



## Características projetuais do design responsivo para interfaces digitais

### *Projective features of responsive design applied to digital interfaces*

Lucas Schwartz dos Anjos, Tobias Tessmann Mülling

design, hipermídia, guidelines, web design responsivo

Neste artigo são abordados alguns conceitos técnicos e as tecnologias que são utilizadas no web design responsivo, a fim de compreender características projetuais do design de interfaces, especialmente no que tange a responsabilidade do designer. Neste processo, são abordados elementos do web design responsivo como *grid* flexível, *media queries*, imagens flexíveis, *breakpoints*, entre outras características projetuais. O conceito do web design responsivo se faz relevante especialmente relacionado ao surgimento de diversos dispositivos com diferentes especificidades. Deste modo, projetar uma interface que “responda” a estes dispositivos, bem como compreender suas variáveis de projeto tornam-se importantes no design de hipermídia multi-plataforma.

*design, hypermedia, guidelines, responsive web design*

*This article examines some technical concepts and technologies that are used in responsive web design in order to understand projective characteristics of interface design, regarding the responsibility of the designer. In this process, are discussed elements of responsive design as flexible grid, media queries, flexible images, breakpoints, among other projective characteristics. The concept of responsive design is relevant especially related to the emergence of various devices with different specificities. Therefore, designing an interface that "answer" to these devices as well and understand their design variables are important in multi-platform hypermedia design.*

## 1 Introdução

Web design é uma área do design que vem sofrendo constantes mudanças, principalmente em função do seu amadurecimento e das novas possibilidades que a tecnologia tem oferecido. No passado, as características projetuais do design de interfaces eram mais limitadas que as atuais. Os primeiros websites eram exclusivamente pensados com estruturas fixas. Depois, passamos ao layout flexível, que gerava problemas por não ser tão “flexível” assim: era aplicável apenas aos textos e colunas. Paralelo a isto, cresce aceleradamente a utilização dos dispositivos móveis e a necessidade de interfaces adaptadas para estes. Como uma solução a estes problemas, e associado ao crescente momento de ubiquidade computacional, chegamos ao web design responsivo.

O web design responsivo não é uma opção que deva ser utilizada em todos os websites. Ao projetar, o designer deve avaliar o *briefing* e decidir sobre a utilização, ou não, deste conceito. Deve se verificar o público alvo, as principais vias de acesso, o conteúdo, o prazo, o orçamento disponível, etc. Alguns autores divergem sobre os melhores casos para o emprego do web design responsivo. Parte deles diz não ser apenas para websites pequenos, deixando subentendido que o foco principal seria este. Outros afirmam que é ideal para os websites com

muito conteúdo, acesso e atualização: os portais. Balizando as opiniões, acreditamos que o ideal seja analisar caso a caso.

Segundo Marcotte (2011), falando puramente em termos de front-end *layout* (HTML, CSS e JavaScript), o web design responsivo necessita de três ingredientes principais: *grid* flexível, imagens flexíveis e *media queries*. No entanto, segundo Wroblewski (2011), estes três ingredientes focam apenas no *layout*, o que não é web design responsivo por completo. Por isto, serão abordados aqui não somente os elementos apontados por Marcotte (2011), mas também serão indicados alguns outros conhecimentos importantes para o web design responsivo.

## 2 Características projetuais do design responsivo

Com o intuito de elucidar os elementos projetuais que permeiam o web design responsivo, bem como discuti-los visando a compreensão destes, são apresentados a seguir as principais características de um web design que responde a diferentes resoluções de tela e dispositivos.

### 2.1 Grid flexível

As interfaces baseadas em *grid* apresentam estruturas modulares que permitem maior flexibilidade, mobilidade e facilidade no momento de reorganização do conteúdo apresentado. O *grid* flexível é parte importante da arquitetura da informação do web design responsivo. Não é possível projetar websites responsivos sem a utilização de *grids* flexíveis. A manipulação da largura da *grid* ocorre sempre de forma percentual, o que torna possível a adaptação do website a diferentes resoluções.

Esta forma de utilizar a *grid* flexível, com um valor percentual, é o que permite uma flexibilidade controlada dentro do *layout* (diferente do que acontece no *layout* fluido). Pois a porcentagem é calculada levando-se em consideração o tamanho do objeto e o seu contexto. O designer pode seguir definindo as *grids* de uma maneira semelhante a de projetos tradicionais. A diferença é que elas devem ser pensadas para que se redimensionem de maneira percentual.

