



## Sobre Este Curso

### Público Alvo

Programadores que têm como objetivo atuar como Desenvolvedor Front-End e criar interfaces para produtos digitais.

### Pré-Requisitos

Conhecimentos de Lógica de Programação, Html, CSS e Javascript.

Desenvolvido por Ralf Lima e licenciado com exclusividade para  
Apex Ensino





## Índice

|                                                        |            |
|--------------------------------------------------------|------------|
| <b>SOBRE ESTE CURSO .....</b>                          | <b>I</b>   |
| PÚBLICO ALVO .....                                     | I          |
| PRÉ-REQUISITOS .....                                   | I          |
| <b>ÍNDICE.....</b>                                     | <b>I</b>   |
| <b>COPYRIGHT .....</b>                                 | <b>III</b> |
| EQUIPE.....                                            | III        |
| HISTÓRICO DAS EDIÇÕES .....                            | III        |
| <b>CAPÍTULO 01 – O QUE É DESIGN RESPONSIVO .....</b>   | <b>4</b>   |
| O QUE É DESIGN RESPONSIVO.....                         | 5          |
| SURGIMENTO DO DESIGN RESPONSIVO .....                  | 6          |
| <b>CAPÍTULO 02 – MOBILE FIRST .....</b>                | <b>8</b>   |
| MOBILE FIRST.....                                      | 9          |
| <b>CAPÍTULO 03 – UNIDADES DE MEDIDA .....</b>          | <b>10</b>  |
| UNIDADES DE MEDIDA .....                               | 11         |
| UNIDADE EM .....                                       | 11         |
| UNIDADE REM.....                                       | 12         |
| <b>CAPÍTULO 04 – VIEWPORT .....</b>                    | <b>14</b>  |
| VIEWPORT .....                                         | 15         |
| <b>CAPÍTULO 05 – GRID VIEW .....</b>                   | <b>16</b>  |
| O QUE É GRID VIEW.....                                 | 17         |
| <b>CAPÍTULO 06 – MEDIA QUERIES.....</b>                | <b>22</b>  |
| MEDIA QUERIES.....                                     | 23         |
| <b>CAPÍTULO 07 – IMAGENS E VÍDEOS RESPONSIVOS.....</b> | <b>25</b>  |
| IMAGENS E VÍDEOS RESPONSIVOS.....                      | 26         |
| <b>CAPÍTULO 08 - BOOTSTRAP.....</b>                    | <b>28</b>  |
| BOOTSTRAP.....                                         | 29         |
| <b>CAPÍTULO 09 – DICAS PARA DESENVOLVEDORES.....</b>   | <b>35</b>  |
| DICAS PARA DESENVOLVEDORES.....                        | 36         |

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |



## Copyright

As informações contidas neste material se referem ao curso de **Design Responsivo** e estão sujeitas as alterações sem comunicação prévia, não representando um compromisso por parte do autor em atualização automática de futuras versões.

A **Apex** não será responsável por quaisquer erros ou por danos acidentais ou consequenciais relacionados com o fornecimento, desempenho, ou uso desta apostila ou os exemplos contidos aqui.

Os exemplos de empresas, organizações, produtos, nomes de domínio, endereços de e-mail, logotipos, pessoas, lugares e eventos aqui representados são fictícios. Nenhuma associação a empresas, organizações, produtos, nomes de domínio, endereços de e-mail, logotipos, pessoas, lugares ou eventos reais é intencional ou deve ser inferida.

A reprodução, adaptação, ou tradução deste manual mesmo que parcial, para qualquer finalidade é proibida sem autorização prévia por escrito da **Apex**, exceto as permitidas sob as leis de direito autoral.

## Equipe

### Conteúdos

- Ralf Lima

### Diagramação

- Fernanda Pereira

### Revisão

- Fernanda Pereira

## Histórico das Edições

| Edição | Idioma    | Edição        |
|--------|-----------|---------------|
| 1ª     | Português | Julho de 2019 |

© Copyright 2019 **Apex**. Desenvolvido por Ralf Lima e licenciado com exclusividade para Apex Ensino.

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Capítulo 01 – O que é Design Responsivo



### Objetivos:

Neste capítulo você irá aprender:

- O que é Design Responsivo
- Surgimento o Design Responsivo

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |



## O que é Design Responsivo

Design responsivo é uma técnica para que as páginas web consigam se adaptar independente da resolução de tela e do dispositivo móvel. Tem como base utilizar o CSS e o JavaScript.

As principais técnicas para trabalhar com o design responsivo são compostas por:

- Media Queries
- Grid Layout
- Flexbox

Além das técnicas citadas anteriormente há frameworks, que são ferramentas que auxiliam na elaboração de páginas web responsivas, que garantem padronização nos projetos, velocidade no desenvolvimento e ampla comunidade de desenvolvedores para auxiliar nas eventuais dúvidas, os principais frameworks disponíveis no mercado são:

- Bootstrap
- Foundation
- Skeleton
- HTML KickStart
- Montage HTML5

Neste treinamento da Apex Ensino vamos ensinar a utilizar o Bootstrap, que atualmente é o framework responsivo mais utilizado no mundo, fornecendo ao estudante a realidade atual das empresas de desenvolvimento web.

