

# WEB DESIGN RESPONSIVO

**Maurício Samy Silva**

Novatec

Copyright © 2014 da Novatec Editora Ltda.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998.

É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates

Revisão gramatical: Marta Almeida de Sá

Editoração eletrônica: Carolina Kuwabata

Capa: Carolina Kuwabata

ISBN: 978-85-7522-392-5

Histórico das impressões:

Junho/2014          Primeira edição

Novatec Editora Ltda.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

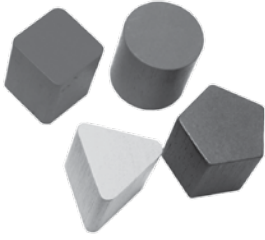
E-mail: [novatec@novatec.com.br](mailto:novatec@novatec.com.br)

Site: [www.novatec.com.br](http://www.novatec.com.br)

Twitter: [twitter.com/novateceditora](https://twitter.com/novateceditora)

Facebook: [facebook.com/novatec](https://facebook.com/novatec)

LinkedIn: [linkedin.com/in/novatec](https://linkedin.com/in/novatec)



## CAPÍTULO 1

# Introdução ao Design Responsivo

Neste capítulo, será feita a apresentação do design responsivo, relatando-se como surgiu, suas finalidades e destinação. Será feita uma breve descrição histórica da evolução das técnicas de construção de layout, mostrando-se como eram desenvolvidos os layouts desde a época do surgimento da web até os dias atuais. Estudaremos os princípios que regem o design responsivo, seus componentes básicos, bem como as tecnologias que constituem os seus pilares. Discutiremos os benefícios e mitos do design responsivo e também introduziremos o conceito de "mobile first".



Neste livro adotamos, sempre que considerado conveniente, a sigla inglesa RWD (Responsive Web Design) para nos referirmos ao termo design responsivo.

### 1.1 Breve visão histórica de layouts Web

Em 1990, Tim Berners-Lee, o inventor da web e atual diretor do W3C – World Wide Web Consortium (Figura 1.1) –, criou o protótipo de um navegador para rodar em computadores da NeXT.

Inicialmente o navegador foi chamado de WordWideWeb (Figura 1.2), e, posteriormente, renomeado como Nexus, a fim de evitar confusão com a Word Wide Web.

O Nexus suportava um mínimo de marcação HTML e estilização nativas. O suficiente para formatar textos, na sua maioria de caráter científico, afinal a finalidade da invenção era a de compartilhar documentos científicos entre os cientistas ao redor do mundo. Não havia preocupação com layouts, e os

conteúdos, estritamente textuais, eram apresentados com uma formatação mínima em um layout de uma coluna.

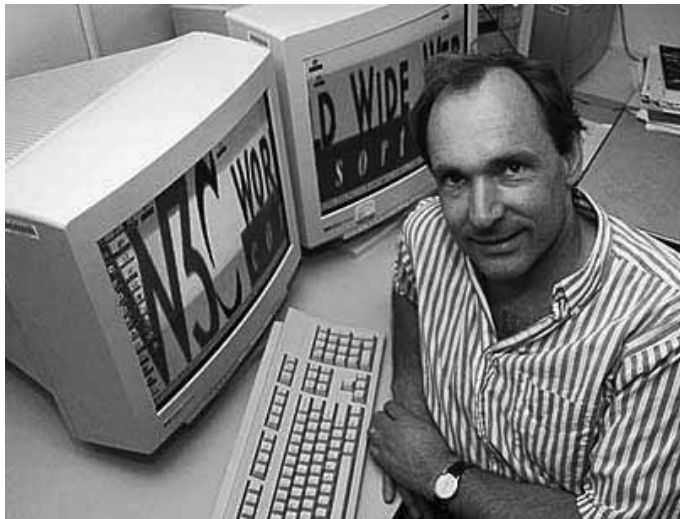


Figura 1.1 – Tim Berners-Lee

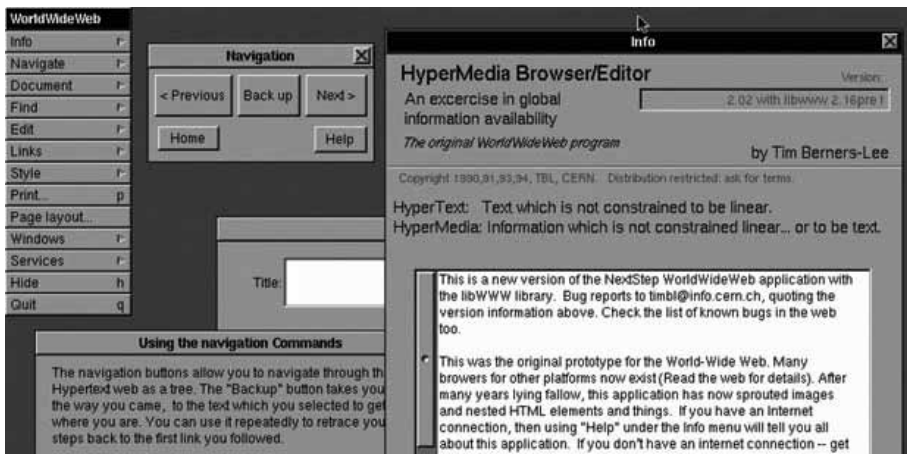


Figura 1.2 – Navegador WorldWideWeb.

