

Web design responsivo - Bootstrap

Marcos Tomazini¹, Luiz Fernando Braga Lopes²

MBA EM DESENVOLVIMENTO DE APLICAÇÕES PARA DISPOSITIVOS MÓVEIS –
Faculdade Cidade Verde (FCV) – Maringá – PR – Brasil
(marcos.tomazini@gmail.com), (prof_braga@fcv.edu.br)

Abstract. *This meta-paper aims to discuss the importance of using responsive design in building websites and web applications in order to ensure an adaptable, hotspot design, without compromising the usability for users regardless of the device used: tablet, smartphone, notebook, or smart tv 42-inch, and the use of a framework to facilitate and standardize this process responsiveness called bootstrap.*

Resumo. *Esse artigo tem por objetivo discutir a importância de utilizar o design responsivo na construção de sites e aplicações web, a fim de garantir um design adaptável, agradável, sem abrir mão da usabilidade para os usuários independentemente do dispositivo utilizado: tablet, smartphone, notebook, ou uma smart tv 42 polegadas, e a utilização de um framework para facilitar e padronizar todo esse processo de responsividade chamado bootstrap.*

Palavras chave: design responsivo, layout fluido, bootstrap.

Introdução

Quando olhamos para a história, o ser humano assim como todas as coisas a sua volta, estão em constante evolução. Mensagens que eram passadas por bilhetes com o passar do tempo caíram em desuso, sendo substituídos por fax, cartas, sms e outros meios eletrônicos que existem hoje em dia.

Com os dispositivos eletrônicos não é diferente. Tudo evoluiu, trazendo melhorias para esses dispositivos. Temos smart tvs com resolução de 4k e aparelhos celulares com tela de 320px. Como garantir que seu site atinja ao menos 90% dos dispositivos atuais com tanta variedade de resoluções de tela? A resposta é: Utilizando o Web Design Responsivo.

Uma coisa importante para ter em mente são as diferenças na usabilidade entre dispositivos móveis e Desktops. O papa da usabilidade, Jakob Nielsen, fala que as diferenças são tão brutais

III SEMINÁRIO EMPRESARIAL E III JORNADA DE TI
ISBN: 978-85-68323-04-5

que precisamos de design diferente para atacar esses públicos. Isso pode ser feito de várias maneiras: sites diferentes pra mobile e Desktop; servidor otimizando a página; ou design responsivo e adaptação do design no cliente. (LOPES, 2013, p. 10).

Assim, neste artigo discutimos a importância de utilizar o design responsivo na construção de sites, a fim de garantir um site adaptável, agradável, sem abrir mão da usabilidade para os usuários de dispositivos moveis. Inicialmente, caracterizamos o que vem a ser a design responsivo e seu nascimento. Em seguida, apresentamos a tríade para construir um site adaptável a qualquer tamanho de resolução de tela e a utilização de um framework para tornar esse processo menos moroso e tornar o projeto padronizado onde com isso conseguimos trabalhar com um time de desenvolvedores todos seguindo o mesmo padrão de código chamado bootstrap.

O que é design responsivo?

É uma técnica de estruturação html e css, em que a pagina web se adapta a resolução do browser sem precisar definir diversas folhas de estilos para cada resolução.

O site Estadão (www.estadao.com.br) optou por utilizar sites diferentes para desktop e mobile. Ao abri-lo em um navegador que esteja instalado em um sistema operacional desktop, veremos a versão desktop do site. Se abirmos em um navegador com sistema operacional mobile (android, ios) este abrirá uma outra versão com uma URL diferente da versão desktop. Como apresentados nas figuras a seguir.