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Surgimento do Design Responsivo

O termo design responsivo ou layout responsivo foi utilizado pela primeira vez em maio de 2010 em um blog chamado A List Apart, pelo escritor Ethan Marcotte.

Na época Ethan havia escrito que com a revolução digital dos tablets e smartphones seria inevitável a criação de conteúdos que pudessem oferecer boa ergonomia para a leitura e visualização de demais elementos disponibilizados nas páginas.

Uma introdução que Ethan faz em seu artigo é a seguinte: O controle conhecido pelos designers no meio de impressão e, muitas vezes, desejado no meio virtual é simplesmente uma função da limitação da página impressa. Nós devemos aceitar o fato de que a web não tem as mesmas restrições e design para essa flexibilidade. Mas primeiro nós temos que aceitar o fluxo e refluxo das coisas.

A internet possui diversos sites com conteúdos para segmentos específicos, o usuário não deve se adaptar ao conteúdo, mas sim o conteúdo ao usuário, por isso é importante desenvolver um site que consiga oferecer o máximo de conforto visual.

Mecanismos de busca como o Google levam em consideração se há versões responsivas dos sites para deixar na primeira página de buscas, sendo assim não é apenas uma questão de estética e ergonomia ao usuário.

O fluxo de pesquisas através de smartphones e tablets é algo cada vez mais frequente, sendo assim nada mais justo que disponibilizar conteúdos adequados para os usuários.

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |





O site [developers.google](https://developers.google.com) aponta alguns motivos para o uso de sites responsivos:

- Facilita o compartilhamento e a vinculação do conteúdo com um único URL para os usuários;
- Ajuda os algoritmos do Google a atribuir de maneira precisa propriedades de indexação à página, em vez de sinalizar a existência de páginas correspondentes para computadores ou dispositivos móveis;
- Exige menos tempo de engenharia para manter diversas páginas com o mesmo conteúdo;
- Reduz a possibilidade de erros comuns que afetam os sites para dispositivos móveis.

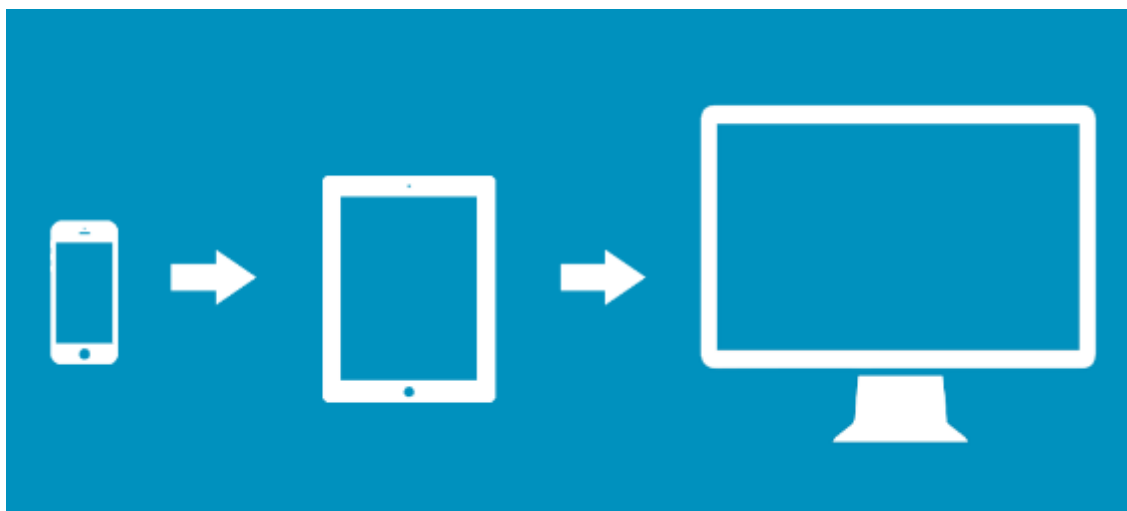
**Para mais informações sobre o design responsivo e suas vantagens para o Google acesse:**

<https://developers.google.com/search/mobile-sites/mobile-seo/responsive-design>

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Capítulo 02 – Mobile First



### Objetivos:

Neste capítulo você irá aprender:

- Técnica Mobile First

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |

## Mobile First

Mobile First é uma técnica utilizada para ter foco no site em dispositivos móveis, isso não significa deixar os modelos para desktops de lado.

Segundo o site Agência Brasil em 2017, em todo o território nacional, 49% dos lares utilizavam a internet através de dispositivos móveis, esse dado é importante, pois podemos ter uma visão mais cuidadosa na elaboração dos sites e sistemas web voltados para determinado dispositivo.

Na imagem abaixo você vai conseguir compreender melhor a atenção que os desenvolvedores precisam ter quando utilizada essa técnica.



Note que na figura da direita os links, textos e imagens estão visivelmente melhores, facilitando a navegabilidade, já a imagem a imagem do dispositivo móvel a esquerda os conteúdos estão seguindo uma característica idêntica ao do computador (desktop).