Em 1992, no NCSA – Centro Nacional de Aplicações para Supercomputadores da Universidade de Illinois –, foi desenvolvido o primeiro navegador gráfico para a web, que durante quatro anos foi responsável pela popularização da internet. Foi com o navegador denominado Mosaic (Figura 1.3) que tudo começou.



Figura 1.3 – Navegador Mosaic.

Esse navegador introduziu o elemento `img` para apresentação de imagens, os elementos para criar listas aninhadas e os elementos para criar campos de formulário para preenchimento de textos.

A partir daí a criação de layouts simples era feita com uso do elemento HTML pre criado pela HTML2, lançada em 1994.

Em 1997 foi lançada a HTML3.2 e, nela, a criação do elemento `table` viria revolucionar a web, pois era oferecida ao designer e ao desenvolvedor uma poderosa ferramenta para dispor conteúdos na tela e criar layouts complexos e de grande impacto visual em contraste com a até então apresentação de conteúdos, na sua maioria textuais, dispostos verticalmente em coluna única.

Tabelas continuam sendo uma fantástica ferramenta para criar layouts, os mais complexos que o leitor possa imaginar, bem como torná-los fluídos e adaptáveis a diferentes resoluções de tela de uma maneira muito simples.

Apesar de ser uma ferramenta fantástica, infelizmente, vários fatores inviabilizam, nos dias atuais, o uso de tabelas para criação de layouts. Entre eles, destacamos:

- dificuldade extrema de manutenção do código;
- é muito difícil e as vezes impossível fazer até mesmo mínimas alterações no layout;
- criam layouts inacessíveis a vários agentes de usuário, tal como leitores de tela e robôs de busca;
- interferem diretamente com a performance;
- é impossível desvincular a ordem de marcação da ordem de apresentação;
- é impossível determinar a hierarquia da marcação com sérios prejuízos para SEO.

Observe na figura 1.4 um típico layout com tabela.

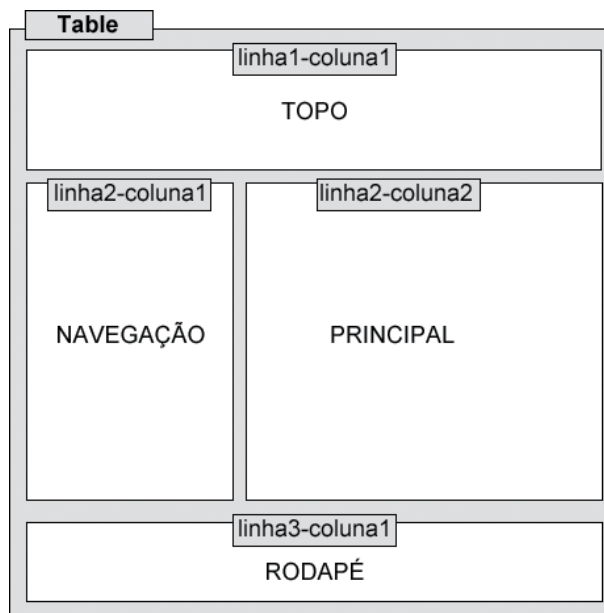


Figura 1.4 – Layout com tabelas.

O passo seguinte no aperfeiçoamento dos mecanismos para criação de layouts foi o surgimento das CSS1, em dezembro de 1996, que apesar de terem sido lançadas um mês antes da HTML3.2, que criou o elemento `table`, somente ganharam força alguns anos depois, no Brasil, por volta de 2003.

As CSS mudaram radicalmente a então metodologia de uso de tabelas para criar layouts, introduzindo uma técnica cuja filosofia de desenvolvimento preconiza a separação da marcação de conteúdos da sua apresentação. Criou-se o conhecido e propagado mantra tão festejado nos idos de 2005.

HTML para estruturar e CSS para apresentar.

A apresentação do layout no dispositivo do usuário é função única das CSS. Isso trouxe como consequência a criação de códigos cuja manutenção e atualização são extremamente simples, amigáveis a mecanismos de busca e indexação, acessíveis a tecnologias assistivas e acima de tudo capazes de serem fluídos ou fixos em função do dispositivo do usuário.

Contudo o desenvolvimento de layouts com uso das CSS2 ainda apresenta várias inconsistências entre plataformas e navegadores e até mesmo entre diferentes versões do mesmo navegador.

Por exemplo, a propriedade `float` das CSS cuja presença é quase obrigatória em uma folha de estilos para definir um layout ainda não é consistentemente suportada de maneira uniforme por diferentes navegadores e obriga autores a usar artifícios (hacks) para servir os estilos de forma crossbrowser.

As CSS3 preveem diferentes módulos destinados à construção de layouts que certamente irão facilitar a tarefa, contudo tais módulos, à época que este livro foi lançado, ainda se encontram em fase de estudos no W3C, e não é prudente utilizá-los.

Por outro lado, com o aparecimento, que se deu a partir de 2011, dos hoje incontáveis frameworks CSS, a tarefa de construção de layouts foi facilitada. Neste livro não adotaremos um framework. Entendemos que o aprendizado da codificação "a unha" fornecerá ao leitor bases sólidas para decidir sobre o uso de um deles no seu projeto e até mesmo proporcionar condições para que o leitor crie seu próprio framework responsivo.

## 1.2 Design responsivo entra em cena

No dia 25 de maio de 2010, Ethan Marcotte escreveu uma matéria no conhecido site AlistApart <http://alistapart.com/article/responsive-web-design> (<http://kwz.me/6B>) que mudaria radicalmente a filosofia e os métodos de desenvolvimento de layouts.