Figura 1 – Versão Desktop



Fonte: www.estado.com.br

Figura 2 – Versão Mobile



Fonte: m.estadao.com.br

Um dos problemas que facilmente é detectado na versão desktop deste site são as medidas fixas (pixels). Se redimensionarmos a janela do navegador, perderemos parte do site, obrigando ao usuário visualizar o site sempre em tela cheia. Já na versão mobile, temos o problema ainda maior. O que é definido por mobile? Apenas os

III SEMINÁRIO EMPRESARIAL E III JORNADA DE TI

ISBN: 978-85-68323-04-5

dispositivos smartphone? Se abrirmos o site em um tablet, a versão do site do padrão seria a mobile. Mas o tamanho da tela seria como um computador desktop. Ou seja, iria abrir a versão mobile dentro de uma “tela desktop”, fazendo com que os elementos do site se expandissem, deixando um visual nada agradável. A partir desse problema nasce uma pergunta: Como podemos definir o que é mobile?

Definição de mobile

Definir características para o mobile não é algo simples de se fazer, pois, ambos dispositivos têm (desktop, mobile), como touch, rede móvel entre outras características comuns entre eles. Pode parecer algo exagerado, mas o que temos hoje são os extremos, desktop numa ponta e mobile em outra ponta. E entre esses extremos uma enorme gama de dispositivos que herdaram suas características de ambas as partes. Para Lopes (2013, p.1) “não existe a definição de web mobile. Existe a web, que acessamos tanto do computador quanto de um celular, tablet ou smart tv. É o HTML, CSS e JavaScript que rodam nos navegadores independente do dispositivo”.

Para alcançarmos o maior número de dispositivos precisamos ter em mente a visão de projetar e arquitetar sites responsivos, pois esse é o futuro da web. Uma característica constante da web é a flexibilidade e temos que sempre buscar adaptar nossos sites a essa característica. Segundo Zemel (2013, p. 10), desenvolver para a web única:

Não se trata de uma “moda” ou um hype da internet; não se trata de algo que chegou, vai angariar alguns fãs e sumir na próxima estação. O Web Design Responsivo é uma nova forma de pensar a web e, dentro de pouco tempo, será tão vital e importante aos desenvolvedores e à experiência do usuário quanto o próprio HTML ou o CSS.

Nascimento da web responsivo

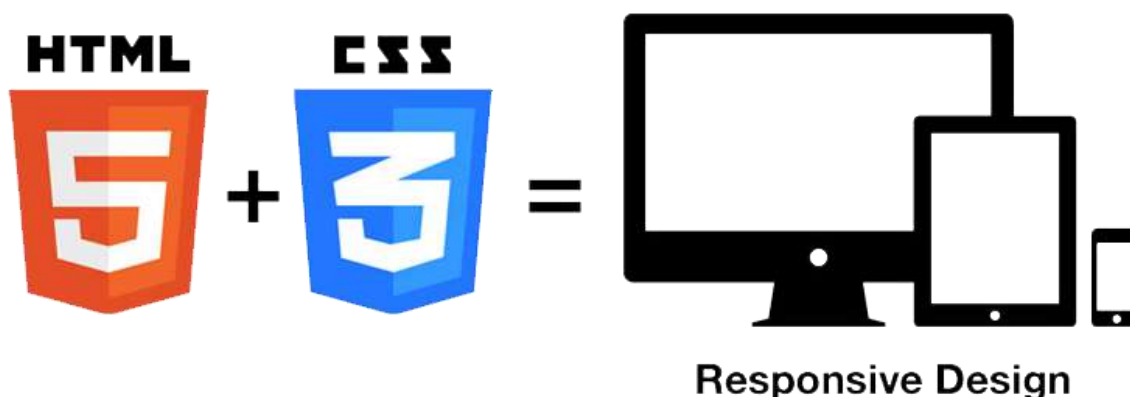
Tudo começou em 2010, pensando em melhorar a sua experiência de navegação, Ethan Marcotte criou o artigo *Responsive Web Design* para o blog *A List Apart*, Ethan explica seus conceitos e sugestões (usando somente as tecnologias que existiam até o momento de sua publicação) para que as páginas fossem projetadas, usando o que ele chamou de web design responsivo. Em seu artigo também somos