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Capítulo 03 – Unidades de Medida



### Objetivos:

Neste capítulo você irá aprender:

- Unidades de Medidas EM
- Unidades de Medida REM

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |



## Unidades de Medida

Provavelmente você já utilizou alguma medida no CSS como o px ou percentual em seus projetos correto? Quando utilizamos essas unidades pode haver problemas em determinadas resoluções como quebra indevida de elementos, pixelização em imagens ou fontes de tamanhos desproporcionais, sendo assim é interessante haver uma unidade que possa ajudar nessa padronização.

### Unidade EM

Uma alternativa é utilizar a unidade de medida EM, que é uma unidade tipográfica, onde seu tamanho equivale a letra **M**, para converter uma fonte por exemplo de px para em basta fazermos um cálculo simples que é **target / context**.

Vamos entender de maneira simples, imagine que você tenha essa estrutura:

```
1  h1{
2    |    font-size: 20px;
3    |
4    |
5  p{
6    |    font-size: 12px;
7    |
8    }
```

O corpo da página está como padrão 16px, os demais elementos possuem tamanhos de fontes específicas, mas lembre-se que o px pode haver uma certa dificuldade em ter padronização em todos os dispositivos, sendo assim vamos alterar para EM, assim teremos uma maior confiabilidade no tamanho dessas fontes.

Vamos utilizar o cálculo **target / context**, onde o **target** é o corpo da página e o **context** são as tags h1 e p, lembrando que o padrão de tamanho de fontes como o elemento <p> é de 16px.

Iremos adicionar o padrão no body de 100%, que equivale aos 16px, já os elementos que estarão envolvidos pela tag body deverão ser divididas por 16, sendo assim faremos os cálculos:

- **h1:** 20px / 16px
- **p:** 12px / 16px

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



Fazendo os cálculos teremos nossa folha de estilos com a seguinte estrutura utilizando a unidade de medida EM:

```
1  h1{
2    |    font-size: 1.25em;
3  }
4
5  p{
6    |    font-size: 0.75em;
7  }
```

## Unidade REM

Além do EM existe o padrão REM, onde o R vem da palavra **root** que significa padrão, imagine que você queira usar como padrão o body da página. Se você deixar o body em 100% e usar um h1 com 1.25rem, você está dizendo que quando a página está utilizando a fonte 16px que equivale a 100%, 20px equivalem a 1.25 rem.

Confuso? O Rem tem como base usar o body como base, sendo assim se diminuirmos os 100% do body para 80%, quer dizer que nosso h1 será diminuído, isso também funciona se você quiser aumentar o body de 100% para 120%.

Em outras palavras, as tags agora estarão utilizando o body como referência, vamos ao exemplo:





```
1  body{
2    |    font-size: 100%;
3  }
4
5  h1{
6    |    font-size: 1.25rem;
7  }
8
9  p{
10   |    font-size: 0.75rem;
11 }
```

Agora que sabemos como funciona o EM e o REM podemos deixar nossas páginas bem mais ergonômicas para nosso usuário, sabendo utilizar melhor o espaço disponibilizado pelas telas.

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |

Para aqueles que têm dúvidas em qual utilizar fica a dica que o REM é mais flexível, porém ele não funciona em alguns navegadores como o Internet Explorer 8 ou inferior.

Atualmente, com a velocidade da atualização dos browsers, o uso do REM é cada vez mais frequente, utilizar o REM ou o EM vai depender de alguma característica marcante do navegador utilizado pelos usuários, abaixo segue uma tabela contendo as unidades de medida e os navegadores que possuem suporte.

| Length Unit                       |  |  |  |  |
|-----------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| em, ex, %, px, cm, mm, in, pt, pc | 1.0                                                                               | 3.0                                                                               | 1.0                                                                                 | 1.0                                                                                 |
| ch                                | 27.0                                                                              | 9.0                                                                               | 1.0                                                                                 | 7.0                                                                                 |
| rem                               | 4.0                                                                               | 9.0                                                                               | 3.6                                                                                 | 4.1                                                                                 |
| vh, vw                            | 20.0                                                                              | 9.0                                                                               | 19.0                                                                                | 6.0                                                                                 |
| vmin                              | 20.0                                                                              | 12.0                                                                              | 19.0                                                                                | 6.0                                                                                 |
| vmax                              | 26.0                                                                              | 16.0                                                                              | 19.0                                                                                | 7.0                                                                                 |

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Capítulo 04 – Viewport



### Objetivos:

Neste capítulo você irá aprender:

- O que é Viewport

▪

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |





## Viewport

No capítulo anterior vimos o uso das medidas EM e REM, porém uma coisa muito importante que devemos levar em consideração é a área do dispositivo. Quanto vale 100% do tamanho de tela?

