30
2.2.1. Corpo	30
2.2.2. Cabeçalhos.....	30
2.2.3. Textos <i>inline</i>	31
2.2.4. Abreviaturas, endereços e citações.....	33
2.2.5. Listas	35
2.2.6. Código.....	35
2.3. Tabelas.....	36
2.4. Imagens	39
2.5. Formulários.....	39
2.5.1. <i>Layouts</i>	40

2.5.2. Elementos de formulário	42
2.5.2.1. Elemento <input>	42
2.5.2.2. Elemento <textarea>	44
2.5.2.3. Elemento <select>	44
2.5.3. Estados	45
2.5.3.1. Validação com cores	45
2.5.3.2. Validação com ícones	46
2.5.4. Dimensionamento	47
2.6. Botões	48
2.6.1. Dimensões	49
2.6.2. Estados dos botões	49
2.6.3. Ícones de fecho e de <i>dropdown</i>	50
3. COMPONENTES.....	51
3.1. Categoria Texto	51
3.1.1. Componentes <i>Jumbotron</i> e <i>PageHeader</i>	51
3.1.2. Componentes <i>Label</i> e <i>Badge</i>	52
3.1.3. Componente <i>Alert</i>	53
3.1.4. Componente <i>Panel</i>	54
3.2. Categoria <i>Media</i>	56
3.2.1. Ícones	57
3.2.2. <i>Thumbnails</i>	58
3.2.3. Componente <i>MediaObject</i>	60
3.2.4. <i>Media</i> responsivo	62
3.2.5. Componente <i>Dropdown</i>	63
3.2.6. Componente <i>ProgressBar</i>	65
3.3. Categoria Navegação.....	66
3.3.1. Componentes <i>Tabs</i> e <i>Pills</i>	67
3.3.2. Componente <i>Breadcrumb</i>	69
3.3.3. Componente <i>Navbar</i>	69
3.3.4. Componente <i>Pager</i>	73
3.4. Categoria Agrupamentos.....	74
3.4.1. Grupos de botões.....	75
3.4.2. Grupos de <i>inputs</i>	76
3.4.3. Grupos de itens de lista	77
4. PLUGINS	81
4.1. Introdução.....	81
4.2. <i>Plugin Modal</i>	82
4.3. <i>Plugin Dropdown</i>	85
4.4. <i>Plugin Scrollspy</i>	87
4.5. <i>Plugin Tab</i>	88
4.6. <i>Plugin Tooltip</i>	89
4.7. <i>Plugin Popover</i>	91
4.8. <i>Plugin Button</i>	92
4.9. <i>Plugin Collapse</i>	93

4.10. <i>Plugin Carousel</i>	95
4.11. <i>Plugin Affix</i>	98
5. PROJETO I: SITE PESSOAL	101
5.1. Introdução	101
5.2. Página inicial	102
5.3. Secção “Quem sou eu?”	103
5.4. Secção “O que faço?”	106
5.5. Secção “Hobbies”	108
5.6. Barra de navegação	109
5.7. Rodapé	111
5.8. Configurações finais	113
6. PROJETO II: SITE EMPRESARIAL	115
6.1. Introdução	115
6.2. Página inicial	117
6.3. Barra de navegação	118
6.3.1. Criação da barra	118
6.3.2. Estilização da barra	119
6.3.3. Adição do plugin <i>Scrollspy</i>	120
6.4. <i>Slider</i> de imagens	121
6.4.1. Estilização do <i>slider</i>	123
6.5. Secção “Equipa”	125
6.5.1. Texto introdutório	125
6.5.2. Fotos dos colaboradores	125
6.6. Secção “Produtos”	128
6.6.1. Criação de painéis	129
6.6.2. Criação da janela modal	131
6.7. Secção “Contactos”	134
6.7.1. Criação de formulário	134
6.7.2. Criação do mapa	135
6.8. Secção “Rodapé”	137
6.9. Configurações finais	138
7. PROJETO III: LOJA ONLINE	141
7.1. Introdução	141
7.2. Página principal: “Catálogo de produtos”	142
7.2.1. Barra de navegação	143
7.2.2. Categorias de produtos	147
7.2.3. Catálogo de produtos	148
7.3. Página de detalhe de item	150
7.4. Página “Wish List”	152
7.5. Página de pagamento	155
8. PROJETO IV: PORTFOLIO DO INSTAGRAM	161
8.1. Introdução	161