apresentados ao arquiteto inglês Christopher Wren. Christopher que certa vez disse que a arquitetura é uma arte que “objetiva a eternidade”. Afinal de contas, todo arquiteto que se preze quer construir um prédio que seja admirado e habitado por séculos. Um exemplo prático é o coliseu de Roma que foi construído no ano de 1979 e ainda é uma das maiores atrações turísticas do mundo. Só que o design digital não tem permanência da arquitetura, projetar para a internet é já estar desatualizado daqui a duas semanas. Temos que ter em mente que para projetar sites com design responsivo hoje em dia não é mais um diferencial e sim uma obrigação se é que queremos que esse site atinja sempre o maior número de usuário possível. Hoje tudo já se encaminha para o responsivo, existe engenheiros de várias áreas e indústrias que já estão desenvolvendo projetos de produtos responsivos. Vidro inteligente que se torna opaco, uma vez que é identificado que existem pessoas no quarto; ambientes que podem se modificar para melhor atender as necessidades de seus ocupantes; são exemplos de projetos que são desenvolvidos para serem responsivos assim como os sites.

Tríade para construção de um site adaptável

Para para desenvolver sites adaptáveis a qualquer tamanho de tela, precisará de três ingredientes: 1. Layout baseado em grid flexível; 2. Media queries; 3. Imagens e Media Flexíveis; onde usaremos as linguagens html5 e css3 como mostra a imagem 1.

Figura 1 – Representação de linguagem.



Fonte: <http://www.covenanttechnologypartners.com/Services/Pages/HTML-5CSS3.aspx>

1. Layout fluído

Segundo Lopes (2013) a grande estrela de um web design responsivo é o layout fluído. Isso quer dizer não usar medidas fixas como pixels (ou pontos, centímetros, milímetros etc.) para programar o design. Não existe mais como copiar as medidas no photoshop da imagem estática que o designer criou com o layout do site. Layout fluído é usar medidas flexíveis e é tão velho quanto o HTML em si. A exemplo do site do estadão, trabalhar com medidas fixas vai contra tudo que a filosofia de uma web adaptável prega. Pois ao redimensionar a janela do navegador os componentes do site não se adaptam, ficando fixos. Trabalhar com layout fluidos nada mais é trabalhar que trabalhar com medidas flexíveis.

2. Media queries

As medidas flexíveis fazem com que o site adapte-se ao redimensionamento do navegador. Mas existem certos momentos, no redimensionamento da janela do navegador, em que o nosso site não ficara bom. Este momento (ponto/gap) é chamado de breakpoint. Segundo Lopes (2013, p. 62) breakpoint “é o ponto de quebra do nosso layout fluído onde uma reestruturação maior é necessária, [...] para ajustar o design e melhorar a experiência do usuário.” Para ajustar nosso design nesses pontos precisaremos das media queries. É com elas que é possível ocultar, fazer aparecer e reposicionar elementos e interações conforme a resolução atual que esteja sendo usada no momento da visitaç o do site. Elas surgiram de um recurso j  existente no css, as media types. Mas diferente das media types, as medias queries n o avalia o tipo de dispositivo, mas sim a resoluç o da tela. Um  timo exemplo do uso de media queries   na utilizaç o para o menu responsivo. Onde teremos um menu na vers o mobile e outro na vers o desktop, sendo definido apenas por media queries em um breakpoint definido.

```
@media (max-width: 320px) {  
    CSS para o menu responsivo  
}
```

Os valores *max-width* e *min-width* equivalem respectivamente **a menor que ou igual a** e **maior ou igual**. Logo o exemplo acima informa que, quando a tela (portview) for menor que 320px ou igual, devem ser aplicadas as regras CSS que estão dentro deste bloco.