Para termos essa informação é necessário utilizar um comando chamado viewport, que ficará no cabeçalho do nosso HTML, ele nos auxiliará com as medidas, veja abaixo sua estrutura:

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <title>Design Responsivo</title>
8  </head>
9  <body>
10
11 </body>
12 </html>

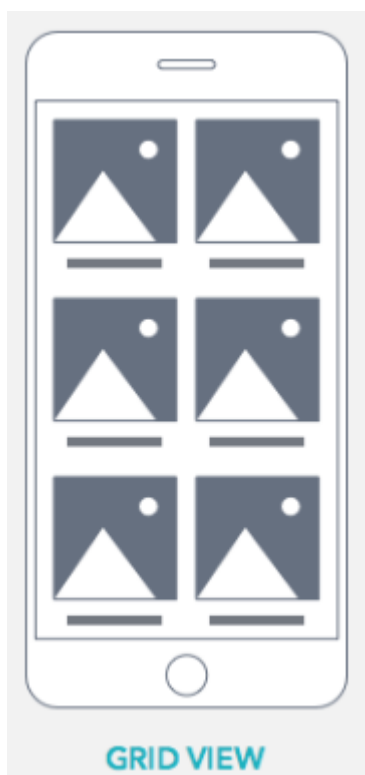
```

Na imagem acima veja que a linha 4 possui uma tag meta chamada **viewport**, é através dela que conseguiremos ter uma noção do tamanho disponibilizado para nossas páginas, importante que ele possui um **content**, que coloca a escala inicial em 1, isso serve para remover qualquer tipo de zoom, sendo assim o usuário que acessar a página estará com a tela em 100%.

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Capítulo 05 – Grid View



### Objetivos:

Neste capítulo você irá aprender:

- O que é Grid View

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |



## O que é Grid View

A técnica Grid View é muito utilizada em sites responsivos, sua finalidade é prover blocos para encaixar os elementos das páginas HTML como textos, imagens, vídeos, formulários, entre outros.

Frameworks como o Bootstrap utilizam o Grid View para facilitar o desenvolvimento das páginas web, além de ser um padrão que pode ser desenvolvido do zero sem qualquer framework.

Um Grid é composto por 12 colunas, sendo assim imagine uma planilha do Excel contendo 12 colunas, porém como em um Excel você poderá mesclar essas colunas, para exemplificar o uso do Grid View vamos montar uma estrutura básica em HTML:

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <title>Trabalhando com Grid View</title>
8  </head>
9  <body>
10
11      <header>
12          <h1>Topo do site</h1>
13      </header>
14
15      <nav>
16          <a href="#">Link 01</a>
17          <a href="#">Link 02</a>
18          <a href="#">Link 03</a>
19      </nav>
20
21      <main>
22          <h1>Conteúdo principal</h1>
23      </main>
24
25  </body>
26  </html>

```

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



Essa estrutura HTML possui um topo, menu e conteúdo principal, agora vamos criar uma folha de estilos ou adicionar o comando `<style></style>` entre as tags `<head></head>` com o seguinte código:

```
1  /*Margin e Padding serão englobados em comandos de WIDTH e HEIGHT*/
2  * {
3  |   box-sizing: border-box;
4  }
5
6  /*Topo*/
7  header {
8  |   border: 1px solid red;
9  |   padding: 15px;
10 }
11
12 /*Menu*/
13 nav {
14 |   width: 25%;
15 |   float: left;
16 |   padding: 15px;
17 |   border: 1px solid red;
18 }
19
20 /*Links*/
21 a{
22 |   display: block;
23 |   padding-bottom: 8.8px;
24 }
25
26 /*Conteúdo principal*/
27 main {
28 |   width: 75%;
29 |   float: left;
30 |   padding: 15px;
31 |   border: 1px solid red;
32 }
```



Na imagem acima você tem acesso ao nosso CSS, veja que na linha 3 há um comando chamado box-sizing, ele é muito importante para nosso desenvolvimento responsivo.

Vamos imaginar que você queira criar uma div com 400px de largura, porém necessita adicionar uma borda de 1px e 15px de padding. Sua estrutura terá o total de 432px de largura, mas lembre-se que você quer que o tamanho seja de 400px, utilizando o box-sizing ele fará com que essas características que aumentarão o tamanho de sua div fiquem dentro dos 400px exigidos.

Sempre que você adicionar uma borda ou margem será contabilizado como uma largura ou altura extra, agora com o box-sizing esse problema não ocorrerá, isso vai fazer com que suas páginas tenham maior flexibilidade e com as medidas mais precisas.

Nosso exemplo com HTML e CSS ficará da seguinte maneira:

|                                                                               |                           |
|-------------------------------------------------------------------------------|---------------------------|
| <b>Topo do site</b>                                                           |                           |
| <a href="#">Link 01</a><br><a href="#">Link 02</a><br><a href="#">Link 03</a> | <b>Conteúdo principal</b> |

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



Ainda não terminamos, pois utilizamos apenas percentuais básicos, agora que o Grid View entra em ação, vamos adicionar o seguinte código em nosso CSS:

```
35  /*Colunas*/
36  .col-1 {width: 8.33%;}
37  .col-2 {width: 16.66%;}
38  .col-3 {width: 25%;}
39  .col-4 {width: 33.33%;}
40  .col-5 {width: 41.66%;}
41  .col-6 {width: 50%;}
42  .col-7 {width: 58.33%;}
43  .col-8 {width: 66.66%;}
44  .col-9 {width: 75%;}
45  .col-10 {width: 83.33%;}
46  .col-11 {width: 91.66%;}
47  .col-12 {width: 100%;}
48
49  /*Chamando as colunas via CLASS*/
50  [class*="col-"] {
51      float: left;
52      padding: 15px;
53      border: 1px solid red;
54  }
```

