

Desenvolvimento responsivo na web - Padrões e Técnicas

APRESENTAR CONCEITOS, PADRÕES E TÉCNICAS ACERCA DO DESENVOLVIMENTO RESPONSIVO NA WEB.

AUTOR(A): PROF. JORGE HENRIQUE PESSOTA

Apresentação e Fundamentos

A internet está em constante evolução e a forma como a utilizamos também. Hoje é possível acessar a internet a partir de diferentes dispositivos eletrônicos, como *desktops*, *tablets*, *smartphones*, *smart TVs*, *video-games* etc. Isso aumenta o desafio para o desenvolvedor de aplicações web. Isso porque é esperado que o usuário possua uma boa experiência no acesso aos conteúdos publicados, independente do dispositivo que esteja utilizando. Sendo assim, não basta apenas que se construa um bom *layout* para as páginas web. É necessário utilizar padrões e técnicas que garantam que a página desenvolvida seja funcional, independente da forma como ela é acessada. Diante deste contexto, surge o termo “*Design Responsivo*”.

Definição

Design responsivo consiste na construção de páginas web que se adaptem às características de tela do dispositivo eletrônico em uso. Isso significa que o *layout* e o conteúdo das páginas acessadas devam ser dimensionados, ampliados, movimentados ou ajustados, de forma dinâmica, para que fiquem apresentáveis e funcionais para qualquer tipo de tela.

Tecnicamente, o design responsivo é aplicado a partir de HTML e CSS e o seu correto emprego descarta a necessidade de criar diversas folhas de estilo para diferentes dispositivos, o que reflete em menor tempo e custo para o desenvolvimento e para a manutenção de projetos web.

Meta Tag Viewport

Para iniciar o desenvolvimento de uma aplicação web utilizando o conceito de design responsivo, deve-se utilizar a meta tag “*Viewport*” no documento HTML. Esta meta tag indica ao navegador como tratar as dimensões e escalas da página web a ser *renderizada*.

Na imagem a seguir é apresentado um exemplo de utilização da meta tag “*Viewport*”:

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

A propriedade `content="width=device-width"` indica que a largura da página deve seguir a largura da tela do dispositivo utilizado.

A propriedade `initial-scale` indica o nível de zoom inicial quando a página é carregada pela primeira vez pelo navegador. Por padrão, utiliza-se o valor 1.0 como escala.

Containers

Para que se tenha páginas responsivas utilizando *Bootstrap*, todos os elementos visuais da página devem estar em pelo menos um *container*. *Container* pode ser definido como um componente que engloba outros componentes, permitindo assim, um maior controle sobre o *layout*, uma vez que se pode definir margens, *padding*s e posicionamento de um conjunto de componentes visuais.

Em HTML, um *container* possui uma *tag* de abertura e uma *tag* de fechamento. A tag *div* e as tags semânticas *nav*, *header*, *section*, *aside* e *footer* presentes na HTML 5 são exemplos de *containers*.

Há duas classes de *containers Bootstrap*:

`.container`

`.container-fluid`.

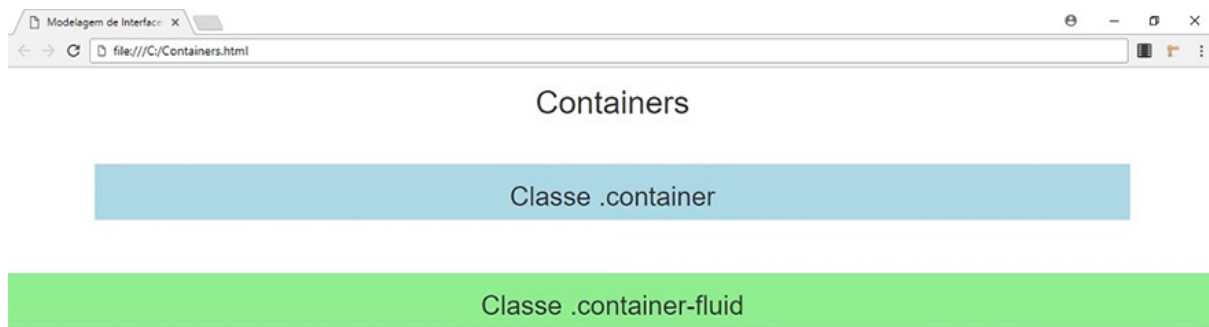
A classe `.container` gera um *container* responsivo com largura fixa e centralizado.