8.2. Configurações iniciais	162
8.3. Desenvolvimento	163
8.3.1. Cabeçalho	164
8.3.2. <i>Feed</i> do Instagram	165
8.3.3. Rodapé	169
9. BOOTSTRAP 4	171
9.1. Migração para o Bootstrap 4	171
9.2. Introdução ao Sass	173
9.2.1. Instalação	174
9.2.1.1. Linha de comandos	174
9.2.1.2. Aplicação <i>desktop</i>	175
9.2.1.3. Implementação específica do Sass (via LibSass)	176
9.2.1.4. Ambiente <i>online</i> de teste	177
9.2.2. Variáveis	178
9.2.3. Aninhamentos de regras	180
9.2.4. Herança	182
9.2.5. <i>Mixins</i>	183
9.2.6. Funções	184
9.2.7. Estruturas de decisão e cíclicas	185
9.2.8. Formatos de <i>output</i>	186
9.3. Introdução à <i>flexbox</i>	187
9.3.1. <i>Layout</i> automático	188
9.3.2. Altura das colunas	189
9.3.3. Alinhamento	190
9.3.4. Ordenação	192
9.4. <i>Templates</i> e temas	193
ÍNDICE REMISSIVO	199

4

PLUGINS

O Bootstrap inclui vários componentes JavaScript na forma de *plugins*. Estes providenciam elementos gráficos e interativos adicionais, tais como janelas de diálogo e *tooltips*. Neste capítulo destacam-se e exploram-se os *plugins* mais populares do Bootstrap.

4.1 INTRODUÇÃO

O Bootstrap suporta o uso de componentes interativos. Ao descarregar o Bootstrap, obtém-se uma pasta **js** que inclui os *plugins* nativos do Bootstrap em duas versões: uma versão comentada (14 KB) e uma versão minificada (10 KB). Para além deste ficheiro, é necessário incluir a biblioteca **jQuery**. Sendo assim, todos os exemplos deste capítulo deverão ter uma referência para estes dois ficheiros, conforme o próximo excerto de código:

```
<!-- Referência local para ficheiro Bootstrap com os plugins -->
<script type="text/javascript" src="js/bootstrap.js"></script>
<!-- Referência via CDN para ficheiro com a biblioteca jQuery -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">
</script>
```



Consulte a secção 1.4 para encontrar outras formas de referenciar estes ficheiros.

Antes de começar a olhar individualmente para os *plugins* do Bootstrap, tenha em conta que existem duas formas de interagir com os mesmos. Através de:

- **HTML** – uso de um atributo `data-*` no elemento HTML que recebe o *plugin*;
- **jQuery** – uso da biblioteca **jQuery** para invocar os métodos nativos do *plugin*.

Por exemplo, na secção 3.2.5 foi apresentado o modo de usar o componente *Dropdown*, mais concretamente como exibir o menu após clique no botão. Essa ação é gerida pela inclusão do atributo `data-toggle="dropdown"`. A mesma ação poderia ter sido controlada através da invocação do método `dropdown` usando a sintaxe jQuery:

```
$(' .myDropdown' ).dropdown();
```

Apesar da flexibilidade da biblioteca **jQuery**, aconselha-se que todos os *plugins* Bootstrap sejam usados através da API de marcação (atributos `data-*`) sem escrever uma única linha de JavaScript. Neste capítulo será dado destaque apenas a esta abordagem.

4.2 PLUGIN MODAL

O *plugin Modal* é uma caixa de diálogo (ou janela *popup*) que é exibida na parte superior da página atual. O objetivo deste *plugin* é exibir conteúdo de uma fonte separada e permitir ao utilizador interagir com a nova janela sem sair da janela-pai. Habitualmente, este *plugin* é usado para incluir informação adicional após uma determinada ação ou para incorporar um formulário de autenticação ou registo.