Na imagem acima temos agora um Grid View, onde você especifica qual será o percentual que determinada coluna possui, dependendo da quantidade estipulada via HTML, sempre que chamada alguma class iniciando com **col-** será colocada alinhada a esquerda com um padding e uma borda, vamos refazer nossa estrutura HTML utilizando esses novos elementos:

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <title>Trabalhando com Grid View</title>
8  </head>
9  <body>
10
11      <div class="col-12">
12          <h1>Topo do site</h1>
13      </div>
14
15      <div class="col-3">
16          <a href="#">Link 01</a>
17          <a href="#">Link 02</a>
18          <a href="#">Link 03</a>
19      </div>
20
21      <div class="col-9">
22          <h1>Conteúdo principal</h1>
23      </div>
24
25  </body>
26  </html>

```

Removendo as tags **<header>**, **<nav>** e **<main>**, adicionamos as colunas criadas anteriormente, o resultado foi exatamente o mesmo, porém agora você tem a chance de mesclar as colunas, deixando sua página mais flexível.

O resultado pode parecer o mesmo, mas verá que futuramente com o uso das Media Queries, nossas páginas irão ficar ainda mais fáceis de serem utilizadas, além de garantir uma boa ergonomia aos usuários e um código mais limpo para os desenvolvedores.

Vale a dica de que as duas estruturas utilizadas neste exemplo estão corretas para desenvolver as aplicações responsivas, mas a grande ideia é demonstrar que há inúmeras maneiras de você implementar seus sites utilizando o Design Responsivo.

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Capítulo 06 – Media Queries

# MEDIA QUERIES



### Objetivos:

Neste capítulo você irá aprender:

- Uso de Media Queries



## Media Queries

O uso de Media Queries em sites responsivos é indispensável, se você já trabalhou com alguma linguagem de programação deve conhecer a estrutura condicional IF ou SWITCH, caso não conheça não se assuste, veremos como funciona agora.

A ideia é criar uma estrutura que vale a resolução de tela, trabalhamos com um tamanho mínimo ou máximo de largura para a tela do dispositivo, vamos ao exemplo via CSS:

```

1  @media screen and (max-width:500px){
2      body{
3          background-color: ■red;
4      }
5  }
6
7  @media screen and (min-width:501px) and (max-width:900px){
8      body{
9          background-color: ■green;
10     }
11 }
12
13 @media screen and (min-width:901px) and (max-width:1200px){
14     body{
15         background-color: ■blue;
16     }
17 }
18
19 @media screen and (min-width:1201px){
20     body{
21         background-color: ■yellow;
22     }
23 }
```

Veja que há uma estrutura para algumas resoluções, a primeira por exemplo deixará o fundo da página em vermelho caso a tela seja até 500px, entre 501px e 900px ficará verde, entre 901px e 1200px azul e acima de 1200px amarelo.

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



Dentro de cada @media você adiciona as tags HTML que deseja que tenham alterações de acordo com a resolução.

Talvez você ache que um CSS com vários @media seja algo que vai dar um trabalho muito grande de manutenção, pois há várias resoluções em um só arquivo CSS, realmente caso você esteja trabalhando com uma loja virtual, portal ou algo parecido realmente será um trabalho bem grande deixar organizado, mas você pode separar em arquivos CSS e especificar a Media Queries na chamada do arquivo.

Vamos exemplificar isso na imagem abaixo:

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7   <!-- Media Queries -->
8   <link rel="stylesheet" href="estilo1.css" media="screen and (max-width: 500px)">
9   <link rel="stylesheet" href="estilo2.css" media="screen and (min-width: 501px) and (max-width: 900px)">
10  <link rel="stylesheet" href="estilo3.css" media="screen and (min-width: 901px) and (max-width: 1200px)">
11  <link rel="stylesheet" href="estilo4.css" media="screen and (min-width: 1201px)">
12
13  <title>Trabalhando com Media Queries</title>
14 </head>
15 <body>
```

Note que entre as linhas 8 e 11 há uma referência para estilos CSS diferentes, após adicionar o local do arquivo há o comando media, onde é utilizado screen e as larguras, sendo assim você pode criar arquivos específicos para cada resolução e chamar entre as tags <head></head> da sua página.