A matéria com o título "Responsive Web Design" começa citando John Allsopp, que disse:

O controle que os designers têm sobre conteúdos publicados na mídia impressa e tão desejado para aplicação na web é, simplesmente, consequência das restrições que a página impressa impõe. Nós devemos nos conscientizar de que a web não impõe as restrições da mídia impressa, ela necessita de design flexível. Devemos, acima de tudo, "aceitar o fluxo e o refluxo das coisas".

E fica claro que Ethan buscou, nas palavras de John, a motivação para escrever a matéria.

Na matéria que escreveu, Ethan demonstra com uso de um exemplo prático como criar uma interface que se adapte às diferentes resoluções de tela e, no final do artigo, afirma o seguinte:

Web design responsivo, finalmente, nos oferece um caminho a seguir, o que nos permite "projetar para o fluxo e o refluxo das coisas".

Não é exagero afirmar que no dia 25 de maio de 2010 nasceu o design responsivo trazido ao mundo por Ethan Marcotte, este simpático jovem mostrado na figura 1.5.



*Figura 1.5 – Ethan Marcotte, o pai do design responsivo.*

### **1.2.1 Etimologia responsiva**

Para escrever sobre etimologia responsiva eu convidei o Bernard De Luna, que escreveu o presente subitem deste livro.

Bernard De Luna já palestrou na HTML5DevConf em San Francisco, no WWW do W3C e em mais de sessenta eventos pelo Brasil.



Foi *team leader* de produto na Petrobras, *creative director* na Melt DSP e tem atuado nos últimos anos como *product manager*, alinhando UX com códigos e processos ágeis. Atualmente é *head* de produto do Videolog e um dos principais palestrantes e colunistas sobre front-end no Brasil.

Obrigado, Bernard. A palavra é toda sua. Bem-vindo e sintase em casa.

Em 1960, um pesquisador americano filho de gregos chamado Nicholas Negroponte analisou um movimento da arquitetura pós-moderna, passando a aplicar a robótica para criar ambientes sustentáveis e adaptáveis às condições predefinidas ou desejáveis, seja ao corpo humano, às cores, à quantidade de pessoas e até mesmo às condições climáticas.

Pensar em desenvolver formas e objetos adaptáveis exige uma solução que combine uma aplicação tecnológica com uma excelente qualidade final do produto. Porém, para criar um projeto responsivo, é necessário avaliar tantas variáveis que exige uma gama de disciplinas envolvidas comprometidas a trabalhar nas mais diversas camadas do produto como arquitetura, engenharia, eletrônica, psicologia, química, entre outras, podendo criar estruturas otimizadas e sustentáveis.

Temos como exemplo o designer multifacetado Richard Buckminster Fuller. Bucky (<http://bfi.org/>), como era conhecido, foi um arquiteto renomado com mais de cinquenta doutoramentos *honoris causa*. O que mais chama a atenção é sua visão de edifícios desmontáveis, onde as instalações eram feitas com materiais sustentáveis, flexíveis e leves, de forma que pudessem ser transportados com facilidade por caminhões ou helicópteros, criando um projeto modularizado e de fácil adaptação (<http://kwz.me/8q>). Se por um minuto projetarmos essas aplicações para o meio digital, veremos que tudo faz sentido com a época em que vivemos e com o que construímos na internet, lindos edifícios digitais, focados em performance, com módulos reutilizáveis e adaptáveis ao usuário, isso é o que chamamos de Responsive Web Design.

O tão cunhado Responsive Web Design, ou RWD, veio recentemente, mais precisamente em 2010. Tudo começou com um post no blog *A List Apart*, onde Ethan Marcotte retrata a mudança de comportamento e crescimento de usuários mobile, sinalizando que isso já estava afetando (em 2010) empresas que queriam seus websites disponíveis para novos dispositivos. A felicidade de Ethan foi comparar esse movimento ao trabalho de arquitetura responsiva de Negroponte, em que temos de adaptar de forma inteligente o ambiente às variáveis de interação do usuário, nascido na arquitetura pós-moderna e atualmente apadrinhado na área de desenvolvimento web.

Projetar uma arquitetura exige mais disciplinas e conhecimento das variáveis externas do que a criação de um projeto web, além de edifícios serem projetos permanentes e duradouros, diferente de projetos web, que são transitórios e constantemente alterados. Assim, seguindo o modelo responsivo de criar uma arquitetura sensível à luminosidade, à temperatura, às cores, aos pesos e às pessoas, podemos utilizar o modelo responsivo para criar uma interface sensível ao dispositivo, ao tamanho e à orientação de tela, à capacidade de cores, à luminosidade do ambiente, ao comportamento dos olhos, aos leitores de tela e a muitas outras variáveis da relação homem-interface.

O homem é o principal transformador da interface, e a tecnologia respondeu com máquinas cada vez mais portáteis e poderosas, por exemplo, o celular, que não só nos mantém sempre conectados como está cada vez mais poderoso. Você sabia que o AGC, computador da Apollo 11 (primeira nave tripulada a pousar na Lua), tinha 2 Kb de memória RAM e processamento de 2.048 MHz, enquanto o iPhone 5s tem 1 Gb de RAM e processamento Dual-Core de 1.2 GHz? Então, podemos afirmar categoricamente que temos mais potência tecnológica no bolso que Neil Armstrong (primeiro homem a pisar na Lua (<http://kwz.me/8v>)). Esse novo homem conectado deixa de ser um mero consumidor de mídia, passando a ser produtor de conteúdo e se tornando bastante exigente para com a experiência que sites e aplicativos oferecem a ele no dia a dia (<http://kwz.me/8w>)).

