

DESIGN DE INTERFACES PARA WEB BASEADOS NO SISTEMA DE GRADE DO BOOTSTRAP 3

INTERFACE DESIGN FOR WEB BASED ON GRID SYSTEM BOOTSTRAP 3

João Pedro Albino¹, Eduardo Martins Morgado², Márcio Jesus de Moraes³, José Antônio Gallo Junior⁴

Abstract — *This paper presents how the Grid system the new version of Bootstrap deals with the responsiveness of web pages. Based on column blocks, the system classifies the Grid screens devices into categories according to their pixel resolution, allowing the development of specific designs for each group. Understand the ways of the Bootstrap Framework Grid System allow the creation of websites, which have a better quality in their functionality, responsive mode for any screen size, thus making your browsing also more attractive, benefiting the user experience and usability. I hope with this work, demonstrate how sessions of a website can be distributed smoothly in different layouts for different screen sizes also using texts, images and videos that fit these layouts.*

Index Terms — *Accessibility, Bootstrap Framework, Grid System, Responsive Design.*

INTRODUÇÃO

Conforme anunciado pelo Google Webmaster Central em fevereiro desse ano, uma nova atualização em seu algoritmo, vem impactando fortemente nos resultados de busca orgânica. O Mobilegeddon, como chamam, segue a premissa de beneficiar sites mobile friendly em sua classificação, que segundo o porta-voz do Google Zineb Ait Bahajji, traria efeitos mais visíveis do que as grandes mudanças anteriores, Panda e Penguin. Tal previsão foi confirmada no último dia 2 de junho no SMX Advanced Seattle, quando Gary Illyes, também do Google, reiterou que a atualização Mobilegeddon de fato teve um impacto maior do que as duas anteriores juntas.

A Stone Temple Consulting apresentou dados dos sites posicionados nas TOP 10 posições para 15.235 termos de pesquisa antes do Mobilegeddon (na semana de 17 de abril) e comparou com seus respectivos posicionamentos após a atualização (na semana de 18 de Maio). Sites não mobile friendly sofreram quedas de posicionamento cerca de 2.3 vezes maior do que aumentos, enquanto os sites mobile friendly apresentaram um crescimento maior do que a queda.

Agora que Mobilegeddon passou, e o tráfego mundial da internet móvel está se aproximando de 60%, é preciso um considerar e implementar o design responsivo e otimização para a web móvel, o que não trata apenas sobre a aparência, mas realmente entender o que precisa ser considerado para criar a melhor experiência de usuário possível, levando em consideração por exemplo as inconsistências dos navegadores, resoluções de tela, preferências do usuário, fontes instaladas, largura de banda e outros fatores que precisam ser tratados para que a informação seja devidamente apresentada.

WEB DESIGN RESPONSIVO

O número de usuários e recursos para os dispositivos móveis vem aumentando exponencialmente nos últimos anos. Segundo Zemel [1], em nosso dia-a-dia, já é possível perceber isso. Note que, nos lugares mais inusitados, as pessoas estão andando não somente com seus smartphones ligados e em pleno uso, mas, também, com seu tablets, e-Readers e muitos outros dispositivos com nomes extravagantes. O número dos usuários de celular vai crescer ainda mais; A Cisco previu que seriam cerca de 788 milhões de usuários até 2015 [2].

Tal acontecimento faz com que Designers sejam forçados a pensar em novas formas de desenvolver páginas da web para usuários móveis, pois uma página não pode ter sua apresentação estática, uma vez que os tamanhos dos dispositivos móveis são muito variados. Em 2010, Ethan Marcotte trouxe uma luz a esta situação chamada web design responsivo (WDR), que hoje, tornou-se uma prática popular em web design para entregar páginas da web em diferentes tamanhos da janela de exibição [1].

No artigo, Marcotte relata:

"O controle que os designers têm no meio impresso e, muitas vezes, desejam ter no meio web, é simplesmente um reflexo da limitação da página impressa. Devemos aceitar o fato de que a web não tem as mesmas restrições e projetar (a web design) para essa flexibilidade"[4].