Segundo Guerrato (2012), existe uma “formula mágica” quando o assunto é *grid* flexível. Para transformar a medida obtida em pixels para um valor percentual a fórmula é a seguinte: objeto ÷ contexto = resultado. Na programação de um *grid* flexível, os valores em *pixels* recebidos da tela do designer não são codificados em *pixels*. A fórmula citada é utilizada para transformar pixels em valores percentuais, criando uma *grid* flexível.

### 2.2 Imagens flexíveis

As imagens podem vir a ser uma das particularidades no projeto de um website responsivo. As maiores dificuldades são relacionadas a questões de peso (tamanho em kbytes), velocidade de *download* e utilização de banda da internet. Novas tecnologias vêm surgindo para melhor resolver estes problemas.

Alguns designers e desenvolvedores ainda trabalham utilizando as imagens no maior tamanho possível que ela pode ocupar no projeto, ou seja, no tamanho em 100% da área em que ela ocupa na tela planejada para maior resolução. Todavia, desta maneira, mesmo que o acesso de um usuário seja feito sempre via *smartphones* ou *tablets*, visualizando a imagem em um tamanho reduzido, o *download* realizado vai corresponder a imagem com resolução maior. O que vem a ser um problema tratando-se de tempo de espera, gasto de banda, utilização de espaço e tráfego de dados. No que se refere às possibilidades de apresentação visual das imagens nos diferentes dispositivos, temos duas opções principais de mudanças: é possível redimensionar ou *cropar*.

O redimensionamento é feito proporcionalmente de acordo com o tamanho da *div* ou *container*, literalmente escalonando a imagem, mostrando ela por inteiro em um tamanho menor ou maior.

*Cropar* é esconder parte da imagem, e vem associado a ideia da ferramenta “*crop*”, utilizada nos principais editores de imagem. É a alteração dinâmica que define o quanto a imagem é mostrada ou escondida. Neste caso a imagem não muda de tamanho, apenas é alterada a sua

área visível. Ainda no *crop* é possível definir se ele deve ser feito na altura, largura ou em ambos. Também podem ser definidos os pontos de ancoragem. A partir do conteúdo mostrado na imagem, é possível optar por ancorar à direita, à esquerda, abaixo, acima, ou ao centro.

Existem dois tipos de imagens flexíveis: imagens de primeiro plano e imagens de fundo. As imagens de primeiro plano são as imagens de conteúdo; as imagens de fundo são o *background* do website. Na maioria dos casos, quando a imagem de fundo é meramente decorativa, o tratamento ideal para adaptação é o *crop*. O exemplo a seguir (figura 1) mostra o funcionamento de ambos os tratamentos. Enquanto a ilustração é cortada, a marca é redimensionada.

Figura 1: Cropando (cortando) e redimensionando.



### 2.3 Media Queries

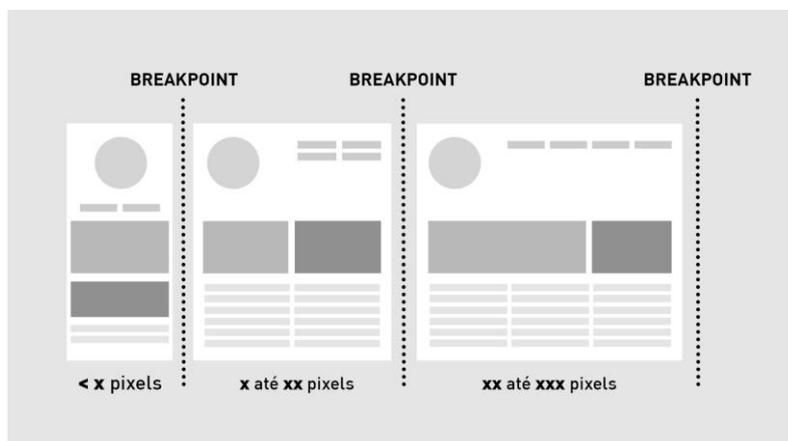
As *Media Queries* são as expressões lógicas de CSS3 que são utilizadas para mudar o *layout* dos websites em diferentes *devices* sem mudar o conteúdo. É a tecnologia da parte da programação que detecta a resolução, orientação, e demais parâmetros, fazendo a conexão com a “folha de estilos” correta a ser apresentada.

As *Media Queries* verificam não só o dispositivo, mas também suas capacidades. São uma espécie de questionário de verdadeiro ou falso que é aplicado ao navegador. Através delas é possível checar altura e largura do navegador, orientação, resolução, etc. Permitem muito maior controle sobre o *layout*. Elas setam os *breakpoints*, definem os *ranges* e linkam a determinada folha de estilos.

