

Início Fácil

Existem várias formas de incluir o Bootstrap num projeto. A forma mais fácil de importar o Bootstrap envolve a utilização de uma *Content Delivery Network* (CDN – Rede de Distribuição de Conteúdo). Na documentação oficial é possível encontrar uma excelente introdução ao processo de importação com a utilização de uma BootstrapCDN, [aqui](#).

Também é possível instalar o Bootstrap no projeto como dependência (*dependency*) utilizando **npm**, caso utilize uma biblioteca JavaScript como a React ou uma framework como a Angular.

```
npm install --save bootstrap
```

Depois de instalar o Bootstrap, pode proceder com a importação do ficheiro CSS.

- Para React, inclua o Bootstrap no ficheiro de entrada da sua aplicação (*entry file*); normalmente **index.js**.
- Para Angular 2+, inclua o Bootstrap no ficheiro de configuração; provavelmente o seu ficheiro **angular.json**.

Inclua também o jQuery, Popper e Bootstrap *minified bundle*, uma vez que são necessários para garantir o funcionamento correto de determinados componentes.

Exemplo React

Importe o Bootstrap para o seu ficheiro **index.js**.

```
import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import * as serviceWorker from './serviceWorker';

import 'bootstrap/dist/css/bootstrap.min.css';

import $ from 'jquery';

import Popper from 'popper.js';

import 'bootstrap/dist/js/bootstrap.min.js';
```

Exemplo Angular 2+

Importe o Bootstrap para o seu ficheiro **angular.json**.

```
"build": {
  "builder": "@angular-devkit/build-angular:browser",
  "options": {
    "outputPath": "dist/output",
    "index": "src/index.html",
    "main": "src/main.ts",
    "polyfills": "src/polyfills.ts",
    "tsConfig": "src/tsconfig.app.json",
    "assets": [
      "src/favicon.ico",
      "src/assets"
    ],
    "styles": [
      "node_modules/bootstrap/dist/css/bootstrap.min.css"
    ],
    "scripts": [
      "node_modules/jquery/dist/jquery.min.js",
      "node_modules/popper.js/dist/umd/popper.min.js",
      "node_modules/bootstrap/dist/js/bootstrap.min.js"
    ]
  }
}
```

Alternativamente, é possível descarregar o código fonte do Bootstrap [aqui](#) e utilizá-lo localmente.

Capacidade de Resposta

A funcionalidade responsiva pela qual o Bootstrap é tão famoso, é o **12-Column Grid System** mencionado anteriormente. A grelha divide-se em 3 conjuntos importantes de classes: O **container**, **row (linha)** e **column (coluna)**.

- **Container** é a “embalagem” (*wrapper*) para as linhas (rows) e colunas (columns).
- **Row (linha)** hospeda as colunas.
- **Columns (colunas)** contêm o conteúdo principal, obrigatoriamente dentro de linhas.

Para construir o *container*, é possível utilizar a classe **container** ou **container-fluid**. A principal diferença entre estes dois *containers* **responsivos** é que a classe **container** especifica uma caixa de largura fixa para o ecrã de cada dispositivo ou dimensão do *viewport*, enquanto a classe **container-fluid** especifica um *container* que preenche sempre a largura disponível.

Mantendo esta ideia em mente, podemos começar a criar o nosso *container*. Neste exemplo, temos um *container* de largura fixa com 2 linhas (rows) no seu interior. As linhas serão os nossos *wrappers* para as colunas. Pode criar qualquer número de linhas necessário com diferentes tamanhos de coluna.

```
<div class="container">
  <div class="row">. . .</div>
  <div class="row">. . .</div>
</div>
```

Após criar uma linha, pode utilizar o **sistema de 12 colunas (12-Column Grid System)**, providenciado pelo Bootstrap, para colocar colunas flexíveis de diferentes dimensões no interior das linhas criadas. As colunas irão hospedar o conteúdo principal, obrigatoriamente no interior de linhas. A nomenclatura para criar colunas é a seguinte:

col-<breakpoint>-<size-of-column>

O *Breakpoint* aplica um tamanho para a coluna, com base no ecrã do dispositivo ou tamanho do *viewport*, aplicado na seguinte ordem:

xs > sm > md > lg > xl

- **xs** – para dispositivos extra pequenos, como telemóveis em modo de retrato (< 576px); (**Atualmente não suportado pelo Bootstrap 4; substituído por col-<size-of-column>**)
- **sm** – para dispositivos pequenos, como telemóveis em modo de paisagem (>=576px);
- **md** – para dispositivos médios, como tablets (>=768px);