Devido ao fator de resposta da comunidade por maiores explicações e detalhes sobre o assunto, no ano seguinte foi

¹ João Pedro Albino, Chefe do Departamento de Computação da Faculdade de Ciências de Bauru – UNESP – 2013-atual, jpalbino@fc.unesp.br

² Eduardo Martins Morgado, Coordenador do grupo de pesquisa do Laboratório de Tecnologia da Informação Aplicada – LTIA – UNESP – 1998-atual, emorgado@fc.unesp.br

³ Márcio Jesus de Moraes, Mestrando do Programa de Pós-Graduação em Mídia e Tecnologia (PPGMiT) da Faculdade de Arquitetura, Artes e Comunicação - UNESP, marciojmorais@gmail.com

⁴ José Antônio Gallo Junior, Mestrando do Programa de Pós-Graduação em Mídia e Tecnologia (PPGMiT) da Faculdade de Arquitetura, Artes e Comunicação - UNESP, gallojunior@gmail.com

DOI 10.14684/WCSEIT.2.2015.150-154

© 2015 COPEC

November 19 - 22, 2015, Vigo, SPAIN

lançado o livro que levou o mesmo nome do artigo que revolucionou a maneira de se fazer design na web: Web Design Responsivo [1].

Web Design Responsivo basicamente permite que um site passe a responder ou adaptar-se a um tamanho diferente de janela de exibição, menor ou maior, sem ter que definir um domínio/subdomínio específico para quem utiliza dispositivo móvel. A aparência do site pode ser mantida para que possa ter experiências semelhantes entre diferentes tamanhos de dispositivos. Inicialmente isso foi possível com o uso de meta tag viewport e CSS3 media queries [3].



FIGURA. 1

RWD – UM MESMO WEB SITE EM DIFERENTES DISPOSITIVOS E JANELAS DE EXIBIÇÃO.

Adotando puramente CSS3, media queries permitem que as diretivas possam ser aplicadas condicionalmente, dependendo do resultado de uma consulta booleana baseada em alguma característica do dispositivo, como o tamanho da tela por exemplo. Na prática, isso significa uma única folha de estilo CSS pode conter um conjunto de diretrizes para uma tela com 480 pixels, bem como diferentes diretivas para uma tela de 1200 pixels [5].

A chave para o design responsivo é fazer um design flexível e adaptável, que se ajuste às características do navegador, do dispositivo e do contexto do usuário.

Criar um site responsivo não é fácil. Trabalhar com flexibilidade e adaptação é bem mais complicado que um site fixo em pixels. As ferramentas de desenho ainda não estão preparadas e os designers gráficos costumam ter dificuldades para criar com responsividade em mente [6].

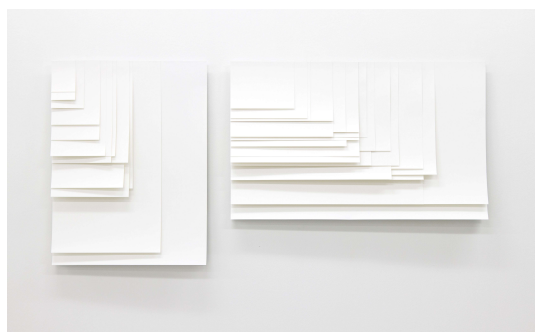


FIGURA. 2

RESPONSIVIDADE – EXEMPLOS DOS TAMANHOS DE TELAS.

Certamente, para um designer iniciante ou até mesmo um especialista, a criação de um site responsivo a partir do zero pode ser complicada, provavelmente por causa de alguns problemas técnicos na **WDR**, tais como a determinação do número apropriado de colunas na grade e o cálculo da porcentagem da largura para cada coluna, a determinação do ponto de quebra correto, e outros aspectos técnicos que geralmente aparecem na fase de desenvolvimento. No entanto, em vez da criação de um web design responsivo a partir do zero, é possível recorrer ao uso de Frameworks, que ajudam a tornar as coisas um pouco mais fáceis no processo [4].

Se usado corretamente, um framework pode obviamente economizar um bom tempo. Geralmente vem com estilos e regras pré-definidas, tais como a largura da grade, dos estilos, tamanhos de fonte, estilos de formulário, redefinição de CSS e outros aspectos usados para construir um site. Sendo assim, não é preciso ter que repetir o mesmo processo desde o início mas basta seguir as instruções para aplicar os estilos e estruturar a marcação [4].

MOBILE FIRST

A maior diferença entre um dispositivo móvel e um Desktop, segundo Lopes é o espaço disponível para o conteúdo. Obviamente sempre é possível utilizar as barras de rolagens, mas é fato que cabe menos informação no celular que na tela do computador [6].