Media queries é a solução para quando o layout fluido já não é mais o suficiente. Ela trabalha diretamente com o breakpoint. Muitos sites na web trazem breakpoints definidos, mas cada implementação você deve descobrir seu próprio breakpoint. Lopes (2013, p. 93) sugere que você ache seus próprios breakpoints:

- Abra sua página original no navegador;
- Vá redimensionando a janela devagar até o design parecer ruim — se fez mobile-first, abra pequeno e vá aumentando a janela; senão, abra grande e vá diminuindo a janela;
- Quando achar um ponto em que o design quebra, copie o tamanho da janela e crie uma media query com esse valor lá no seu CSS;
- Recarregue a página, veja se as mudanças melhoraram o design, e continue redimensionando pra achar o próximo breakpoint.
- Algumas ferramentas que podem te ajudar nisso:
 - O responsive mode do Firefox (https://developer.mozilla.org/docs/Tools/Responsive_Design_View);
 - FitWeird (<http://davatron5000.github.com/fitWeird/>), um excelente bookmarklet que te dá o tamanho da tela, inclusive em em;
 - Meu responsive play (<http://sergiolopes.org/responsive-video-play/>).

3. Imagens responsivas

Segundo Zemel (2013) nada adiantaria se o conteúdo se adaptasse as mais diferentes resoluções e se as imagens e outras medias do site, também, não se adaptassem e não fossem flexíveis, correto? Então, por meio de alguma técnicas, é possível sim fazer com que os recursos do site também se adaptem para garantir uma melhor experiência do visitante, independente do dispositivo que ele esteja usando. Em alguns casos também podemos evitar os pixels em imagens. Nas logo marcas podemos optar por imagens vetoriais, pois se adaptarão ao layout do site responsivo sem perder nenhuma qualidade, ou iconfonts. Mas em outros casos não temos como fugir das imagens compostas por pixels como uma galeria de fotos.

Para imagens de pixels em um design fluído sem que elas percam qualidade ou desestruturar o layout todo, existe algumas técnicas. Uma delas e a mais simples é utilizar também o *max-width* e *min-width* nas imagens, o que chamamos de imagens fluídas. Outra técnica é utilizar as media queries para carregar as imagens, dependendo do tamanho da tela. Por exemplo, numa viewport menor que 320px a imagem seria carregada com uma qualidade e tamanho menor, para as telas maiores que 320px seria carregado uma imagem com propriedades que serviram melhor para o site.

```
propaganda {  
    background-image: url(propaganda-mobile.jpg);  
}  
  
@media (min-width: 550px) {  
    .propaganda {  
        background-image: url(propaganda-medio.jpg);  
    }  
}  
  
@media (min-width: 1000px) {  
    .propaganda {  
        background-image: url(propaganda-grande.jpg);  
    }  
}
```

Sendo assim, conseguiríamos atingir os três principais objetivos das imagens responsivas:

- Performance - economizar bytes enviando somente a imagem certa para o tamanho certo de cada tela.
- Qualidade - obter uma qualidade visual na renderização da imagem de acordo com a resolução do usuário que está acessando.
- Direção da arte - usar imagens de conteúdos diferentes para adaptar-se à necessidade de design.

Conhecendo bootstrap

Vimos acima principais técnicas para desenvolver sites adaptáveis a qualquer tamanho de tela, utilizando recursos já existentes no CSS3. Também podemos notar que não é simples desenvolver ou adaptar um site para se tornar responsivo, pois requer muitas linhas de código a mais e alguns cálculos, análises para trabalhar com layouts fluidos. Portanto, conhecer sobre layout fluídos, media queries e imagens responsivas para a construção de um site responsivo é muito importante para os web design antes de iniciar com um framework.