- **lg** – para dispositivos grandes, como portáteis/laptops ($\geq 992\text{px}$);
- **xl** – para dispositivos extra grandes, como desktops ($\geq 1200\text{px}$);

É importante salientar que o Bootstrap é uma coleção de ficheiros JavaScript e CSS, portanto, caso necessite de adicionar um *breakpoint* a qualquer momento, ou modificar os *breakpoints* existentes, basta sobrepor os novos componentes aos ficheiros originais. Exibiremos um exemplo deste processo no final deste tópico.

Como uma fila pode conter até 12 colunas, a dimensão das colunas varia entre **1 e 12**. O tamanho das colunas é representado em percentagens e encontra-se sempre relacionado com o “*elemento-mãe*” (*parent*), sendo este a própria linha. Seguindo o exemplo providenciado anteriormente, podemos começar a utilizar algumas colunas nas nossas linhas.

```

<div class="container">

  <div class="row">

    <div class="col-sm-4" style="background: lightgreen">. . </div>

    <div class="col-sm-8" style="background: aqua">. . </div>

  </div>

  <div class="row">

    <div class="col-lg-4" style="background: lime">. . </div>

    <div class="col-lg-4" style="background: lightgray">. . </div>

    <div class="col-lg-4" style="background: lightblue">. . </div>

  </div>

</div>

```

Este processo irá criar os seguintes resultados:



Neste exemplo, a primeira linha tem 2 colunas com uma proporção de 1:2. A primeira coluna terá uma largura de 33% e a segunda coluna terá uma largura de 66%, uma vez que a largura das colunas é definida em percentagens e em relação à largura da linha.

Estas colunas irão aplicar o estilo para dispositivos pequenos (**col-sm-***) e manter as suas proporções para dispositivos de tamanho intermédio (**col-md-***) até dispositivos extra grandes (**col-xl-***). Isto acontece porque o *breakpoint sm* é baseado num *media query* que aplica o estilo a dispositivos com um ecrã de dimensão igual ou superior a 576px se, no entanto, a classe **col-md-*** não estiver presente.

A segunda linha possui 3 colunas com o mesmo tamanho, onde cada coluna ocupa 1/3 da largura do *container*. Estas colunas irão manter a sua proporção desde dispositivos grandes (**col-lg-***) a dispositivos extra grandes (**col-xl-***). Como não especificamos uma classe para dispositivos pequenos (**col-sm-***) e dispositivos médios (**col-md-***), a segunda linha terá 3 colunas separadas em novas linhas. Isto é ideal para dispositivos com ecrãs mais pequenos, uma vez que a experiência do utilizador será mais agradável. A segunda linha também pode ser criada da seguinte forma:

```
<div class="row">
  <div class="col-lg" style="background: lime">. . .</div>
  <div class="col-lg" style="background: lightgray">. . .</div>
  <div class="col-lg" style="background: lightblue">. . .</div>
</div>
```

Caso não especifiquemos o tamanho da coluna, o Bootstrap assumirá uma linha com colunas de largura igual. Note que a soma das colunas deverá ser igual a 12 se pretender manter as colunas numa única linha e lado a lado. Na primeira linha, existem duas colunas com 4 + 8 e na segunda linha temos a soma 4 + 4 + 4, sendo que ambos os resultados são 12.

É possível, obviamente, criar linhas mais flexíveis, que alteram a orientação da coluna de acordo com o ecrã do dispositivo. O seguinte exemplo contém 5 componentes retirados de um projeto integral desenvolvido na empresa. Estes componentes são alinhados horizontalmente para resoluções superiores e decompõem-se em novas linhas de acordo.