Definitivamente existem diferenças claras no uso de aparelhos móveis em comparação ao uso do Desktop, mais ainda: Mobile não é só diferente, parece ser também mais limitado, afinal a tela é menor, a rede é mais lenta, o hardware é mais lento, a tela de toque é menos precisa que mouse, etc. [6].

Sendo assim segundo Lopes, é muito mais fácil começar seu design tendo em mente as restrições do mobile e depois evoluir para o modo Desktop, que é menos limitado. O contrário é bem mais trabalhoso pois será necessário criar no Desktop sem restrições, e tentar limitar depois conforme for adaptando para o mobile [6].

Em termos práticos, um design mobile-first nos obriga a focar mais e a priorizar melhor o conteúdo, adotando as restrições do mobile, chegando em um design mais simples e funcional, já que não há muito espaço para informações desnecessárias [6]. Sendo assim, quando evoluímos o design para a versão Desktop, o resultado é também uma interface mais objetiva.

O termo mobile-first, nasceu em um post em 2009 pelo especialista mobile Luke Wroblewski, no qual ele falava mais da ideia de design e foco [7].

PROGRESSIVE ENHANCEMENT

O progressive enhancement é um conceito relativamente simples, fácil de explicar, aplicá-lo na prática é bem mais complicado. É preciso uma mudança da maneira de pensar e de estratégia de desenvolvimento, é preciso um excelente

planejamento e muito cuidado na execução. Segundo Lopes, ainda assim é o único caminho para um mundo cheio de browsers e dispositivos diferentes, alguns mais avançados outros mais simples [6].

Uma técnica importante para usar progressive enhancement adequadamente é só usar os recursos novos e avançados em navegadores com suporte para eles, pois usar uma funcionalidade nova em uma página e executá-la no Internet Explorer 6, por exemplo, resultará inúmeros erros por conta de APIs não suportadas [6].

Ainda de acordo com Lopes, é preciso um cuidado a mais, detectar qual navegador suporta qual funcionalidade e só adicionar o recurso nos que são compatíveis, fazendo o que é chamado de detecção de falhas (*feature detection*), algo como detecção de características dos navegadores [6].

A progressive enhancement se baseia em um princípio: tolerância a falhas, que se trata da capacidade de um sistema para continuar a operar quando um erro inesperado acontece, sendo assim, o mestre da detecção de falhas é o projeto Modernizr. Várias detecções são bem mais complicadas e demandam certo esforço, o papel do Modernizr é já cuidar disso e você só se preocupa em escrever um por exemplo, uma instrução condicional simples.

A tolerância a falhas é a razão do progressive enhancement funcionar e faz com que seja possível garantir todo o conteúdo entregue na web é acessível e disponível para todos.

GRACEFUL DEGRADATION

Para Eis, quando um visitante não consegue acessar seu site por causa da resolução, ou por meio de algum dispositivo, ou por algum sistema de voz etc., estamos falando de acessibilidade. A ideia do Graceful Degradation é exatamente essa: dar a melhor experiência possível ao dispositivo/meio que o usuário estiver utilizando sem prejudicar a acessibilidade [9].

De acordo com Koch, Graceful Degradation significa que seu site continuará funcionando mesmo quando visto com um software defasado, em que os efeitos avançados não funcionam [10].

Graceful Degradation parte da premissa que você vai desenvolver seu site com a melhor tecnologia disponível, plug-ins e tudo mais, e mesmo que o usuário acesse a página utilizando um navegador que não renderize os efeitos utilizados, sua navegação vai acontecer “sem problemas”, porque uma vez que a funcionalidade está correta em um browser, são efetuadas mudanças no código até garantir que a funcionalidade rode em todos os navegadores possíveis. Partimos da premissa que a experiência que o usuário terá em browsers mais modernos será suavemente degradada para funcionalidades mais simples em browsers mais antigos [11].

Detectar as funcionalidades de cada browser como vimos aqui nos ajuda a implementar o design adaptativo e tem tudo a ver com design responsivo e suporte a vários dispositivos e navegadores [6]. Um determinado usuário utiliza a resolução

800×600 porque precisa e não porque quer. Embora haja alguns que nem saibam o que é resolução, acessibilidade é tudo sobre a necessidade das pessoas.

LAYOUTS FLUÍDOS

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, pois segundo Lopes, não se pode mais utilizar a imagem estática que o designer criou como layout do site. Layout fluído é usar medidas flexíveis e é tão velho quanto o HTML em si [6].