Tudo parece muito bonito quando olhamos como usuário, mas essa beleza se transforma em um trabalho difícil e demorado quando olhamos como profissional. Para muitos, criar um projeto responsivo envolve pensar em todos os dispositivos e suas resoluções. A história da arquitetura nos mostra que criar um projeto adaptável é permitir uma adaptação orgânica e permanente, diferente da maneira que foi interpretado inicialmente o RWD. Em algum momento de nossa vida, quando fizemos a passagem dos layouts fluidos criados com tabelas para layouts diagramados com *divs*, começamos a sintetizar o "pixel perfect" em larguras e medidas predefinidas, gerando os primeiros layouts fixos para 800 x 600. Anos depois, com o aumento das resoluções de monitores, tivemos que repensar o tamanho dos sites, e novamente, em vez de criarmos uma solução "adaptativa", nos iludimos com a sensação de dever cumprido ao limitar a largura dos projetos para 1024 x 768. Assim sendo, um projeto responsivo orgânico deveria, em sua concepção natural, independentemente do navegador, atuar com fluidez, daí o nome de layout fluido, trabalhando com porcentagens e se adequando a qualquer resolução de forma permanente.

Infelizmente, a prática não é tão simples como parece na teoria, pois o comportamento de um layout fluido com *divs* não é tão simples como era utilizando tabelas, gerando muitos problemas não só de usabilidade como de estética, encorajando designers e desenvolvedores a deixar o fluido de lado, optando pelos layouts fixos. Esse problema se agravou no movimento responsivo, pois, além de calcular larguras fixas, temos que descobrir as resoluções dos novos devices, dar suporte aos devices anteriores e identificar o surgimento de novos devices. Essa prática chega a ser tão custosa e cansativa que gerou um movimento criado por Brad Frost contra o avanço exponencial de dispositivos e a incoerência de se contemplar cada dispositivo na hora de criar um projeto. O movimento chama-se "Device Fatigue", ou, se preferir, fadiga de dispositivo (<http://kwz.me/8x>), e pode ser visto no próprio blog de Brad Frost.

Recomendo um dia analisar algumas peças arquitetônicas responsivas. Observe como, embora tenham uma complexidade tecnológica, elas são simples e únicas em sua fachada. Esse pensamento deve ser mantido para RWD, priorizando interfaces leves e flexíveis, sem detalhes e efeitos desnecessários, focando em usuários que você ainda não conhece ou que ainda vão surgir, mas sabendo quais são os seus desejos, os seus interesses e as suas preocupações. Uma arquitetura adaptável não é projetada apenas para mulheres americanas com até 70 quilos e de estatura mediana, e dificilmente uma arquitetura faz um teste com todos os tipos de climas e pessoas da cidade antes de projetar. Esse deve ser o foco de um projeto web responsivo, centralizado na harmonia com qualquer tipo de ambiente, dispositivos e pessoas, garantindo a sobrevivência do seu projeto.

## 1.3 Princípios do design responsivo

### 1.3.1 Introdução ao RWD

Antes de qualquer coisa, é necessário que fique muito claro que design responsivo não diz respeito simplesmente e somente à adaptação do layout ao tamanho da tela.

Vai muito além disso, pois o conceito de design responsivo na sua forma ampla deve ser entendido como o design capaz de "responder" às características do dispositivo ao qual é servido. Responder, nesse contexto, tem o sentido de movimentar-se expandindo e contraindo.

Em outras palavras, o design responsivo ou layout responsivo expande e contrai com a finalidade de se acomodar de maneira usável e acessível à área onde é visualizado ou, mais genericamente, ao contexto onde é renderizado, seja um smartphone, um tablet, um desktop, um leitor de tela, um mecanismo de busca etc.

### 1.3.2 Ingredientes do RWD

Ethan, o pai do design responsivo para a web, usou o termo "ingredientes técnicos" para relacionar os três componentes, ou as três tecnologias fundamentais para desenvolvimento do design responsivo. São elas.

- grid fluído;
- imagens e mídia flexíveis;
- media queries.

Grid fluído é um termo que foi popularizado e incorporado definitivamente ao jargão de desenvolvimento graças ao aparecimento dos frameworks CSS. Antes do advento desses frameworks, o termo usado era o layout com seus componentes fluídos, que continua válido. Se você preferir, siga com ele.

Layout fluído é aquele cujas medidas CSS são definidas em unidades relativas, tais como porcentagens e `ems`, e não em medidas absolutas como `pixel` e `mm`. No capítulo 4, estudaremos com detalhes o ingrediente grid fluído do design responsivo.

Mídia e imagem flexíveis são aquelas capazes de contrair ou expandir suas dimensões ou ter suas dimensões alteradas por scripts em função do contexto onde são renderizadas. As mídias mais comuns presentes em layouts são: vídeos, áudio e widgets em geral. No capítulo 4, estudaremos com detalhes o ingrediente imagens, e no capítulo 5, as mídias flexíveis do design responsivo.

Media queries é uma tecnologia CSS que permite ao autor servir folhas de estilos baseadas em determinadas características do dispositivo ao qual o layout será servido. No capítulo 3, estudaremos com detalhes o ingrediente media queries do design responsivo.