Sobre o suporte de Media Queries, os principais browsers possuem suporte há algum tempo, sendo assim certifique-se os navegadores que os usuários mais acessam suas páginas para garantir a melhor experiência possível, abaixo segue uma imagem contendo os principais navegadores e as versões mínimas que compreendem os comandos de Media Queries:

| Property |  |  |  |  |
|----------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| @media   | 21                                                                                  | 9                                                                                   | 3.5                                                                                 | 4.0                                                                                   |

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |

## Capítulo 07 – Imagens e Vídeos Responsivos



### Objetivos:

Neste capítulo você irá aprender:

- Como trabalhar com imagens e vídeos responsivos

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Imagens e Vídeos Responsivos

Trabalhar com imagens ou vídeos podem ser considerados por muitos um problema devido a chance de pixelização, se você não conhece esse termo iremos explicar de maneira bem simples. Imagine que você possua uma imagem no tamanho de 150x150px, utilizando o CSS iremos alterar a largura e altura para 800px, o que acontecerá? Simples, a imagem ficará distorcida, esse processo pode ocorrer quando a imagem é muito pequena ou até mesmo se for muito grande e tentar diminuir.

Para trabalhar com o desenvolvimento responsivo há algumas técnicas que podemos utilizar para amenizar alguns problemas, então vamos iniciar com o modelo mais básico, que é trabalhar com percentual.

Imagine que você esteja trabalhando em um site para smartphones e tablets com resolução de largura até 780px, neste caso se houver uma imagem de 500px e esticarmos um pouco, possivelmente não iremos notar muito os problemas de pixelização, sendo assim basta colocar a imagem em percentual:

```
1  img {  
2      width: 100%;  
3      height: auto;  
4  }
```

Note na figura acima que a imagem terá 100% da largura do elemento que a comporta, claro que nem sempre será útil, mas pensando em resoluções para smartphones ou tablets, essa é uma técnica muito utilizada.

Outra alternativa interessante e muito útil é utilizar uma media queries com o comando picture, veja abaixo sua estrutura:

```
16  <body>  
17      <picture>  
18          <source srcset="imagem_pequena.jpg" media="(max-width: 400px)">  
19          <source srcset="imagem_grande.jpg">  
20            
21      </picture>  
22  </body>
```

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |



Na imagem você pode observar que há um comando chamado `<source srcset="">`, esse comando junto com o **media** fará com que a imagem que está sendo chamada pelo `<img src="">` seja escolhida.

Na linha 18 é adicionada uma imagem chamada **imagem\_pequena.jpg**, sua media está referenciando a largura máxima de 400px, já na linha 19 há uma referência para a imagem chamada **imagem\_grande.jpg**, caso a linha 18 seja verdadeira, neste caso a largura de tela seja até 400px será exibida a imagem pequena, caso contrário a linha 19 será exibida.

Então é uma condicional bem simples. Se a largura da tela for de 400px ou inferior será exibida a imagem de tamanho pequeno, caso contrário será exibida a imagem de tamanho grande.

Lembrando que para funcionar o `<source-srcset="">`, é necessário estar englobado entre as tags `<picture></picture>`.

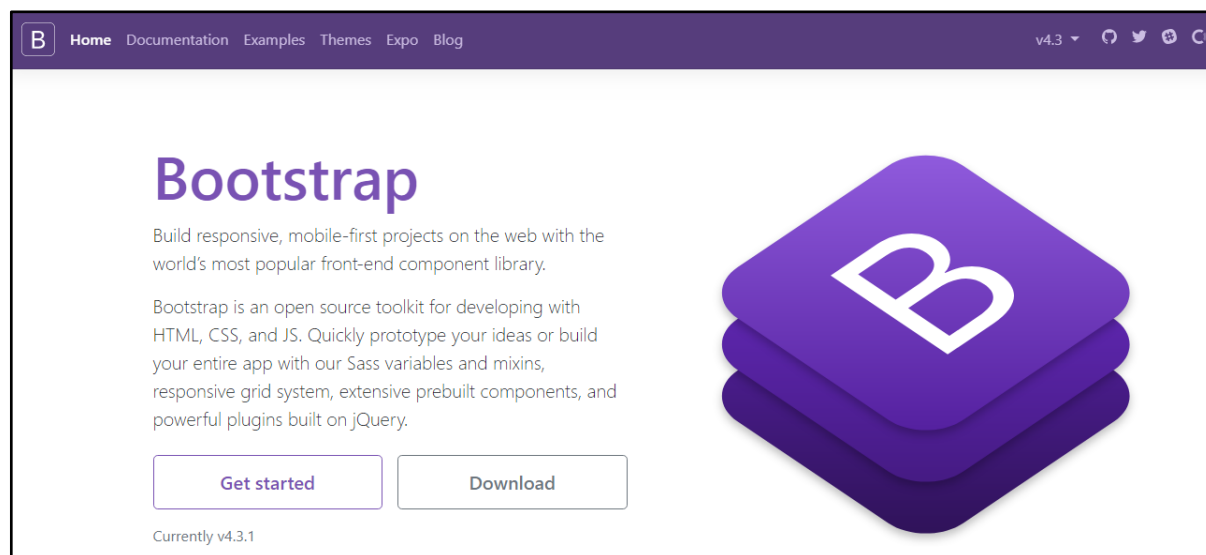
Caso queira trabalhar com vídeos podemos utilizar simplesmente o percentual de largura de 100%, como em nosso primeiro exemplo de imagens responsivas. Você pode utilizar o Media Query com um `display block` ou `none` para exibir ou ocultar o vídeo dependendo da resolução de tela.