BOOTSTRAP FRAMEWORK

Bootstrap é um produto de código aberto de Mark Otto e Jacob Thornton que, quando foi lançado inicialmente, eram ambos funcionários da Twitter. Havia uma crescente necessidade de padronizar os conjuntos de ferramentas de front-end (desenvolvimento de interfaces) de engenheiros em toda a empresa. No blog de lançamento, Mark Otto apresentou o projeto como um kit básico contendo diversos componentes web prontos para que você possa desenvolver a sua aplicação web/mobile de forma mais fácil e objetiva, sem necessitar um sólido conhecimento em Javascript e CSS para isso [12].

Mark Otto, escreveu que nos dias anteriores do Twitter, engenheiros usavam quase qualquer biblioteca que eles estavam familiarizados para atender aos requisitos de front-end. Inconsistências entre as aplicações individuais surgiram e tornou-se difícil de manter tais aplicações, sendo assim, o Bootstrap começou como uma resposta a estes desafios e foi rapidamente acelerado durante o primeiro Hackweek do Twitter. Até o final de Hackweek foi desenvolvida uma versão estável que os engenheiros poderiam usar em toda a empresa [13].

O Bootstrap é utilizado para desenhar telas em html, que serão acessadas via navegador web ou dispositivo mobile. Tudo que você precisa saber sobre Bootstrap é, na verdade, html. Com ele, podemos criar sites inteiros e estruturas complexas, mas que podem ser acessadas facilmente em diferentes dispositivos [12].

Segundo Schmitz, a instalação do framework está diretamente ligada a incluir no documento html que você precisa criar algumas bibliotecas Javascript e css. Existem diversas formas para que você possa incluir estas bibliotecas, use a forma que achar mais conveniente para o seu projeto [12].

Outra forma de incluir as bibliotecas do Bootstrap em seu projeto é a utilização de um serviço chamado CDN, que é uma forma de distribuir conteúdo livre através de servidores espalhados pelo mundo [12].

Bootstrap está intimamente ligado ao jQuery, que é um framework Javascript para manipulação da DOM do documento HTML. O Bootstrap precisa do jQuery, então você deve incluí-lo no seu projeto [12].

GRADE BOOTSTRAP

Hoje em dia, todos os principais frameworks CSS vêm com um sistema de grade (Grid System), e Bootstrap não é exceção. O sistema de grade recebeu grandes melhorias quando a versão 3 do Bootstrap foi lançada. A grade do Bootstrap consiste em doze colunas por padrão, mas podem ser criadas variações menores a partir delas. Às vezes, você vai precisar para criar uma grade personalizada para realizar o que você quer [14].

É possível criar um layout de página através de uma série de classes, linhas (rows) e colunas (cols) que vão abrigar um determinado conteúdo. Para definir a largura máxima do conteúdo da página de acordo com a largura da tela, é necessário usar a classe `.container` [15].

Nesta versão 3 do Bootstrap, você pode configurar o tamanho de cada bloco de acordo com o tamanho do dispositivo em questão. O tamanho da tela é determinado através de 4 tipos:

- **Telas de Telefone (Muito Pequenas):** Inferior a 768 pixels, a largura do container é a mesma largura que a largura da tela;
- **Telas de Tablet (Pequenas):** Resolução entre 768 pixels e 992 pixels, a largura máxima do container é de 750 pixels;
- **Telas de Desktop de Pequeno Porte (Médias):** Resolução entre 992 pixels e 1200 pixels, a largura máxima do container é de 970 pixels;
- **Telas de Desktops Grandes:** Maiores que 1200 pixels de resolução, sendo o máximo largura do recipiente é de 1170 pixels.

Os prefixos de classe utilizados para telas muito pequenas, pequenas, médias e grandes são: `.col-xs-`, `.col-sm-`, `.col-md-`, e `.col-lg-`, respectivamente.

Os espaços em branco entre as colunas (gutters), possuem largura de 30 pixels.