### 2.4 Breakpoints

Os *breakpoints*, traduzindo em português como “limites de quebra” ou “pontos de quebra”, são os pontos limites referentes ao tamanho de tela que são utilizados pelas *Media Queries* para decidir apresentar uma ou outra folha de estilos. Os *breakpoints* limitam um intervalo de espaço dentro do qual o *layout* é flexível. Quando um *breakpoint* é ultrapassado, a folha de estilos é trocada. A figura 2 ilustra a troca que ocorre a cada *breakpoint*.

Figura 2: Breakpoints.



Alguns designers afirmam que não há uma norma quanto ao número ideal de *breakpoints*, tratando como única regra a garantia de que o *layout* sempre seja visualizado de maneira adequada em qualquer tamanho de navegador. Outros fazem estudos frequentes sobre os tamanhos de tela existentes no mercado e quais são os mais utilizados. Obviamente, existem alguns tamanhos de tela mais comuns utilizados atualmente. No entanto, devido a ubiquidade computacional e o lançamento de diferentes *devices*, o tamanho “padrão” das telas pode ser mudado constantemente.

O número de *breakpoints* definido em um projeto tem, enquanto fator positivo permitir uma experiência em uma maior diversidade de dispositivos e tamanhos de tela. Por outro lado, exige um incremento na complexidade do projeto de interface, podendo levar a um aumento de custos e tempo de desenvolvimento.

## 2.5 Navegação

A utilização de menus na posição horizontal é bastante comum atualmente. No entanto, não é possível manter esta aparência quando aplicamos o web design responsivo para telas pequenas como as dos *smartphones*. Eles precisam ser adaptados para manter uma boa usabilidade, sem ocupar muito espaço na tela, para que a maior área possível seja ocupada por conteúdo. Além disto, para tamanhos menores a mutação deve ser mais concreta, tendo em vista que o usuário já reconhece os padrões de interação para *mobile*, segundo Woods apud Martins, Godinho e Rabenhorst (2012).

Quando pensamos em menus para *tablets*, esta transformação muitas vezes não precisa ser tão impactante, pois as resoluções e tamanhos de tela são maiores do que os *smartphones*, muitas vezes já comportando de forma adequada o menu inteiro, de modo parecido com o apresentado no computador pessoal. Nesse caso, a adaptação necessária limita-se ao *touchscreen*.

Existem diferentes possibilidades para solucionar este problema. Algumas das principais consistem em mudar o formato de exposição de acordo com a plataforma de acesso. Segundo Eis (2011), é possível “transformar o menu em um *selectbox* (*combobox*), tirar da interface alguns itens que podem não ser interessantes para usuários de *smartphones*, ou reformatar seu design para que caiba de forma funcional em telas pequenas.”

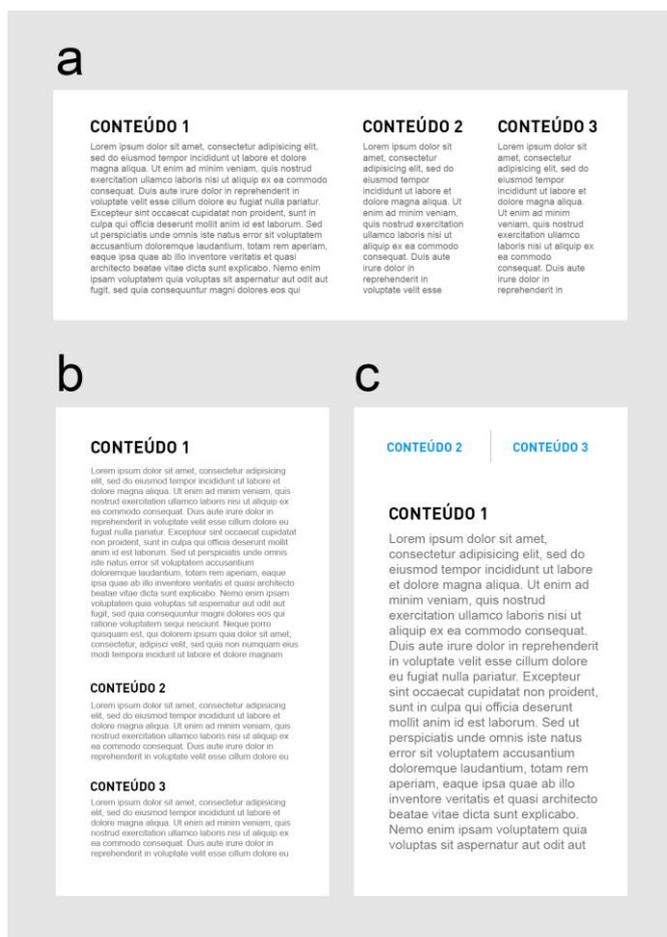
## 2.6 Hierarquia

No web design responsivo podemos diminuir alguns elementos proporcionalmente e alterar a posição de outros conteúdos, de tal maneira que tudo pode ser apresentado, mesmo em uma tela diminuída. No entanto, Knight (2011) avisa que nem sempre mostrar todo este conteúdo é a melhor solução. Existem boas práticas para projetar para ambientes *mobile*: navegação simples; conteúdo mais focado; listas e linhas no lugar de múltiplas colunas.