Diferente das imagens, os vídeos geralmente são mais maleáveis e fáceis de utilizar, essa técnica também pode ser utilizada como iframes.

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Capítulo 08 - Bootstrap



### Objetivos:

Neste capítulo você irá aprender:

- Como funciona o Framework Bootstrap

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |



## Bootstrap

O Bootstrap é um framework desenvolvido e mantido pelo Twitter, sua principal finalidade é disponibilizar aos desenvolvedores uma ferramenta capaz de criar sites responsivos de maneira rápida, fácil e com uma documentação bem explícita.

O Twitter sempre teve como alvo os smartphones, como havia uma grande dificuldade para os desenvolvedores criarem suas páginas responsivas, surgiu a ideia de criar esse projeto e ser uma referência neste segmento, atualmente o Bootstrap é a ferramenta responsiva mais utilizada no mundo, ganhando versões para outros frameworks como o Angular e o ReactJS.

Para ter acesso a framework basta acessar o site [www.getbootstrap.com](http://www.getbootstrap.com) e começar, abaixo segue a imagem da página inicial.

É possível utilizar o Bootstrap de duas maneiras, a primeira é através de arquivos on-line disponibilizados por várias empresas como o Google e a CDN, quando clicado em Get Started aparecem as opções de CSS e JavaScript:

### CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-gg2y/5G8g7/W80ht器官/wr464Woz78Y7z8c2yh5Wf7BC7hdh7KYFZf3Q3qzZBv3ql" sha384-gg2y/5G8g7/W80ht器官/wr464Woz78Y7z8c2yh5Wf7BC7hdh7KYFZf3Q3qzZBv3ql">
```

### JS

Many of our components require the use of JavaScript to function. Specifically, they require [jQuery](#), [Popper.js](#), and our own JavaScript plugins. Place the following `<script>`s near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

We use [jQuery's slim build](#), but the full version is also supported.

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqUg6VpMjyvYl98j0xhTIDbPb77wFQhV48jOOq" sha384-q8i/X+965DzO0rT7abK41JStQIAqUg6VpMjyvYl98j0xhTIDbPb77wFQhV48jOOq"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-Uh2uZkm1R5OPtNkR1CDwYqVjEHCxBp5ik17vgT0rB764ngM7ykGLzWb+K6fgA" sha384-Uh2uZkm1R5OPtNkR1CDwYqVjEHCxBp5ik17vgT0rB764ngM7ykGLzWb+K6fgA"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVg" sha384-JjSmVg"></script>
```

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



Mesmo sendo repositórios com grande renome, há chances de haver lentidão ou até mesmo falha no carregamento dos arquivos, sendo assim uma dica é baixar os arquivos e deixar no diretório de seu site, mas caso queira fazer algum teste é válido o uso desse tipo de técnica.

A segunda maneira é baixando os arquivos no site do Bootstrap:

## Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v4.3.1** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins

This doesn't include documentation, source files, or any optional JavaScript dependencies (jQuery and Popper.js).

Download

Note que há um aviso informando que não há os arquivos jQuery e Popper.js, caso você queira utilizar algum componente como: dropdown, collapse, carousel, modal, entre outros é necessário o jQuery, para baixar o jQuery basta acessar o site: <https://jquery.com/>

Já o Popper.js é uma biblioteca que trabalha com efeitos como o jQuery, elementos como dropdown e tooltips do Bootstrap utilizam essa biblioteca, caso utilize favor baixar em: <https://popper.js.org>

Voltando ao nosso conceito de Bootstrap, após que baixado você vai se deparar com vários arquivos, não se assuste, não precisaremos utilizar todos.

|  |
|--|
|  |
|  |
|  |
|  |
|  |



Na documentação oficial são explícitos o que fazem cada um desses arquivos, veja na imagem a seguir:

| CSS files                                                                              |                  |              |              |                     |
|----------------------------------------------------------------------------------------|------------------|--------------|--------------|---------------------|
| Bootstrap includes a handful of options for including some or all of our compiled CSS. |                  |              |              |                     |
| CSS files                                                                              | Layout           | Content      | Components   | Utilities           |
| bootstrap.css<br>bootstrap.min.css                                                     | Included         | Included     | Included     | Included            |
| bootstrap-grid.css<br>bootstrap-grid.min.css                                           | Only grid system | Not included | Not included | Only flex utilities |
| bootstrap-reboot.css<br>bootstrap-reboot.min.css                                       | Not included     | Only Reboot  | Not included | Not included        |

| JS files                                                                         |              |              |
|----------------------------------------------------------------------------------|--------------|--------------|
| Similarly, we have options for including some or all of our compiled JavaScript. |              |              |
| JS files                                                                         | Popper       | jQuery       |
| bootstrap.bundle.js<br>bootstrap.bundle.min.js                                   | Included     | Not included |
| bootstrap.js<br>bootstrap.min.js                                                 | Not included | Not included |

Resumindo o que precisamos em nosso site são os seguintes arquivos:

1. bootstrap.min.css
2. jQuery.js
3. bootstrap.min.js

Importante nos arquivos .js é deixar como primeiro na hierarquia o jQuery, em seguida chamar os demais arquivos do Bootstrap, caso essa ordem não seja respeitada os componentes JavaScript não irão funcionar.