FIGURA. 3

SISTEMA DE GRADE BOOTSTRAP – TELAS DE DIVERSOS DISPOSITIVOS COM UM LAYOUT BASEADO EM COLUNAS

Utilizando o Sistema de Grades do Bootstrap, é possível criar linhas e colunas na página usando as classes CSS de linha (`.row`) e coluna (`.col`), onde as linhas terão a função de organizar um grupo horizontal de colunas, que poderão conter outras linhas e essas por sua vez, novas colunas, indefinidamente, de acordo com a necessidade do layout. Normalmente, é possível usar as classes de grade `.col-md-` para desenvolver um sistema que será exibido em dispositivos

pequenos (verticalmente), pois eles se ajustarão horizontalmente em telas grandes. O número total de colunas em uma linha deve não ser maior do que doze, que é o padrão para o número máximo de colunas em qualquer tamanho de dispositivo [15].

col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1	col-md-1
col-md-8								col-md-4			
col-md-4				col-md-4				col-md-4			
col-md-6						col-md-6					

FIGURA. 4

COLUNAS DA GRADE BOOTSTRAP – A SOMA DAS COLUNAS DEVE TOTALIZAR NO MÁXIMO 12.

De acordo com Schmitz, o sistema de grid é a base para que você possa desenhar uma tela que seja compatível tanto com dispositivos mobile quanto desktop. O uso do grid system é importante porque não será você que irá definir os tamanhos e margens de cada bloco, mas sim o framework, de acordo com a largura atual da tela. Isso significa que o Bootstrap pode até definir que um bloco irá ficar abaixo do outro, mesmo tendo a configuração horizontal [12].

Se o designer deseja dividir seu conteúdo, basta usar uma única coluna de largura 12 (Caso 1) [17].

Se desejar dividir seu conteúdo em 2 partes iguais, de mesma largura, basta criar 2 colunas de largura 6 cada uma (Caso 2) [17].

Se você desejar dividir seu conteúdo em 4 partes iguais, de mesma largura, basta criar 4 colunas de largura 4 cada uma (Caso 3) [17].

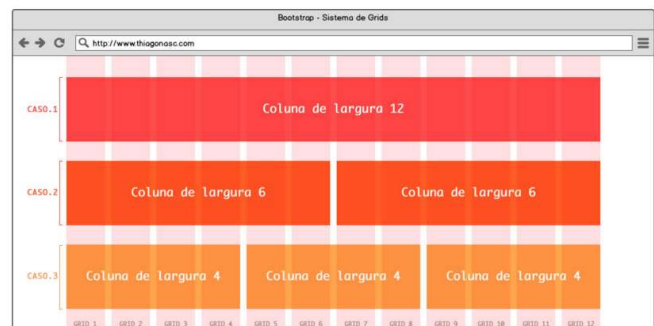


FIGURA. 5

SISTEMA DE GRADE BOOTSTRAP – DIVISÃO EM VÁRIOS LAYOUTS DE COLUNAS

Um offset é, resumidamente, um espaço em branco adicionado à esquerda do bloco de conteúdo, recurso utilizado para a centralização de colunas. Por exemplo, se for preciso centralizar um conteúdo, é possível dividir as 12 colunas em 3 blocos (de 4 colunas cada), com o código abaixo, exemplo para tela grande [12]:

`.col-lg-4 .col-lg-offset-4`

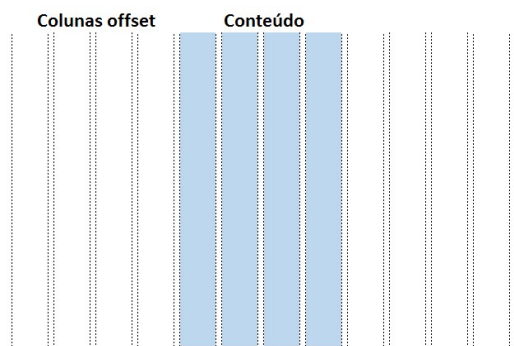


FIGURA. 6

SISTEMA DE GRADE BOOTSTRAP – 4 COLUNAS EM UM BLOCO OFFSET A ESQUERDA PARA CENTRALIZAR O CONTEÚDO DE 4 COLUNAS

COREOGRAFIA DO CONTEÚDO

Uma das questões mais preocupantes com o design responsivo é a de como manter a hierarquia com os elementos na tela quando são redimensionadas e refluídos. Trent Walton foi o primeiro a chamar a atenção desse procedimento chamando-o de coreografia do conteúdo, que mostra como a hierarquia visual fica perdida quando as colunas caem uma abaixo da outra [16].

Como um layout muda de um widescreen para um tablet a um smartphone, o número de colunas é geralmente reduzido de três ou de quatro para um valor para baixo [16].

A solução típica e mais fácil é deixar cair as colunas uma por uma e empilhá-los uns em cima dos outros [16].