1. O que é o bootstrap

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

O desenvolvimento e a engenharia da web são um trabalho especializado muitos dizem que é uma arte, e cada engenheiro tem a sua própria maneira de codificar. Isso funciona em alguns casos, mas quando há vários engenheiros trabalhando no mesmo projeto com abordagens de codificação ligeiramente diferentes, as inconsistências são inevitáveis. Com o tempo, as inconsistências de engenharia da web podem se transformar em problemas de codificação sérios que criam incerteza e aumentam o tempo de manutenção. O Bootstrap foi uma ferramenta desenvolvida originalmente por

Mark Otto e Jacob Thorton, então engenheiros do Twitter, como uma tentativa de incentivar o uso de uma única estrutura pela equipe de engenharia do Twitter, reduzindo essas inconsistências. A iniciativa do Bootstrap certamente foi bem-sucedida no Twitter, permitindo que toda a equipe trabalhasse com maior rapidez e eficiência e menos inconsistências.

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

Podemos definir o bootstrap como: É uma coleção de vários elementos e funções personalizáveis para projetos da web de código aberto (opensource), empacotados previamente em uma única ferramenta. Em palavras simples, é um conjunto de ferramentas para facilitar o desenvolvimento de sites e sistemas web. Ao projetar um site com o Bootstrap, os desenvolvedores podem escolher quais elementos querem usar. E, o mais importante, podem ter a certeza de que os elementos escolhidos não conflitarão entre si e serão totalmente responsivos funcionando tanto para desktop como quaisquer outros tamanhos de tela. É como um quebra-cabeças, exceto que cada peça se encaixa perfeitamente com as outras, quaisquer que sejam as peças escolhidas.

Figura 1 – Feito para todos



Fonte: <http://getbootstrap.com/2.3.2/>

Esses elementos personalizáveis contidos no Bootstrap são uma combinação de HTML5, CSS3 e JavaScript. Graças ao maravilhoso conceito do software livre, o Bootstrap é aprimorado continuamente.

Figura 2 – Linguagens e logotipo bootstrap



Fonte: <http://www.techmynd.org/demos/html5-multiple-attachment-email/>

A seguir listaremos algumas vantagens e desvantagens do bootstrap:

Vantagens

- É otimizado para o desenvolvimento de layouts responsivos;

- Possui uma interface super amigável e moderna;
- Padronização de códigos;
- Aumento da produtividade;
- Atualmente possui uma grande diversidade de temas;
- Grande quantidade de plugins adaptados ou desenvolvidos para o framework;
- Integração com qualquer linguagem de programação;
- Sistema responsivo;
- Um dos frameworks mais utilizados no desenvolvimento de portais e sistemas do mundo;
- Possui documentação detalhada e de fácil entendimento;
- Download facilitado e totalmente grátis.

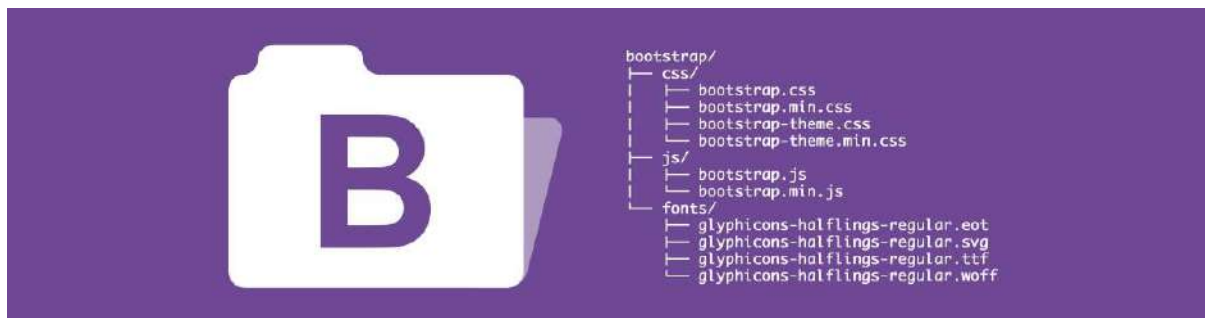
Desvantagens

- Seu código terá de seguir os “padrões de desenvolvimento Bootstrap”;
- Tema padrão e comum do Bootstrap (caso não faça ajustes visuais, ou troque o tema seu projeto se parecerá com outros que também utilizam o Bootstrap).