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



Os demais arquivos são complementares, e você poderá ler a documentação para saber se tem interesse em utilizar ou não em suas aplicações, mas geralmente apenas os arquivos base já resolvem nossos problemas com o Design Responsivo.

O Bootstrap tem como uma característica forte no desenvolvimento responsivo utilizar o sistema de Grid View, porém com a implementação de Media Queries, veja na imagem abaixo suas características:

|                     | Extra small<br><576px | Small<br>≥576px | Medium<br>≥768px | Large<br>≥992px | Extra large<br>≥1200px |
|---------------------|-----------------------|-----------------|------------------|-----------------|------------------------|
| Max container width | None (auto)           | 540px           | 720px            | 960px           | 1140px                 |
| Class prefix        | .col-                 | .col-sm-        | .col-md-         | .col-lg-        | .col-xl-               |
| # of columns        | 12                    |                 |                  |                 |                        |

Por padrão o Bootstrap trabalha com até cinco resoluções, porém você poderá implementar mais tipos caso desejar.

Nessa etapa do módulo de Bootstrap iremos implementar uma estrutura simples e correta de um Grid View, observe a imagem a seguir:

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7   <!-- CSS -->
8   <link rel="stylesheet" href="bootstrap/bootstrap.min.css">
9
10  <!-- jQuery -->
11  <script src="bootstrap/jquery.js"></script>
12
13  <!-- JS -->
14  <script src="bootstrap/bootstrap.min.js"></script>
15
16  <title>Trabalhando com Bootstrap</title>
17 </head>
```

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |

Estrutura do cabeçalho da página HTML, nela chamaremos os arquivos necessários para utilizarmos as principais funcionalidades do Bootstrap, porém nessa parte do Grid View bastaria apenas o arquivo CSS.

Entre as tags `<body></body>` iremos adicionar o Grid View:

```

19  <body>
20  |
21  |   <div class="container">
22  |   |   <div class="row">
23  |   |   |   <div class="col-12 col-sm-6 col-md-4">Coluna 1</div>
24  |   |   |   <div class="col-12 col-sm-6 col-md-4">Coluna 2</div>
25  |   |   |   <div class="col-12 col-sm-6 col-md-4">Coluna 3</div>
26  |   |   </div>
27  |   </div>
28  |
29  </body>
30
31  </html>

```

Vamos tentar compreender essa estrutura de maneira fácil:

Na linha 21 utilizamos uma div com a class **container**, essa classe container define o tamanho máximo onde os conteúdos serão exibidos, caso você queira trabalhar com 100% da largura da tela basta utilizar a classe **container-fluid**.

Na linha 22 há uma outra div com a classe **row**, as boas práticas pedem para que as colunas estejam dentro de uma linha, assim fica mais organizado nosso código.

Nas linhas 23, 24 e 25 estão nossas classes de Grid View que são:

- col-12, que vai ocupar 100% da largura do container;
- col-sm-6, que irá ocupar 50% da largura do container;
- col-md-4, que irá ocupar 33% da largura do container.

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



Como visto no capítulo de Grid View, o padrão é trabalhar com 12 colunas, e o Bootstrap segue essa mesma ideia. Dentro de cada coluna você pode adicionar textos, imagens, vídeos, iframes, e demais conteúdos para suas páginas.

O Bootstrap também possui componentes como alertas, botões, campos de formulários, tabelas, barras de navegação, entre outras funcionalidades que irão agregar ainda mais valor aos projetos, além de estarem responsivos, abaixo segue uma imagem dos tipos de botões que você encontrará no Bootstrap:



|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |

## Capítulo 09 – Dicas para Desenvolvedores



### Objetivos:

Neste capítulo você irá aprender:

- Dicas Especiais de Design Responsivo

|  |           |
|--|-----------|
|  | Anotações |
|  |           |
|  |           |
|  |           |
|  |           |



## Dicas para Desenvolvedores

Em nossos tópicos sobre o Design Responsivos aprendemos diversas técnicas como: Media Queries, Grid View, Mobile First, Bootstrap, entre outras complementações, porém podemos agregar ainda mais em nossos projetos, sendo assim abaixo há algumas dicas interessantes:

1. Testar compatibilidade com dispositivos móveis: O Google possui o site: <https://search.google.com/test/mobile-friendly>, onde o desenvolvedor pode verificar como o site é visto pelo mecanismo de busca, é totalmente gratuito e fácil de utilizar.
2. Extensão para Chrome e Firefox **Window Resizer**: Ideal para testar seu site em várias resoluções, além de resoluções pré-definidas, você pode criar testes com outras resoluções. Componente simples, leve e bem prático de ser utilizado.
3. Há um site para complementar seus projetos em Bootstrap que é o <https://hackerthemes.com>, nele desenvolvedores encontram estruturas, ferramentas e temas gratuitos.

|           |  |
|-----------|--|
| Anotações |  |
|           |  |
|           |  |
|           |  |
|           |  |