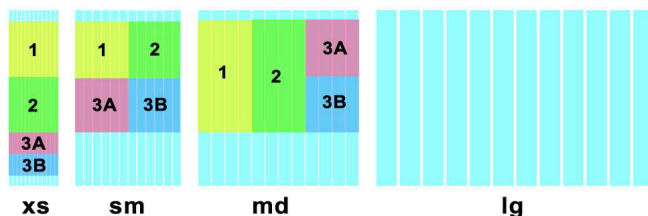


FIGURA. 7

COREOGRAFIA DO CONTEÚDO – A DISPOSIÇÃO DAS LINHAS E COLUNAS É AJUSTADA DE ACORDO COM O TAMANHO DA TELA.

CONCLUSÃO

Frameworks facilitam o trabalho dos designers de interfaces que não precisam se preocupar com cálculos enquanto desenvolver seus layouts, portanto torna importante usá-los sempre que possível, principalmente em projetos ágeis. O Bootstrap, particularmente em sua versão 3, possui um sistema de Grade muito sólido e capaz de permitir o desenvolvimento de um conjunto de páginas que se ajustam ao dispositivo garantindo então uma melhor experiência durante a navegação. Bootstrap aumenta sua produtividade de desenvolvimento, tornando fácil a criação e edição de páginas responsivas.

AGRADECIMENTOS

Agradecemos a todos os nossos familiares e amigos, que são o sustento para o desenvolvimento dos nossos estudos, e também a todos aqueles que pesquisam e buscam criar uma web mais acessível e segura, nossos sinceros agradecimentos.

REFERÊNCIAS

- [1] Zemel, T., "Web Design Responsivo: Páginas adaptáveis para todos os dispositivos", São Paulo: Casa do Código, 2013.
- [2] Cisco, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014–2019. Disponível em: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html. Acessado em 07/08/2015.
- [3] Marcotte, E., "Responsive Web Design". Disponível em: <http://alistapart.com/article/responsive-web-design>. Acessado em 05/08/2015.
- [4] Firdaus, T., "Responsive Web Design by Example: Beginners Guide", Birmingham: Packt Publishing, 2013.
- [5] Datta, A., "Be Responsive: A History of Responsive Design". Disponível em: <http://shout.setfive.com/2012/03/12/be-responsive-a-history-of-responsive-design/>. Acessado em 05/08/2015.
- [6] Lopes, S., "A Web Mobile: Programe para um mundo de muitos dispositivos", São Paulo: Casa do Código, 2013.
- [7] Wroblewski, L., "Mobile First". Disponível em: <http://www.lukew.com/ff/entry.asp?933>. Acessado em 05/08/2015.
- [8] Gustafson, A., "Web Design: Crafting Rich Experiences with Progressive Enhancement", Chattanooga: Easy Readers, 2011.
- [9] Eis, D., "Graceful degradation é tudo sobre Acessibilidade". Disponível em <http://tableless.com.br/graceful-degradation-e-tudo-sobre-acessibilidade/>. Acessado em: 07/08/2015.
- [10] Kock, P., P., "Penser fluide". Disponível em: <http://www.christopher-jablonski.com/fr/reperes/fluidthinking.shtml>. Acessado em 07/08/2015.
- [11] Franklin, A., "Bem vindo a Xangri-lá – Parte 1". Disponível em: <http://tableless.com.br/bem-vindo-a-xangrila-parte-1/>. Acessado em 06/08/2015.
- [12] Schmitz, D., "Bootstrap: Framework front-end para aplicações web e mobile". Victoria, British Columbia: Learnpub, 2014.
- [13] Spurlock, J., "Bootstrap", Sebastopol: O'Really Media, Inc., 2014.
- [14] Niska, C., "Extending Bootstrap: Understand Bootstrap and unlock its secrets to build a truly customized project!", Birmingham: Packt Publishing, 2014.
- [15] Shenoy, A., Sossou, U., "Learning Bootstrap: Unearth the potential of Bootstrap to create responsive web pages using modern techniques", Birmingham: Packt Publishing, 2014.
- [16] Bradley, S., "How To Maintain Hierarchy Through Content Choreography", Freiburg, Smashing Media GmbH, 2013.
- [17] Nascimento, T., "Desenvolvendo com Bootstrap 3: um framework front-end que vale a pena?". Disponível em: <http://thiagonasc.com/desenvolvimento-web/desenvolvendo-com-bootstrap-3-um-framework-front-end-que-vale-a-pena>. Acessado em 05/08/2015.