A classe `.container-fluid` provê um *container* responsivo que ocupa toda a largura da tela do dispositivo e que sofre redimensionamento de acordo com a dimensão da tela.

Exemplo:

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.
4.   <head>
5.
6.     <meta charset="utf-8">
7.       <meta name="viewport" content="width=device-width, initial-scale=
8.     <title>Modelagem de Interfaces - Bootstrap</title>
9.     <link href="css/bootstrap.min.css" rel="stylesheet">
10.
11.   </head>
12.
13.   <body>
14.
15.     <h1 style="text-align: center;">Containers</h1>
16.
17.     <br><br>
18.
19.     <div class="container" style="background-color: lightblue">
20.       <h2 style="text-align: center;">Classe .container</h2>
21.     </div>
22.
23.     <br><br><br>
24.
25.     <div class="container-fluid" style="background-color: lightgreen">
26.       <h2 style="text-align: center;">Classe .container-fluid</h2>
27.     </div>
28.
29.   </body>
30. </html>
```

Resultado:



Textos responsivos

É adequado que o tamanho das fontes utilizadas em uma página web sejam redimensionadas de acordo com o tamanho da tela do dispositivo em uso. Para isso, basta utilizar a unidade “vw”, que significa “*viewport width*”. Isso fará com que o tamanho da fonte se adeque ao tamanho da janela do navegador.

1. `<!-- Exemplo -->`
2. `<h1 style="font-size:10vw">Texto Responsivo</h1>`

Imagens Responsivas

Imagens responsivas correspondem a imagens que se ajustam de acordo com a resolução de tela do dispositivo.

Propriedade “width”

Para que se tenha imagens responsivas em páginas web, deve-se utilizar porcentagem para definir a propriedade “*width*” (largura). A utilização de *pixels* e unidades de medidas absolutas não são recomendadas, pois não consideram a resolução de tela do dispositivo como referência.

1. `<!-- Exemplo -->`
2. ``

Propriedades “max-width” e “height”

Também é importante que se defina o tamanho máximo de largura que a imagem poderá assumir, para evitar que ela exceda o seu tamanho original em dispositivos com grande resolução de tela. Para isso, utiliza-se a propriedade “*max-width*”.

Como neste caso, a largura da imagem será alterada dinamicamente, também é importante definir a propriedade “*height*” (altura) como “*auto*”, para garantir as proporções da imagem.

1. `<!-- Exemplo -->`
2. ``

Apresentação de diferentes imagens, de acordo com a largura da janela do browser

A utilização das propriedades “*width*”, “*max-width*” e “*height*”, conforme destacadas acima, garantem o dimensionamento adequado da imagem, de acordo com o dispositivo utilizado, mas não reduz a carga na renderização de uma imagem de grandes dimensões por um navegador de *smartphone* por exemplo. Os *bytes* trafegados no carregamento ainda são referentes ao tamanho real da imagem, independente do tamanho do dispositivo.

Como alternativa para esta situação, é possível determinar imagens de diferentes dimensões, de acordo com a resolução da tela do dispositivo, permitindo assim, que a carga para a renderização de uma imagem seja proporcional ao tamanho do dispositivo. Para isso, utiliza-se o elemento `picture`, presente na HTML 5, em conjunto com *Media Queries*.

Exemplo:

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.
4.   <head>
5.
6.     <meta charset="utf-8">
7.     <meta name="viewport" content="width=device-width, initial-scale=1">
8.     <title>Modelagem de Interfaces - Bootstrap</title>
9.     <link href="css/bootstrap.min.css" rel="stylesheet">
10.
11.
12.   </head>
13.
14.   <body>
15.
16.     <div class="container">
17.       <h1>Picture e Media Queries</h1> <br><br>
18.
19.       <picture>
20.         <source srcset="img/imagem_pequena.jpg" media="(max-width: 767px)">
21.         <source srcset="img/imagem_media.jpg" media="(max-width: 1199px)">
22.         <source srcset="img/imagem.jpg">
23.         
24.       </picture>
25.
26.     </div>
27.
28.   </body>
29. </html>
```