Deve haver maior preocupação com a largura do que com a altura. A rolagem vertical é bastante aceitável, não prejudica a experiência. Já a rolagem horizontal, deve ser evitada ao máximo. O web design responsivo não se trata apenas de *layouts* flexíveis, com alcance para variadas plataformas e tamanhos. Segundo Knight (2011), é também sobre habilitar o usuário a acessar todo o conteúdo. O CSS nos permite mostrar e ocultar conteúdos com facilidade.

Na figura 3 podemos observar duas adaptações diferentes para o mesmo caso. Na imagem “b” as colunas laterais simplesmente foram apresentadas abaixo da coluna principal. Já na imagem “c”, percebemos que houve preocupações com as questões que envolvem as boas práticas para *mobile*. O conteúdo é mais focado, mas ainda é possível acessar o conteúdo relacionado nos links em azul.

Figura 3: Escondendo conteúdo.



## 2.7 Touchscreen x cursor

As *touchscreens* (telas sensíveis ao toque) possuem diretrizes de design de interação diferentes das de *cursor* por apresentarem capacidades diferentes. Para projetar para ambos é preciso aliar algumas características comuns entre os dois e contemplar todas as restrições de ambos.

Em interfaces *touch*, não é possível sobrepor o *cursor* a um elemento, porque não existe o estado *hover*. Também não são utilizadas teclas de atalho para executar ações. O “clique” ocorre direto, deixando mais próxima a interação e o seu objetivo. Segundo Eis (2011), “normalmente as ações importantes estão expostas na interface, facilitando o acesso rápido. Isso é muito importante porque nos ensina criar interfaces mais intuitivas, com a curva de aprendizado menor.”

Uma das principais preocupações ao projetar para *touchscreen* se refere à área de segurança necessária para efetuar o toque, ou seja, a interação:

Um estudo do MIT, Dandekar, Raju e Srinivasan (2003), mostrou que a área média do toque de um adulto varia entre 1.6 à 2.0 centímetros, o que faz com que a necessidade de espaçamento entre elementos e tamanho de objetos suscetíveis a toque precise ter um tamanho relativo a estas medidas para facilitação da utilização pelo usuário e tornar desnecessária a utilização do zoom. A importância de se seguir esta regra é vital para a usabilidade da aplicação. (Martins, Godinho e Rabenhorst, 2012: 8)

De acordo com este estudo do MIT, convertendo para *pixels*, um tamanho seguro estaria entre 45 a 57 *pixels*.

## 2.8 Contraste

Segundo Martins, Godinho e Rabenhorst (2012), outra questão interessante de ser pensada ao projetar web design responsivo é a do contraste. Principalmente nas telas projetadas para os dispositivos menores, onde muitas vezes a utilização ocorre ao ar livre e em condições climáticas variáveis. Martins, Godinho e Rabenhorst (2012: 08) nos dizem que “a tela da grande maioria dos celulares não se comporta da mesma maneira sob condições de forte iluminação solar. Um meio de atenuar esta condição é elevando os níveis de contraste entre os elementos de acordo com o dispositivo.”

## 2.9 Custos

Projetar soluções sólidas como websites responsivos tem um custo maior do que projetar websites comuns, pois exige um número maior de *layouts* feitos pelo designer e um número maior de folhas de estilos CSS codificadas. Em outras palavras, demanda mais tempo de desenvolvimento ou envolvimento de um número maior de profissionais.

Segundo Simpson (2011), pode demorar cerca de 20 a 30% mais tempo desenvolver um website responsivo, mas ainda é mais rápido do que criar um adicional website mobile ou aplicativo. No entanto, pode ser mais barato à longo prazo, pois demandará menores custos de manutenção e não exigirá retrabalho à medida em que surgirem novos dispositivos.

## 2.10 Testes

Uma importante etapa de qualquer projeto é a etapa de testes. Os projetos de web design responsivo exigem um trabalho de testes ainda maior frente às diversas possibilidades de cenários possíveis. É preciso realizar estes testes nos mais diversos dispositivos para evitar possíveis falhas, desatenção, e também identificar pontos a serem melhorados.