### 1.3.3 Pilares do RWD

As tecnologias que constituem os pilares de sustentação do design responsivo são:

- HTML5
- JavaScript para compatibilidades
- CSS3

O uso da HTML5, no desenvolvimento web em geral, proporciona ao autor, entre outras, as seguintes vantagens:

- código notadamente semântico;
- código mais bem estruturado, facilitando sua manutenção e leitura;
- melhor acessibilidade;
- melhores benefícios para SEO.

O uso de polyfills e scripts JavaScript é indispensável para servir navegadores antigos, notadamente o Internet Explorer, cujas versões ainda em uso, por exemplo, simplesmente não reconhecem os novos elementos da HTML5.

As CSS3 com seus diferentes módulos criaram funcionalidades de estilização que, além do grande impacto visual que produzem, chegaram para facilitar as tarefas do autor, por exemplo, reduzindo drasticamente a marcação HTML necessária para se obter bordas arredondadas e sombras. Entre as inúmeras novas funcionalidades criadas, destacamos as propriedades `text-shadow`, `box-shadow`, `@font-face`, `RGBA`, `linear-gradient`, `opacity`, `transition`, `background`, `border-image`, `animation`, a fantástica `box-sizing`, que criou uma maneira muito mais intuitiva de renderizar o box-model tradicional das CSS, e também as regras `media queries`.

No capítulo 2, estudaremos com detalhes essas tecnologias que constituem os pilares do design responsivo.

### 1.3.4 Mitos do RWD

Apesar de ser uma metodologia recente, alguns mitos já se criaram em torno do design responsivo, e é preciso desmitificá-los.

Para desmitificar os dogmas do design responsivo, convidei a desenvolvedora Dani Guerrato, que escreveu o presente subitem deste livro.

Dani é uma atuante blogueira e divulgadora das técnicas de desenvolvimento web no Brasil.

Dani é sócia proprietária da agência PopUp Design, localizada em São Paulo, cuja proposta é elaborar soluções visuais criativas, diferentes e impactantes elaborando projetos de design gráfico e digital para o mundo todo.

Conheça a agência e sua sócia em <http://www.popupdesign.com.br> (<http://kwz.me/6S>).

Obrigado, Dani. A palavra é toda sua. Bem-vinda e sintase em casa.

Tentar generalizar como as pessoas utilizam seus aparelhos móveis com base em conceitos de fundamentação duvidosa pode muitas vezes levar ao engano. E quem sai prejudicado é o próprio usuário. Conheça os principais mitos sobre o uso de dispositivos móveis.

#### 1.3.4.1 Mito 1

Usuários utilizam dispositivos móveis apenas na rua (e a conexão é sempre lenta).

Entre todos os meus gadgets, o meu tablet é o favorito. Ele fica sempre por perto e nunca desliga. Se eu preciso de uma informação, é só estender o braço e ler. Muito mais rápido e prático do que andar até o escritório, ligar o computador, esperar o navegador abrir e realizar uma busca. Ou seja, o tablet é o meu dispositivo de navegação primária quando estou em casa. Mas não estou baseando esta informação apenas na forma como eu pessoalmente uso a internet.

Acreditar que um usuário de dispositivo móvel necessariamente tem uma conexão de internet pior, está com pressa ou só acessa a internet através do aparelho da rua é um erro. Pesquisas realizadas pelo Google aqui mesmo no Brasil em 2013 (<http://kwz.me/7c>) mostram que a maior parte dos usuários utiliza internet em aparelhos móveis prioritariamente em casa (96% dos entrevistados) ou no trabalho (87%), e só 78% das pessoas utilizam o aparelho na rua.

#### 1.3.4.2 Mito 2

Usuários de dispositivos móveis não precisam do conteúdo total.

Imagine que a sua pizzaria favorita abriu um serviço de delivery. Mas, ao olhar no menu, você percebe que eles só têm disponível para entrega as pizzas de mussarela e calabresa. Você já visitou o restaurante e sabe que eles oferecem

dezenas de outros sabores para os clientes presenciais. Mas, por algum motivo, eles acham que quem pede pizza por delivery não está interessado no cardápio completo. É provável que você fique bravo com o estabelecimento e prefira pedir comida em outro lugar. A mesma coisa acontece com os usuários de dispositivos móveis. Na verdade, 32% dos usuários preferem ver a versão para desktop a ver a versão móvel (<http://kwz.me/7y>). Isso acontece pois, muitas vezes, a versão mobile possui menos conteúdo que a versão desktop. Não é porque alguém tem uma tela menor que ele deve ser sentenciado a ver menos informações. É o design que deve ser modificado para criar a melhor experiência, não o conteúdo.

A média de vendas de carros no eBay mobile do Reino Unido é de um a cada 57 segundos. Foram mais de 70 milhões de libras em vendas só nos primeiros três meses de 2012. Você ainda acha que os usuários mobile não estão interessados em visualizar o conteúdo completo?

### 1.3.4.3 Mito 3

Usuários de dispositivos móveis preferem uma experiência mais simples.

Alguns desenvolvedores pensam que quem acessa um site via smartphone quer ver uma versão lite, como se complexidade fosse necessariamente uma coisa ruim. Não é porque você está em um dispositivo móvel que você quer ter uma experiência menor ou menos complexa do que em um computador de mesa. Isso não significa que você deva mostrar todas as informações espremidas na mesma tela. Complexidade é diferente de complicação. Usuários de dispositivos móveis farão tudo que os usuários de desktop podem desde que o conteúdo seja apresentado com usabilidade. Limitar as interações é o mesmo que decidir pelos usuários o que eles podem ou não fazer em seus próprios aparelhos. Liberdade de escolha é fundamental.