Para criar uma janela modal, use o seguinte código:

```
<!-- Janela modal -->
<div id="myModal" class="modal fade" role="dialog">
  <div class="modal-dialog">
    <!--Conteúdo da Modal -->
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">
          &times;
        </button>
        <h4 class="modal-title">Título da Modal</h4>
      </div>
      <div class="modal-body">
        <p>Corpo de texto da Modal.</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-primary" data-
          dismiss="modal">Fechar
        </button>
      </div>
    </div>
  </div>
</div>
```

Para acionar o mecanismo de abertura/fecho da janela modal, é necessário algum tipo de gatilho (*trigger*). Tipicamente, usa-se um botão (ou *link*) de acordo com o próximo código:

```
<!-- Invocar a janela modal através de um botão -->
<button type="button" class="btn btn-info btn-lg"
  data-toggle="modal"
  data-target="#myModal">
  Abrir Modal
</button>
```


Quer seja um botão ou um *link*, deverá incluir no elemento dois atributos `data-*`:

- `data-toggle="modal"` – invoca o método JavaScript nativo para a abertura da janela modal;
- `data-target="#myModal"` – emparelha com a janela modal através do valor do seu atributo `id`.

O resultado no *browser* é o ilustrado na Figura 4.1.

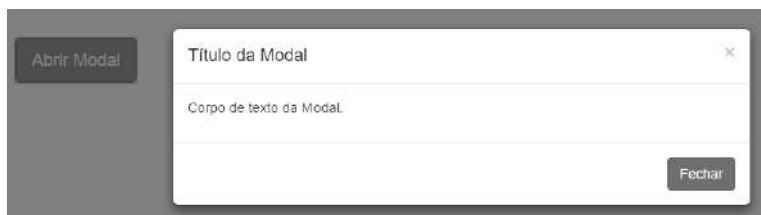


FIGURA 4.1 – *Plugin Modal*

A construção de uma janela modal é baseada numa hierarquia de elementos `<div>` que merece ser explicada. O elemento do tipo *container* contém a classe `.modal`, que identifica o conteúdo da `<div>` como modal e atribui o foco para si. Para fazer um correto emparelhamento entre a janela modal e o botão que a abre, o seu atributo `id` deverá ter o mesmo valor que o atributo `data-target` do botão.

A Tabela 4.1 lista outras classes que poderá usar com este *plugin*.

CLASSE	DESCRIÇÃO
<code>.modal-dialog</code>	Define a largura e a margem ideais para a janela modal
<code>.modal-content</code>	Estiliza a janela com limite, cor de fundo, etc. Deve usar esta classe de forma a poder organizar o conteúdo por: cabeçalho, corpo e rodapé
<code>.modal-header</code>	Define o estilo para o cabeçalho
<code>.modal-title</code>	Define o estilo para o título do cabeçalho
<code>.modal-body</code>	Define o estilo para o corpo
<code>.modal-footer</code>	Define o estilo para o rodapé
<code>.modal-sm</code>	Especifica uma janela modal pequena
<code>.modal-lg</code>	Especifica uma janela modal grande
<code>.fade</code>	Adiciona um efeito de transição que desvanece para dentro e para fora

TABELA 4.1 – Classes do *plugin Modal*

O conteúdo da janela modal deverá ser colocado dentro de um elemento `<div>` com a classe `.modal-content`. Dentro desta `<div>` devem ser adicionados novos *containers* com as classes enumeradas na Tabela 4.1, mais concretamente com:

- Classe `.modal-header` – composta por um cabeçalho estilizado e um botão com a classe `.close` e o atributo `data-dismiss="modal"`, que fecha a modal após clique. O botão contém um **X** definido com a entidade `×`;
- Classe `.modal-body` – onde deve ser adicionado código HTML, tal como parágrafos, formulários, imagens, vídeos, etc;
- Classe `.modal-footer` – onde deverá colocar, pelo menos, um botão que fechará a modal. Note que esta área é, por predefinição, alinhada à direita.

As janelas modais têm outras opções interessantes que podem ser definidas através de atributos `data-*` ou via JavaScript. A Tabela 4.2 mostra as opções disponíveis.