2. Utilização do bootstrap

A estrutura do bootstrap é simples e seu pacote contém três tipos diferentes de arquivos (CSS, JavaScript e Fonts), que vêm devidamente organizados em suas pastas.

Figura 3 – Estrutura bootstrap



Fonte: <http://getbootstrap.com/2.3.2/>

Montar um layout é simples e rápido utilizando sua documentação. Como toda a estrutura do CSS já vem definida, basta procurar o componente necessário e adicionar

seu código. Em poucos minutos seu layout toma forma e está pronto para uso. E o mesmo acontece com o JavaScript.

O bootstrap funciona com um sistema de grids que possibilita sua divisão em até doze colunas de mesma largura. Essas grids são nativamente responsivas, você pode definir grids diferentes para cada tipo de resolução.

Figura 4 – Sistema de grids responsivas (smart tv, desktop, tablet, smartphone)



A seguir podemos analisar como seria a diferença entre criar um layout responsivo utilizando apenas html5 e css3 para um design responsivo e como ficaria utilizando o bootstrap vide imagem 5 e imagem 6 respectivamente.

Figura 5 – Codificando utilizando html5 e css3 para design responsivo.

```

1      <!doctype html>
2      <html>
3          <head>
4              <meta charset="utf-8">
5              <title>Site Responsivo</title>
6              <style>
7                  /* Layout para celulares: menor ou igual a 480 px. */
8                  body {
9                      background-color:blue;
10                 }
11
12                 /* Layout para tablets: 481 px a 768 px.
13                 Herda estilos de: Layout para celulares.
14                 */
15                 @media only screen and (min-width: 481px) {
16                     body {
17                         background-color: black;
18                     }
19                 }
20
21                 /* Layout para desktops:
22                 769 px até o máximo de 1232 px. Herda estilos de:
23                 Layout para celulares e Layout para tablets.
24                 */
25
26                 @media only screen and (min-width: 769px) {
27                     body {
28                         background-color: red;
29                     }
30                 }
31             </style>
32         </head>
33         <body>
34             <h2>Site Responsivo</h2>
35         </body>
36     </html>

```

Figura 6 – Codificando utilizando bootstrap.

```

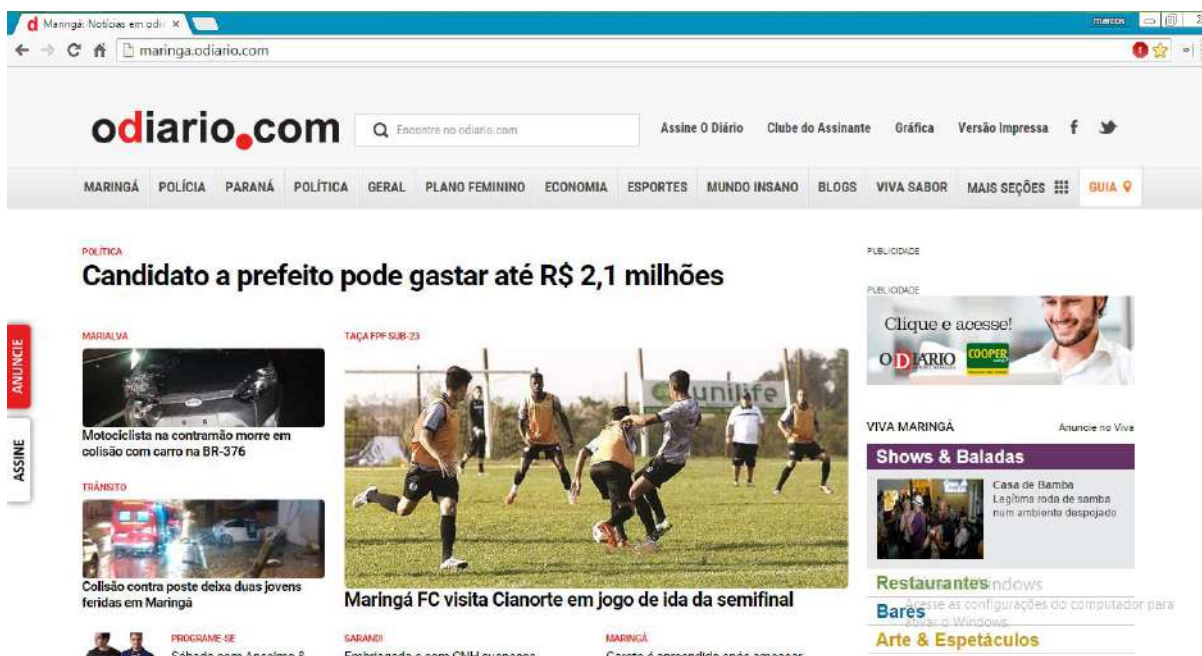
1  <!doctype html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Layout Responsivo</title>
7      <link href="css/bootstrap.min.css" rel="stylesheet">
8      <script src="http://code.jquery.com/jquery-latest.js"></script>
9      <script src="js/bootstrap.min.js"></script>
10 </head>
11 <body>
12 </body>
13 </html>