### 1.3.4.4 Mito 4

Existe uma “internet móvel”

Quem nunca se irritou ao encontrar um link interessante em um mecanismo de busca apenas para ser redirecionado para a página inicial da versão mobile? Isso acontece porque alguns desenvolvedores tratam as experiências de navegação em diferentes aparelhos como mundos separados, como se existissem duas webs – uma “normal” e uma “móvel”. Esta visão pode quebrar um galho

na hora de criar um modelinho de desenvolvimento, mas na real ela é muito rasa. A internet é uma só. Criar divisões artificiais de acordo com o formato do dispositivo só segrega o conteúdo e, conseqüentemente, os usuários. Não faz diferença se você é do time de versões mobile ou do design responsivo, o que importa é centralizar a navegação em um único endereço que possa idealmente ser sincronizado de acordo com o dispositivo. Utilizar uma mesma URL para o conteúdo desktop e mobile não é apenas uma boa prática recomendada pelo Google, é uma questão de bom senso!

#### 1.3.4.5 Conclusão

O lugar comum em termos de discussão entre desenvolvedores gira atualmente em torno de “design mobile *versus* design desktop”, mas por que não pensar nos dois como aliados em vez de antagonistas? Tablets, smartphones, desktops, televisores e games portáteis formam um verdadeiro ecossistema de telas. Em vez de sempre optar pela simplificação, pense em estratégias para tornar a experiência mais atrativa em cada tipo de mídia. Os dispositivos móveis modernos contam com features diferentes de computadores desktops: tela sensível ao toque, microfone, GPS, giroscópio, NFC etc. E ainda assim estamos perdendo a oportunidade de explorar estes features ao querer sempre diminuir o número de interações. Projetar experiências complementares e sincrônicas é a grande tendência para o futuro. Tanto o Safari quanto o Google Chrome já sincronizam abas e favoritos entre dispositivos. O artigo *Patterns for Multiscreen Strategies* aponta ao menos seis modelos de interação entre dispositivos: coesão, sincronização, compartilhamento da tela, migração de dispositivos, complementação e simultaneidade.

Com todos estes caminhos a explorar, por que se contentar com “menos é mais”? Às vezes, menos pode ser menos. Deixe seu usuário decidir. Um exemplo recente disto é o uso das hashtags. Popularizadas pelo Twitter, muitos usuários começaram a utilizar o recurso no Facebook, mesmo sem nenhuma função prática de busca ou registro de palavras-chave. Graças à popularização das hashtags, o Facebook implantou oficialmente o recurso. A lição que podemos aprender aqui é: não tente adivinhar o que usuários querem ou impor qual é o “jeito certo” de interagir em uma aplicação. Escute as solicitações. Assim você terá o feedback necessário para projetar interfaces que as pessoas realmente queiram utilizar.



### 1.3.5 Conceito mobile first

No dia 3 de novembro de 2009, Luke Wroblewski, figura 1.6, publicou uma matéria em <http://www.lukew.com/ff/entry.asp?933> (<http://kwz.me/6b>) cujo título é "Mobile First" defendendo e justificando a necessidade de se começar o projeto de um layout a partir da sua apresentação em dispositivos móveis. A matéria criou o termo mobile first, que foi rapidamente incorporado ao jargão do desenvolvimento para dispositivos móveis.

Luke começa a matéria afirmando o seguinte (em tradução verbosa):

Na maioria das vezes, a criação do layout para servir uma aplicação ou um site para dispositivos móveis só começa depois que o layout para desktop está pronto. Nessa matéria, apresento três razões para que o fluxo de criação do layout seja invertido, isto é, primeiro o layout para dispositivo móvel e a partir daí o layout para desktop.

As razões apontadas na matéria para se projetar pensando primeiro em layout para dispositivos móveis são as seguintes:

- Uma explosão da tecnologia mobile está em curso.
- Layout para dispositivo móvel implica em otimização de dados e ações a serem projetadas com descarte de elementos desnecessários.
- Dispositivos móveis estão incorporando a cada dia mais e mais funcionalidades nativas e capacidade de criação de conteúdos ricos não existentes na maioria dos navegadores para desktop.

Para informações detalhadas, consulte a matéria do Luke.



Figura 1.6 – Luke Wroblewski, o criador do conceito mobile first.

## 1.4 Web Design Adaptivo

O termo Adaptive Web Design (AWD) foi criado por Aaron Gustafson, autor do livro "Adaptive Web Design – Crafting Rich Experiences with Progressive Enhancement", no qual descreve o conceito e as técnicas de criação de layout que se adaptem às características e às capacidades do dispositivo do usuário.

Citamos esse tipo de design neste livro com a finalidade de informar ao leitor sobre sua existência e alertar que o termo não deve ser tomado como sinônimo de RWD, embora o propósito de ambas as tecnologias seja o de servir layouts para desktop e dispositivos móveis, fazendo com que sejam tecnologias paralelas e não divergentes.

A diferença entre ambas está antes de tudo nas premissas de desenvolvimento.