NOME	DESCRIÇÃO
<code>data-backdrop</code>	Especifica se a janela modal deve ter uma sobreposição escura: - <code>True</code> – sobreposição escura (por predefinição); - <code>False</code> – sem sobreposição (transparente); - <code>Static</code> – não é possível fechá-la ao clicar fora dela.
<code>data-keyboard</code>	Especifica se a janela modal pode ser fechada com a tecla <code>Esc</code> : - <code>True</code> – pode ser fechada com <code>Esc</code> (por predefinição); - <code>False</code> – não pode ser fechada com <code>Esc</code> .
<code>show</code>	Especifica se deve ser exibida a janela modal quando inicializada

TABELA 4.2 – Opções gerais do *plugin Modal*

O próximo exemplo (Figura 4.2) mostra uma janela modal pequena, invocada sem sobreposição escura (transparente) através da inclusão, no elemento `<button>`, do atributo `data-backdrop=false`:

```
<button ...
  data-backdrop="false"
  data-toggle="modal"
  data-target="#myModal">...
</button>
```

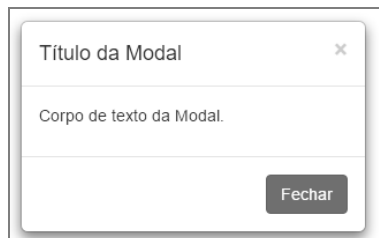


FIGURA 4.2 – Janela modal

4.3 PLUGIN DROPDOWN

O *plugin Dropdown* permite adicionar um menu suspenso a barras de navegação (*navbars*), separadores (*tabs*) e listas de *links* simples (*pills*). Para definir um *dropdown*, agregue todo o conteúdo num elemento `<div>` com a classe `.dropdown`. Depois, na sua construção, pode usar as classes apresentadas na Tabela 4.3.

CLASSE	DESCRIÇÃO
<code>.dropdown-menu</code>	Cria o menu
<code>.dropdown-menu-right</code>	Alinha o menu à direita
<code>.dropdown-header</code>	Inclui um cabeçalho dentro do menu
<code>.dropup</code>	Define um menu com abertura para cima
<code>.disabled</code>	Desativa um item do menu
<code>.divider</code>	Separa itens do menu com uma linha horizontal

TABELA 4.3 – Classes do *plugin Dropdown*

Para abrir o menu, use um botão (ou *link*) com a classe `.dropdown-toggle` e com o atributo `data-toggle="dropdown"`.

Este tipo de menus já foi abordado no Capítulo 3, pelo que podemos apresentar agora dois exemplos práticos. O primeiro adiciona um menu suspenso a um botão na barra de navegação:

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Nome do Site</a>
    </div>
    <div>
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li class="dropdown">
          <a class="dropdown-toggle" data-toggle="dropdown" href="#">Pág. 1
            <span class="caret"></span>
          </a>
          <ul class="dropdown-menu">
            <li><a href="#">Pág. 1.1</a></li>
            <li><a href="#">Pág. 1.2</a></li>
            <li><a href="#">Pág. 1.3</a></li>
          </ul>
        </li>
        <li><a href="#">Pág. 2</a></li>
        <li><a href="#">Pág. 3</a></li>
      </ul>
    </div>
  </div>
</nav>
```

O resultado renderizado no *browser* é similar ao apresentado na Figura 4.3.

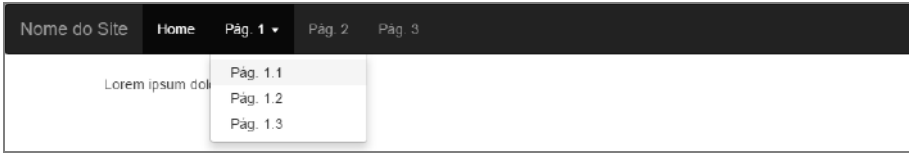


FIGURA 4.3 – Barra de navegação com menu suspenso

O segundo exemplo adiciona um menu suspenso com um formulário de autenticação à barra de navegação:

```
<nav class="navbar navbar-inverse">
... /* mesmo código do anterior */
<ul class="nav navbar-nav navbar-right">
<li class="dropdown">
<a class="dropdown-toggle" data-toggle="dropdown" href="#">Login
<span class="glyphicon glyphicon-log-in"></span></a>
<div class="dropdown-menu">
<form id="formlogin" class="form container-fluid">
<div class="form-group">
<label for="usr">Nome:</label>
<input type="text" class="form-control" id="usr">
</div>
<div class="form-group">
<label for="pwd">Password:</label>
<input type="password" class="form-control" id="pwd">
</div>
<button type="button" id="btnLogin" class="btn btn-
block">Login</button>
</form>
<div class="container-fluid">
<a class="small" href="#">Esqueceu password?</a>
</div>
</div>
</li>
</ul>
</nav>
```

O resultado renderizado no *browser* é similar ao apresentado na Figura 4.4.



FIGURA 4.4 – Barra de navegação com menu suspenso e formulário de autenticação

4.4 PLUGIN SCROLLSPY

O *plugin Scrollspy* é usado para atualizar automaticamente os *links* numa lista de navegação com base na posição de rolagem. O próximo exemplo mostra como criar um *plugin* deste tipo:

```
<!-- Área de scroll -->
<body data-spy="scroll" data-target=".navbar" data-offset="50">
  <!-- A navbar com <a> para aceder às secções na área de scroll -->
  <nav class="navbar navbar-inverse navbar-fixed-top">
    ...
    <ul class="nav navbar-nav">
      <li><a href="#section1">Secção 1</a></li>
      ...
    </ul>
  </nav>
<!-- Secção 1 -->
<div id="section1">
  <h1>Secção 1</h1>
  <p>Use o scroll nesta página e observe a barra de navegação!</p>
</div>
  ...
</body>
```

Para acionar o mecanismo *scrollspy*, basta declarar os atributos `data-spy` e `data-target` para o elemento cuja barra de rolagem se pretende monitorizar. Sendo assim, adicione um atributo `data-spy="scroll"` ao elemento que deve ser usado como a área de rolagem (muitas vezes, este é o elemento `<body>`). Em seguida, adicione o atributo `data-target` com um valor do `id` ou o nome da classe da barra de navegação (`.navbar`). Esta ação garante que a barra de navegação está conectada com a área de rolagem.

Note ainda que os elementos roláveis devem corresponder ao `id` dos *links* dos itens da lista da barra de navegação, mais concretamente `<div id="section1">` emparelha com ``.

Outras opções podem ser passadas por meio de atributos de dados ou do JavaScript. O atributo opcional `data-offset` especifica o número de *pixels* (*px*) a serem deslocados de cima ao calcular a posição de rolagem. Isso é útil quando se nota que os *links* dentro da barra de navegação alteram o estado ativo muito cedo ou muito tarde quando saltam para os elementos roláveis. O valor, por predefinição, é de 10 *px*.



Para funcionar corretamente, o elemento com `data-spy="scroll"` requer a propriedade CSS `position` com o valor `"relative"`.

O resultado renderizado no *browser* é similar ao apresentado na Figura 4.5. À medida que usa o *scroll* para ver o conteúdo da página, a barra de navegação vai realçando a secção atual.

FIGURA 4.5 – *Plugin Scrollspy*

4.5 PLUGIN TAB

O *plugin Tab* é usado para separar o conteúdo em diferentes painéis, em que apenas um painel é visível de cada vez. Este *plugin* estende o componente de navegação com a capacidade para adicionar áreas específicas a cada separador.

O próximo exemplo (Figura 4.6) mostra como criar um cenário para uso dos separadores:

```
<ul class="nav nav-tabs">
  <li class="active"><a data-toggle="tab" href="#home">Home</a></li>
  <li><a data-toggle="tab" href="#menu1">Menu 1</a></li>
  <li><a data-toggle="tab" href="#menu2">Menu 2</a></li>
  <li><a data-toggle="tab" href="#menu3">Menu 3</a></li>
</ul>
<div class="tab-content">
  <div id="home" class="tab-pane fade in active">
    <h3>HOME</h3>
    <p>Esta é a página HOME.</p>
  </div>
  <div id="menu1" class="tab-pane fade">
    <h3>Menu 1</h3>
    <p>Esta é a página do MENU 1.</p>
  </div>
  <div id="menu2" class="tab-pane fade">
    <h3>Menu 2</h3>
    <p>Esta é a página do MENU 2.</p>
  </div>
  <div id="menu3" class="tab-pane fade">
    <h3>Menu 3</h3>
    <p>Esta é a página do MENU 3.</p>
  </div>
</div>
```

Comece por adicionar os separadores a uma lista `<ul class="nav nav-tabs">`. De forma a tornar as guias alternáveis, adicione a cada *link* o atributo `data-toggle="tab"`. Em seguida, crie um elemento `<div>` e associe a classe `.tab-content`. Depois, adicione o painel específico para cada separador através de um elemento `<div>` e associe a classe `.tab-pane`.

Este elemento `<div>` deve possuir um atributo `id` que vai emparelhar com o atributo `href` de cada *link* do *tab*. Se quiser ainda que os separadores façam *fade in* e *fade out* ao clicar neles, combine a classe `.fade` com a `.tab-pane`. O primeiro painel também deve ter a classe `.in` para tornar visível o conteúdo inicial.



FIGURA 4.6 – *Plugin Tab*

4.6 PLUGIN TOOLTIP

O *plugin Tooltip* é uma pequena caixa *popup* que aparece quando o utilizador move o ponteiro do rato sobre um elemento. Para criar um *plugin* deste tipo, adicione o atributo `data-toggle="tooltip"` a um elemento e use o atributo `title` para especificar o texto que deve ser exibido dentro do *tooltip*:

```
<a href="#"
  data-toggle="tooltip"
  title="Olá, eu sou um tooltip!">Passe o rato sobre mim</a>
```



Por questões de *performance*, os *plugins Tooltip* e *Popover* devem ser inicializados manualmente. Uma forma compacta de inicializar todos os *tooltips* de uma página é seleccioná-los pelo seu atributo `data-toggle`:

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

Por predefinição, o *tooltip* aparecerá na parte superior do elemento (Figura 4.7).

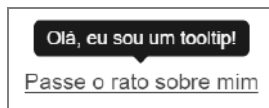


FIGURA 4.7 – *Plugin Tooltip*

A Tabela 4.4 apresenta algumas das principais opções disponíveis que podem ser passadas por meio de atributos de dados.

NOME	DESCRIÇÃO
<code>data-animation</code>	Especifica se é necessário adicionar um efeito de transição de <i>fade</i> CSS ao exibir e ocultar o <i>tooltip</i> : - <code>True</code> – adiciona um efeito de desvanecimento (por predefinição); - <code>False</code> – não adiciona um efeito de desvanecimento.
<code>data-delay</code>	Especifica o número de milissegundos (ms) que serão necessários para mostrar e ocultar o <i>tooltip</i> . Para especificar um atraso para exibir e outro para ocultar, use a estrutura do objeto: - <code>delay: {show: 500, hide: 100}</code> – que demorará 500 ms para mostrar o <i>tooltip</i> , mas apenas 100 ms para ocultá-lo.
<code>data-html</code>	Especifica se as <i>tags</i> HTML podem ser aceites no atributo <code>title</code> do <i>tooltip</i> : - <code>True</code> – aceitar <i>tags</i> HTML; - <code>False</code> – não aceitar <i>tags</i> HTML (por predefinição).
<code>data-placement</code>	Especifica a posição do <i>tooltip</i> . Os valores possíveis são: - <code>"top"</code> – parte superior (por predefinição); - <code>"bottom"</code> – parte inferior; - <code>"left"</code> – lado esquerdo; - <code>"right"</code> – lado direito; - <code>"auto"</code> – permite ao navegador decidir a posição do <i>tooltip</i> . Por exemplo, se o valor for <code>"auto left"</code> , será exibido no lado esquerdo quando possível; se o valor for <code>"auto right"</code> , será exibido à direita.

TABELA 4.4 – Opções gerais do *plugin Tooltip*

O próximo exemplo combina várias opções do *plugin* para personalizar um *tooltip* (Figura 4.8):