```

Nos exemplos acima vimos que na imagem 5 levamos em torno de 25 linhas de codificação para iniciar um layout responsivo em nossa página html e a mesma implementação responsiva levamos 13 linhas para iniciar um layout responsivo utilizando o bootstrap, isso pois no bootstrap todas suas codificações necessárias estão nos arquivos no caso acima minificados em `css/bootstrap.min.css` e `js/bootstrap.min.js`.

Um ótimo exemplo de utilização do bootstrap é o site do Odiário (www.odiario.com) tanto em desktop ou em um smartphone o site abre a mesma url e se mostra muito bem fluído utilizando seu sistema de grid. Como apresentados nas figuras a seguir.

Figura 7 – Versão Desktop



Fonte: www.odiario.com

Figura 8 – Versão Mobile



Fonte: www.odiario.com

Considerações finais

No presente artigo foram discutidas as principais técnicas para desenvolver sites adaptáveis a qualquer tamanho de tela, utilizando recursos já existentes no CSS3. Notamos que não é simples desenvolver ou adaptar um site para ser responsivo, pois

requer mais linhas de código e alguns cálculos e análises para trabalhar com layouts fluidos. Mas de fato é recompensador preparar o site para a atual e ou futura arquitetura da web responsiva. Também citamos vários pontos positivos sobre a utilização de um framework onde facilitaria grande parte desse processo onde é possível dar vida a um projeto responsivo em poucas horas, já que sua curva de aprendizado é curtíssima. Portanto, conhecer sobre a tríade (layout fluídos, media queries, imagens responsivas, e bootstrap) para construção de um site responsivo é importante para os web design, sendo que se aprofundar mais neste assunto contribuirá ainda mais no desenvolvimento de um site responsivo mais estruturado, seguro e expansível.

Referências

LOPES, Sérgio. A web mobile: programe para um mundo de muitos dispositivos. São Paulo: Casa do Código, 2013.

MARCOTTE, Ethan. Responsive Web Desing. New York: A Book Apart, 2011.

ZEMEL, Tércio. Web Design Responsivo: páginas adaptáveis para todos os dispositivos. São Paulo: Casa do Código, 2013.

<https://blog.twitter.com/2011/bootstrap-twitter>

<http://www.covenanttechnologypartners.com/Services/Pages/HTML-5CSS3.aspx>

<http://www.techmynd.org/demos/html5-multiple-attachment-email/>

<http://getbootstrap.com/2.3.2/>