Sem o Bootstrap, seria necessário escrever algo semelhante a este exemplo para atingir o resultado mencionado anteriormente.

```
<div class="main-content">

  <div class="item">. . .</div>

  <div class="item">. . .</div>

  <div class="item">. . .</div>

  <div class="item">. . .</div>

  <div class="item">. . .</div>

</div>
```

```
.main-content {
  position: relative;
  display: flex;
  max-width: 100%;
  padding: 15px;

  .item {
    position: relative;
    display: flex;
    flex-direction: column;
    margin: 15px;
    min-width: 150px;
    max-width: 300px;
    width: 100%;
    height: 400px;
  }
}
```

```

@media only screen and (min-width: 768px) {
  .main-content {
    flex-flow: row wrap;
    justify-content: center;
  }
}

@media only screen and (min-width: 1200px) {
  .main-content {
    flex-wrap: nowrap;
    justify-content: space-evenly;
  }
}

```

Com o Bootstrap, seria necessário escrever apenas o seguinte.

```

<div class="container-fluid">
  <div class="row">
    <div class="col-sm-12 col-md-6 col-lg"> . . </div>
    <div class="col-sm-12 col-md-6 col-lg"> . . </div>
    <div class="col-sm-12 col-md-6 col-lg"> . . </div>
    <div class="col-sm-12 col-md-6 col-lg"> . . </div>
    <div class="col-sm-12 col-md-6 col-lg"> . . </div>
  </div>
</div>

```

Podemos dividir este último exemplo em pequenas partes:

- Definimos um *container* com fluidez de largura, com uma linha;
- A única linha possui 5 colunas que alteram as suas dimensões e orientação de acordo com o ecrã do dispositivo;
- Para dispositivos pequenos (**col-sm-***), cada coluna será separada numa nova linha, uma vez que o tamanho para a coluna deste *breakpoint* é 12, correspondente à totalidade da largura da linha;
- Para dispositivos médios (**col-md-***), é possível reparar que a soma dos tamanhos para este *breakpoint* é igual a 30. Isto implica uma separação da linha em três linhas distintas, cada uma contendo até duas colunas de largura igual.
- Para dispositivos grandes (**col-lg-***) a extra grandes (**col-xl-***), o tamanho da coluna é omitido, significando que o Bootstrap irá criar uma linha com colunas de largura igual.

Personalizar e adicionar novos *breakpoints*

Caso pretenda adicionar um novo *breakpoint* ou modificar os *breakpoints* existentes, é necessário importar ficheiros Bootstrap vitais para o ficheiro CSS primário e substituir as configurações predefinidas. No exemplo abaixo, alterámos o *breakpoint xl* predefinido de 1200px para 1024px e adicionámos um novo *breakpoint xxl*, iniciado com 1440px.

Nota: Neste exemplo, o Bootstrap foi instalado como dependência local.

```
@import "../node_modules/bootstrap/scss/functions";

@import "../node_modules/bootstrap/scss/variables";

$grid-breakpoints: (
  xl: 1024px,
  xxl: 1440px
);

$container-max-widths: (
  xl: 1024px,
  xxl: 1440px
);

@import "../node_modules/bootstrap/scss/bootstrap.scss";
```


Resumo

Com esta breve introdução ao Bootstrap, abordámos algumas das funcionalidades mais úteis providenciadas por esta framework. Resumindo, estas são algumas das principais vantagens em utilizar esta framework e as razões pelas quais é tão popular:

- É extremamente **fácil de utilizar**, pois não requer um conhecimento aprofundado em CSS ou JavaScript. Com uma simples pesquisa na documentação oficial, é possível construir o website com o design pretendido e focar-se apenas nas suas funcionalidades;
- **Poupa imenso tempo e trabalho** no processo de escrita do código CSS, graças às suas classes pré-concebidas com funcionalidades responsivas;
- **Providencia uma vasta gama de componentes pré-concebidos**, todos com um enorme potencial de personalização e fácil implementação;
- **Suportado em todos os browsers mais populares**, tais como o Chrome, Firefox, Edge, Safari, etc.;
- **Open Source** e completamente gratuito.