Segundo definição de Aaron Gustafson, "AWD diz respeito à criação de interfaces que se adaptam às capacidades do usuário, seja na sua forma, seja nas suas funções", e o autor segue dizendo "AWD é outro termo para 'progressive enhancement' – melhoria progressiva –, mas que considera o modo como as funcionalidades individuais daquelas capacidades interagem entre si para determinar o comportamento do todo, uma vez que leva em conta variados níveis de marcação HTML, CSS e JavaScript bem como suporte para tecnologias assistivas".

RWD, como já estudamos anteriormente neste capítulo, diz respeito à criação de interfaces baseadas em grids fluídos, imagens e mídia flexíveis e media queries.

A literatura discutindo o emprego e as vantagens e desvantagens de cada uma dessas duas tecnologias é vastíssima, e não é do escopo deste livro aprofundar esse assunto, pois, como dissemos no início deste tópico, o que aqui relatamos tem caráter meramente informativo.

O consenso é que ambas as tecnologias têm sua aplicação. Há casos em que AWD é preferível à RWD e vice-versa.

Um exemplo clássico de aplicação do AWD (na maioria dos casos) é sua conveniência para transformar um layout existente para desktop com a finalidade de servi-lo a dispositivos móveis. Nesses casos, pode ser que o emprego de AWD reduza drasticamente a necessidade de se alterar códigos e marcação, com aproveitamento total do que já existe.

Por outro lado, RWD (na maioria dos casos) é conveniente para se criar um layout novo, e aqui convém ressaltar que as boas práticas de desenvolvimento recomendadas pelo Google dão prioridade para o RWD, o que obviamente não invalida AWD, mas pesa a favor de RWD.

Nicholas Davidson criou uma interface interativa bem interessante, mostrando um layout de três colunas no qual o usuário escolhe a maneira como deseja que o layout seja apresentado na tela. As opções são layout estático, elástico, responsivo ou adaptivo.

O usuário escolhe um layout e redimensiona a janela do navegador para inspecionar o comportamento dele. Recomendo uma visita a <http://liquidaptive.com/>.

## 1.5 RWD segundo Horácio Pastor Soares

Para fechar o primeiro capítulo do livro, convidei o professor Horácio Pastor Soares, que escreveu o presente item deste livro.

Horácio Soares é especialista em experiência do usuário, usabilidade e acessibilidade Web, é fundador da Acesso Digital (<http://acessodigital.net/>) e da MESTR.ES (<http://mestr.es>), e consultor da Soyuz (<http://soyuz.com.br>).

Professor de experiência do usuário do MBA em Marketing Digital da FGV, tem 16 anos de experiência no desenvolvimento de soluções Web. Faz parte do Conselho Consultivo do Instituto Intranet Portal e do GT de Acessibilidade na Web do W3C Brasil. É analista de sistemas pela PUC (RJ) e mestrando em sistemas de informação pela Unirio.

Obrigado, Horácio. A palavra é toda sua. Bem-vindo e sinta-se em casa.

### 1.5.1 Objetivos semelhantes, experiências diferentes

A cada dia, as pessoas realizam mais e mais tarefas online e, quanto maior a experiência dos usuários no uso do computador e da internet, menor é a tolerância com problemas como a falta de performance, dificuldade de uso, falta de foco, inacessibilidade, entre tantas outras barreiras de uso, infelizmente muito comuns no Brasil.

Se experiências ruins são cada vez menos toleradas na web tradicional, em um contexto em que os acessos são picotados, a atenção dividida, a conexão cara e inconsistente, as telas menores, sem a precisão do mouse e o conforto e feedback dos grandes teclados, experiências ruins são definitivamente proibidas.

Em dispositivos móveis, se um site demorar um pouco mais para carregar do que o normal, se o objetivo não for claro o suficiente e seu funcionamento simples e intuitivo, os usuários irão embora.

### 11.5.2 Desktop x Mobile

Apesar de os objetivos dos usuários ao acessarem sites/sistemas nas versões desktop e mobile serem os mesmos, ou muito semelhantes, as experiências certamente são diferentes e devem ser consideradas durante a concepção de um projeto web.

Vejamos algumas diferenças (Tabela 1.1):

*Tabela 1.1 – Diferenças desktop – mobile*

<b>Desktop</b>	<b>Mobile</b>
Tela grande	Telas pequenas, com diferentes dimensões, mas com substancial perda de área com relação ao desktop
Teclado-padrão	Teclado pequeno e sem feedback tátil nas versões touch
Mouse	Dedo gordo (Fat Finger), sem cursor e feedback
Banda larga (baixo custo)	3G, 4G, limite de banda, custo alto e percentual considerável de usuários com acesso pré-pago
Energia abundante	Energia limitada, principalmente se os recursos como o GPS estiverem ativos
Rede consistente	Rede inconsistente
Alta capacidade de processamento	Processamento limitado
Espaço de sobra no HD	Limite de espaço de armazenamento
Uso em casa e/ou no trabalho	Uso em qualquer hora, em qualquer lugar
Ambiente calmo, confortável, seguro e controlado	Em diferentes contextos, normalmente realizando mais de uma tarefa ao mesmo tempo, sem controle de variáveis como tempo, movimento, iluminação, som, atenção etc

Rachel Hinman, em seu livro *The Mobile Frontier*, faz a seguinte comparação:

“PC experiences are like skuba diving; mobile experience are like snorkeling.”<sup>1</sup>

1 Experiências de PC são como um mergulho skuba; experiência móvel é como snorkeling.

A autora faz uso da metáfora do mergulho para diferenciar as experiências das pessoas ao utilizarem o desktop e o mobile. Em ambos os casos, o objetivo dos mergulhadores é apreciar peixes e plantas submarinas, as belezas naturais dos mares, entretanto o mergulho via desktop (Skuba Diving) é intenso, detalhado, com duração prolongada, em águas profundas, enquanto no mobile (Snorkeling) os mergulhadores ficam apenas na superfície do mar e realizam diversos mergulhos de rápida duração.