```
<button class="btn btn-success btn-md"
  data-toggle="tooltip" data-placement="bottom" data-html="true"
  title="<h1><strong>HTML</strong> dentro <code>da</code>
  <em>tooltip</em></h1>">Passe o rato sobre mim</button>
<script>
$(document).ready(function() {
  $('<code>.btn</code>').tooltip();
});
</script>
```



FIGURA 4.8 – *Tooltip* personalizado

8

PROJETO IV: *PORTFOLIO* DO INSTAGRAM

Neste capítulo explica-se, passo a passo, a criação de um *portfolio* simples baseado no conteúdo de um perfil no Instagram. O destaque vai para o uso do Bootstrap combinado com um *plugin* JavaScript para obter conteúdo personalizado daquela rede social.

8.1 INTRODUÇÃO

Neste capítulo poderá acompanhar o processo de criação de um *site* que integra um *portfolio* de imagens selecionadas do Instagram, chamado **MyInstagram**. O principal objetivo da sua criação é apresentar um subconjunto das imagens submetidas na rede social Instagram através de uma seleção baseada em *hashtags*. O resultado final deverá assemelhar-se ao apresentado na Figura 8.1.



FIGURA 8.1 – *Layout* da página principal do *site* MyInstagram

O processo construtivo deste projeto permitirá ao leitor consolidar os seus conhecimentos nos seguintes tópicos:

- Gestão básica da *grid* do Bootstrap;
- Uso do pré-processador **Sass** (introduzido na secção 9.2);
- Uso de *plugins* JavaScript para extração de conteúdo do Instagram (por exemplo, o **Instafeed.js**¹).

Em seguida são apresentados os detalhes para a criação da página.

8.2 CONFIGURAÇÕES INICIAIS

Para este projeto serão necessários os seguintes componentes:

- Pré-processador **Sass**;
- Bootstrap (versão Sass);
- Biblioteca **Instafeed.js**.

Comece por instalar o **Sass** no seu computador. Existem várias formas de o fazer; uma delas, para todos aqueles que não sejam adeptos da linha de comandos, é através da instalação de uma aplicação-cliente com uma interface gráfica que permita inspecionar e gerar folhas de estilo CSS a partir de ficheiros **Sass** – neste projeto vamos usar o **Koala**.

Em seguida, descarregue o Bootstrap (versão Sass). Finalmente, é necessário um *plugin* para auxiliar na comunicação com o Instagram e obter assim imagens desta rede social na página HTML do projeto. Para conseguir fazer solicitações ao Instagram, será utilizada a biblioteca **Instafeed.js**.

Relativamente à estrutura do projeto, crie uma pasta chamada **MyInstagram** e três subpastas: **css**, **js** e **images**. Na primeira, inclua o ficheiro **bootstrap.scss** e a pasta **bootstrap** obtidas a partir do *download* do Bootstrap. Na pasta **js**, inclua os ficheiros **bootstrap.js** e **instafeed.min.js**.

Abra o **Koala** e adicione a pasta do projeto; em seguida, abra o seu editor preferido: em primeiro lugar, adicione um ficheiro na pasta **css** chamado **style.scss** e, depois, adicione um ficheiro chamado **app.js** dentro da pasta **js**.

¹ Link: <http://instafeedjs.com>.

8.3 DESENVOLVIMENTO

Crie na pasta do projeto um ficheiro chamado **index.html** com o seguinte código HTML:

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width,user-scalable=no">
  <title>Instagram Portfolio</title>
  <link
    href='http://fonts.googleapis.com/css?family=Lato:400,700|Kaushan+Sc
    ript|Montserrat' rel='stylesheet' type='text/css'>
  <link rel="stylesheet" type="text/css" href="css/style.css">
  <script src="js/jquery.min.js"></script>
  <script type="text/javascript" src="js/bootstrap.js"></script>
  <script type="text/javascript" src="js/instafeed.min.js"></script>
  <script type="text/javascript" src="js/app.js"></script>
</head>
<body>
  <header>
    <!-- Cabeçalho -->
  </header>
  <section class="instagram-wrap">
    <!-- Feed do Instagram -->
  </section>
  <section class="footer-bottom">
    <!-- Rodapé -->
  </section>
</body>
</html>
```

O elemento `<head>` inclui, para além de outros componentes, um subelemento `<link>` para as fontes da biblioteca **Google Fonts** que vamos utilizar. O elemento `<body>` inclui três elementos semânticos. Foram atribuídas classes autodescritivas para os elementos `<section>` de forma a facilitar o entendimento da função daquele objeto.



Parte-se do pressuposto de que descarregou a biblioteca **jQuery** e incluiu o ficheiro principal na pasta **js** do projeto.