Resultado:



Media Query

Media Query é uma técnica introduzida na CSS 3, da qual possibilita que diferentes regras de estilo sejam aplicadas sob um mesmo elemento, de acordo com o dispositivo em uso. É possível estabelecer condições, como a largura máxima ou largura mínima da tela, que quando atendidas por um dispositivo, determinadas regras de estilos são aplicadas.

Geralmente as *media queries* são inseridas em arquivos CSS externos, onde são definidos os *breakpoints* através da instrução “@media”. Cada *breakpoint* possui uma condição a ser atendida e as regras de estilização a serem aplicadas.

```
1. /* breakpoint - background preto para telas com no mínimo 500 pixels de largura
2. @media only screen and (min-width: 500px) {
3.     body {
4.         background-color: black;
5.     }
6. }
7. /* breakpoint - background amarelo para telas com no mínimo 600 pixels de largura
8. @media only screen and (min-width: 600px) {
9.     body {
10.        background-color: yellow;
11.    }
```

É possível combinar as *media queries* com o conceito “*mobile-first*”. Neste caso, são definidas as regras de estilo para a apresentação da página web em *smartphones* e então são criados *breakpoints*, especificando as dimensões mínimas de largura da tela para a aplicação de outras estilizações, direcionadas para *tablets* ou *desktops*, por exemplo.

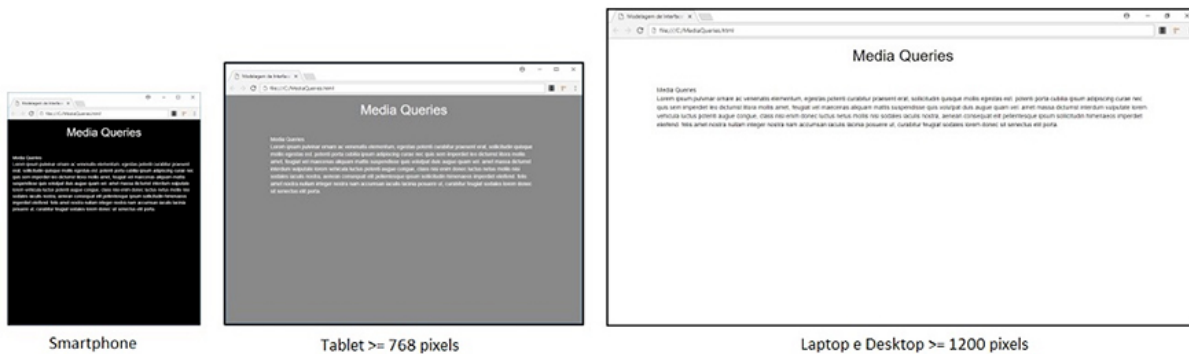
No exemplo a seguir é definida como padrão a cor de fundo preta para a apresentação da seção *body* em *smartphones*, a cor de fundo cinza para a apresentação em *tablets* (largura mínima de 768 *pixels*) e a cor de fundo branca para a apresentação em *laptops* e *desktops* (largura mínima de 1200 *pixels*).


```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.
4.   <head>
5.
6.     <meta charset="utf-8">
7.     <meta name="viewport" content="width=device-width, initial-scale=1">
8.     <title>Modelagem de Interfaces - Bootstrap</title>
9.     <link href="css/bootstrap.min.css" rel="stylesheet">
10.
11.     <style type="text/css">
12.
13.       /* estilização para smartphones */
14.       body {
15.         background-color: black;
16.         color: white;
17.       }
18.
19.       /* breakpoint - estilização para tablets */
20.       @media only screen and (min-width: 768px) {
21.         body {
22.           background-color: grey;
23.         }
24.       }
25.       /* breakpoint - estilização para laptops e desktops */
26.       @media only screen and (min-width: 1200px) {
27.         body {
28.           background-color: white;
29.           color: black;
30.         }
31.       }
32.
33.     </style>
34.
35.   </head>
36.
37.   <body>
38.
39.     <div class="container">
40.       <h1 style="text-align: center;">Media Queries</h1> <br><br>
```

```

41.
42.     <p>Media Queries <br>
43.     Lorem ipsum pulvinar ornare ac venenatis elementum, egestas potenti
44.     curabitur praesent erat, sollicitudin quisque mollis egestas est. po
45.     porta cubilia ipsum adipiscing curae nec quis sem imperdiet leo dict
46.     litora mollis amet, feugiat vel maecenas aliquam mattis suspendisse
47.     volutpat dui augue quam vel. amet massa dictumst interdum vulputate
48.     vehicula luctus potenti augue congue, class nisi enim donec luctus r
49.     mollis nisi sodales iaculis nostra, aenean consequat elit pellentesq
50.     sollicitudin himenaeos imperdiet eleifend. felis amet nostra nullam
51.     nostra nam accumsan iaculis lacinia posuere ut, curabitur feugiat sc
52.     lorem donec sit senectus elit porta. </p>
53. </div>
54.
55. </body>
56. </html>