Ainda que as experiências sejam diferentes, os usuários esperam poder realizar as mesmas tarefas, consultar os mesmos conteúdos e, para eles, não deveria fazer diferença se o acesso é feito via desktop, tablet ou smartphone.

Por exemplo, ao acessar um site de uma companhia aérea, um cliente espera, entre outras coisas, pesquisar voos e preços, comprar, remarcar e cancelar passagens, realizar checkin e verificar os status dos voos. Ele pode precisar fazer isso de sua casa ou no trabalho, mas também enquanto espera uma consulta médica, na fila do cinema, na praia, durante o almoço, em um traslado no táxi ou simplesmente caminhando pela rua.

Contudo, em muitos sites, parte das tarefas e dos conteúdos só pode ser acessada na versão desktop ou "versão completa". Se existe um link para uma versão completa do site, será que os usuários não vão achar que a versão mobile é incompleta? Outro problema é quando a versão móvel herda toda a complexidade da versão para desktop, replicando todos os conteúdos e as funcionalidades, relevantes ou não. Experiências que já são ruins em um ambiente com tela grande tornam-se impossíveis quando acessadas em telas pequenas.

### 1.5.3 Limitações podem ser boas para o negócio

Como os projetos para desktop não sofrem as mesmas restrições do mobile, enquanto há espaços em branco, suas páginas recebem todos os tipos de conteúdo e elementos para atenderem às diferentes demandas departamentais e comerciais das empresas.

Mesmo que o mobile apresente "limitações" quando comparado ao desktop, as restrições impostas podem ser muito boas para o negócio e uma oportunidade para focar no que realmente é importante e prioritário, tanto para os clientes quanto para a empresa.

Restrições que podem nortear o desenvolvimento de sites/sistemas para mobile e impedir assim a criação de soluções lentas, confusas, complexas e com falta de foco. Mas, antes de eliminar as gorduras, ou seja, retirar da versão mobile o excesso de funcionalidades e conteúdos, é necessário identificar o que realmente é essencial para os usuários e prioridade para os negócios da empresa.

### **1.5.4 Mobile First**

Aproveitando as restrições do mundo mobile, um caminho cada vez mais utilizado no desenvolvimento de soluções web é a aplicação da técnica conhecida como “Mobile First”, criada pelo designer Luke Wroblewski. Ainda que a técnica seja descrita pelo designer em um livro com mesmo nome, a sua aplicação é simples.

Resumidamente, ao iniciar um novo projeto web, em vez de começar pela versão desktop, deve-se iniciar a concepção e modelagem pela versão móvel, para só então criar a versão para desktop.

De olho nas limitações do desenvolvimento de soluções móveis, ao conceber primeiro a versão mobile, aumenta-se o foco no que realmente é indispensável para os usuários, evitando assim a retirada de itens necessários ou a inclusão de elementos supérfluos a sua experiência.

Contudo, a aplicação da técnica, além de aumentar a qualidade dos sites para os dispositivos móveis, também pode melhorar a qualidade das versões para desktop. Isso porque, após a concepção e validação da solução móvel, se a equipe mantiver o mesmo foco e as mesmas prioridades, o resultado será uma solução desktop mais simples e com objetivos claros.

Mas como toda mudança de paradigma não é tarefa simples, para funcionar, é preciso ter o apoio de gestores do projeto, além da participação ativa dos envolvidos no projeto.

### **1.5.5 Pesquisa, concepção participativa, validação e mensuração**

Apenas com o uso da técnica mobile first não é possível garantir os resultados esperados. Durante a concepção do projeto, é preciso investigar e descobrir qual é o alvo que se pretende atingir, a sua missão, a razão pela qual o projeto será construído ou remodelado. Mas para isso é fundamental a identificação correta das necessidades dos usuários, assim como são essenciais os objetivos e as metas da empresa com o projeto.

É preciso investir tempo e dinheiro em pesquisa com os usuários, realizar testes de usabilidade, análise de dados históricos de uso, concepção e modelagem participativa, entre outros. Como errar faz parte do processo, é importante sempre obter a validação pela equipe e também com usuários nas diferentes fases do projeto. A validação faz parte integrante do processo e é a única forma de garantir que o projeto tenha qualidade, evitando retrabalhos, desgaste da imagem da empresa, insatisfação dos clientes, diminuição dos lucros etc.

Com tempo e a apuração dos primeiros resultados, as barreiras culturais com o novo paradigma se tornarão cada vez menores.

### 1.5.6 Design Responsivo e o mobile first

Independentemente do caminho ou dos caminhos mobile que serão definidos para o projeto, ou seja, se será em design responsivo, apps nativas ou híbridas ou um site otimizado para mobile e outro para desktop, a aplicação da técnica do mobile first é indicada para todas as soluções.

Entretanto, para o design responsivo, em que uma única solução é desenvolvida para atender a versões para TV, desktop, tablet, smartphone, consoles de jogos etc., iniciar o projeto pela versão mobile é uma condição *sine qua non* para a qualidade do projeto.