Abra o ficheiro **style.scss** e inclua o seguinte código:

```
$main-font: 'Lato', sans-serif;
$sub-font: 'Montserrat', sans-serif;
$fancy-font: 'Kaushan Script', cursive;
$font-size: 16px;

$black: #000;
$white: #FFF;
$grey: #585c65;
$mediumgrey: #9b9b9b;
$lightgrey: #eeeeee;
$blue: #3466a1;
```

```
$padding: 10px;
$margin: 10px;

$font-family-sans-serif: $main-font;
@import 'bootstrap';

body {
  color: $grey;
  font-size: $font-size;
}

header {
  max-height: 600px;
  height: 600px;
  overflow: hidden;
}
```

No excerto anterior são adicionadas variáveis para as fontes, cores e margens. Em seguida, atribui-se a fonte `$main-font` à variável-padrão de fonte no Bootstrap, `$font-family-sans-serif`. Depois, usa-se o recurso `@import` no **Sass** para importar o **bootstrap.scss**. Finalmente, adicionam-se duas regras para o corpo da página HTML e para o seu elemento `<header>`.

Ao gravar o ficheiro, o **Koala** vai automaticamente compilar o ficheiro **style.scss** para o ficheiro de estilo final, **style.css**.

8.3.1 CABEÇALHO

No elemento `<header>` do ficheiro HTML inclua uma referência a uma imagem de topo e um elemento `<div>` com o texto do cabeçalho:

```

<div class="titles">
  <h1>MyInstagram</h1>
  <h2>Ricardo Queiros</h2>
</div>
```

Abra o ficheiro **styles.scss** e adicione regras para os elementos do cabeçalho:

```
h1 {
  text-transform: uppercase;
  font-weight: 700;
  font-size: 36px;
  letter-spacing: 0.06em;
  margin-bottom: $margin;
}

h2 {
  font-size: 16px;
  width: 70%;
}
```

```
    margin: 0 auto;
  }
  .titles {
    position: absolute;
    bottom: 40%;
    width: 100%;
    text-align: center;
  }
  h1, h2 {
    width: 100%;
    color: $white;
    margin: 0;
  }
}
```

O resultado é visível na Figura 8.2.



FIGURA 8.2 – Cabeçalho da página principal do *site* MyInstagram

8.3.2 FEED DO INSTAGRAM

Comece por adicionar o seguinte bloco de código no ficheiro **index.html**, mais concretamente dentro do elemento `<section class="instagram-wrap">`:

```
<div class="container">
  <div class="row">
    <div class="col-xs-12">
      <div class="instagram-content">
        <h3>Ultimas fotos</h3>
        <div class="row photos-wrap">
          <div id="instafeed"><!-- Instafeed div --></div>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
```

Neste código usa-se a *grid* do Bootstrap e define-se um elemento `<div>` com o atributo `id` que contém o valor `instafeed` e que irá receber o *feed* do Instagram. Em seguida, adicione regras CSS para estilizar a cor e as fontes dos elementos anteriores:

```
.instagram-wrap {
  background: $lightgrey;
}
.instagram-content {
  h3 {
    text-transform: uppercase;
    letter-spacing: 0.5em;
    font-size: 12px;
    font-weight: 700;
    text-align: center;
    padding: $padding*5 0;
    color: darken($lightgrey, 20%);
  }
}
```

O resultado de uma hipotética imagem do Instagram é apresentado na Figura 8.3.



FIGURA 8.3 – Imagem da página principal do *site* MyInstagram



Outras regras são usadas para melhorar a estilização destes componentes; contudo, por motivos de legibilidade e de forma a tornar mais compacta a explicação, não serão aqui incluídas. Os ficheiros completos de todos os projetos podem ser descarregados a partir do *site* da FCA.

O próximo passo é importar as fotos do Instagram, com ajuda do **Instafeed.js**. Este *plugin* permite que qualquer pessoa consiga, de forma muito simples, apresentar um feed do Instagram numa página Web. Para utilizar o **Instafeed.js**, é preciso um *token* de acesso fornecido pelo Instagram. Para adquiri-lo tem de se registar no Instagram e visitar a secção de programadores².

² Link: <https://www.instagram.com/developer/clients/manage>.