Segundo Guerrato (2012), nada substitui o teste no próprio dispositivo, e mesmo que o designer não tenha acesso a vários *devices*, é aconselhável pedir para outras pessoas navegarem e informarem sobre a experiência com determinado projeto.

Apesar da preferência por testes práticos nos dispositivos, é possível simular dispositivos virtualmente através de diferentes ferramentas. O ResponsiveTest (<http://responsivetest.net/>) realiza *previews* do website em muitos modelos e tamanhos de monitores *desktops*, *smartphones*, *tablets*, entre outros. Além disto, permite girar a tela para testar o website em outra orientação.

## 2.11 Outras considerações

O web design responsivo tem sido adotado aos poucos, devido a complexidade de projeto associado. Quando se inicia o projeto de um website responsivo, ele deve ser feito do início ao fim: todas as páginas devem ser responsivas. É necessário um grande planejamento, para não interromper o processo em andamento e, conseqüentemente, evitar uma experiência de uso frustrante ao usuário.

É desejável que o designer entenda um pouco sobre como se dá o funcionamento estrutural do código, o conhecimento básico da programação, para facilitar o diálogo. O designer não termina completamente seu trabalho para apenas repassar ao programador, pois não deve-se trabalhar com o processamento de tarefas encadeadas. O processo exige ligação constante entre as partes envolvidas, que devem atuar como um time. O programador pode apontar soluções que melhorem o desempenho e exijam mínimas mudanças de design. O trabalho do designer também consiste na reutilização e reorganização dos elementos do website de forma inteligente, minimizando o retrabalho para cada *layout*.

É interessante também compreender o design para *mobile* quando se tem interesse em trabalhar com web design responsivo, principalmente para montar de maneira mais adequada as telas com menor *breakpoint*. Devido ao crescente número de usuários *mobile*, muitos autores tratam sobre o tema “Mobile First”.

Todas essas características específicas, técnicas e conceitos que ainda estão sendo definidos, nos mostram que existe a necessidade de interação direta entre as diversas áreas e indivíduos envolvidos na construção de websites responsivos. O web design responsivo deve

unir esforços de toda equipe a fim de proporcionar uma adaptação multiplataforma adequada, e conseqüentemente, uma melhor experiência ao usuário.

### 3 Considerações finais

Para projetar websites adequados a mobilidade e a ubiquidade computacional, o web design responsivo se utiliza das possibilidades oferecidas pelo seu meio flexível. O web design responsivo não é um projeto perfeito, absoluto ou sem falhas, mas é o mais completo diante do cenário atual que vem se desenhando frente a diferentes tecnologias. É uma proposta a ser seguida, testada e remodelada. Sendo assim, é importante também que se criem questionamentos e discussões acerca do futuro.

Este artigo preocupou-se em apresentar características relacionadas ao projeto de design, ao invés do foco em códigos e linguagens de programação, particular aos desenvolvedores. Cabe aos designers compreender também do ponto de vista técnico as particularidades concernentes a um projeto de interface, e que venha a responder a diferentes dispositivos.

### Referências

- EIS, D. 2011. Introdução ao Responsive Web Design. In: *Tableless*. <<http://migre.me/9LAWM>>, 04/07/2014.
- GUERRATO, D. 2012. Design Responsivo II - Grids e Texto. In: *Popup Design: Blog*. <<http://migre.me/9LAXf>>, 04/07/2012.
- KNIGHT, K. 2011. Responsive Web Design: What It Is and How To Use It. In: *Smashing Magazine*. <<http://migre.me/9LAWx>>, 02/07/2014.
- MARCOTTE, E. 2011. *Responsive Web Design*. Nova Iorque: A book apart.
- MARTINS, C.; GODINHO, R.; RABENHORST, R. 2012. *Crossroads. Um sistema de resposta multiplataforma*. Pelotas.
- T. A. 2012. Finger-Friendly Design: Ideal Mobile Touchscreen Target Sizes. In: *Smashing Magazine*. <<http://migre.me/9LAVe>>, 04/07/2012.
- WROBLEWSKI, L. 2011. An Event Apart: The Responsive Designer's Workflow. In: *LukeW: Writings*. <<http://migre.me/9LAUu>>, 02/07/2014.

### Sobre os autores

Lucas Schwartz dos Anjos, BA, UFPEL, Brazil <[contato@lucasanjos.com](mailto:contato@lucasanjos.com)>

Tobias Tessmann Mülling, MsC, UFPEL, Brazil <[tmulling@gmail.com](mailto:tmulling@gmail.com)>