```



Resultado:

Também é possível definir regras de estilos de acordo com a orientação do *browser*, sendo “Retrato” (*Portrait*) e “Paisagem” (*Landscape*).

```

1. @media only screen and (orientation: landscape) {
2.     body {
3.         background-color: lightblue;
4.     }
5. }

```

Resumo do Tópico

Neste tópico foram apresentados os conceitos, padrões e técnicas acerca do desenvolvimento responsivo na web. Foram abordados itens como: *Meta Tag Viewport*, *containers*, textos responsivos, imagens responsivas e *Media Queries*.

ATIVIDADE FINAL

O que é Design Responsivo?

- A. Design Responsivo consiste na construção de páginas web estáticas, que não se adaptam às características de tela do dispositivo eletrônico em uso.
- B. Design Responsivo consiste na construção de páginas web que se adaptam somente às características de tela de desktops.
- C. Design Responsivo consiste na construção de páginas web que se adaptam às características de tela do dispositivo eletrônico em uso.
- D. Design Responsivo consiste na construção de páginas web exclusivas para smartphones.

O que são Imagens Responsivas?

- A. Imagens responsivas correspondem a imagens com links para outros conteúdos.
- B. Imagens responsivas correspondem a imagens fixas, que não se ajustam à resolução de tela do dispositivo em uso.
- C. Imagens responsivas correspondem a imagens que se ajustam exclusivamente à resolução de tela de tablets.
- D. Imagens responsivas correspondem a imagens que se ajustam de acordo com a resolução de tela do dispositivo em uso.

Qual o benefício de utilizar Media Query?

- A. Com Media Query é possível aplicar diferentes regras de estilo sob um mesmo elemento/conteúdo, de acordo com o dispositivo em uso.
- B. Com Media Query é possível determinar qual dispositivo deve ser utilizado pela página web.
- C. Com Media Query é possível adicionar validação de dados na página web.
- D. Com Media Query é possível adicionar vídeos na página web.

REFERÊNCIA

SILVA, Maurício Samy. Bootstrap 3.3.5: Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos. 1.^a ed. São Paulo: Novatec, 2015c.

TUTORIALS POINT. Bootstrap Overview, Bootstrap Tutorial. Disponível em:
<http://www.tutorialspoint.com/bootstrap/>

(http://www.tutorialspoint.com/bootstrap/bootstrap_tutorial.pdf). Acesso em: 27 de novembro de 2017.

W3SCHOOLS.COM. Bootstrap Get Started. Disponível em:
http://www.w3schools.com/Bootstrap/Bootstrap_get_started.asp

(http://www.w3schools.com/Bootstrap/Bootstrap_get_started.asp). Acesso em 27 de novembro de 2017.

W3SCHOOLS.COM. HTML Responsive. Disponível em:
https://www.w3schools.com/html/html_responsive.asp

(https://www.w3schools.com/html/html_responsive.asp). Acesso em 27 de novembro de 2017.

