

MANUAL DE WEB DESIGN RESPONSIVO

Projete para Todos os Dispositivos



CHIEF
OF DESIGN

Sumário

Apresentação.....	3
Como surgiu o Design Responsivo (RWD)?	7
Elementos do Design Responsivo.....	9
Resoluções Comuns	12
Conteúdo “First”	13
Wireframe para começar	14
Grids para organizar e adaptar.....	16
Prefira Ícones, CSS e fontes a imagens.....	26
Otimize as suas imagens	27
Dispositivos com Tela de Retina.....	28
Use medidas relativas ao invés de absolutas	29
Utilize somente o necessário	30
Art Direction.....	31
Use mockups para apresentar para o cliente.....	32
Design Responsivo x Design Fluido.....	33
Design Responsivo (RWD) ou Design Adaptativo (AWD)	34
Considerações Finais.....	37
Referências.....	38



Apresentação

Me lembro que comecei a desenvolver as minhas primeiras páginas web por volta de 2008/2009. Eu peguei uma época boa, pois os padrões web estavam se fortificando cada vez mais e os sites feitos com tabelas, um jeito de se estruturar sites através de tabelas html, já estavam em decadência.

Nessa época já se começava a se pensar sobre web design responsivo (apesar que ainda tive que lidar com as horríveis tabelas), mas ainda era muito pouco e esse termo ainda nem existia. Normalmente os web designers criavam no Photoshop, no máximo, uma versão desktop e outra para mobile (por vezes eles eram até bem diferentes.) E quando isso ia para o código a mesma coisa acontecia: criava-se um site para desktop e outro para mobile e não raro essa versão mobile era em tabelas.

Apesar de naquela época não se levar tão a sério o conceito que temos hoje de design responsivo a preocupação de se criar layouts que funcionassem bem e com mesma qualidade em vários dispositivos já era um tanto “real” e crescia rapidamente, praticamente na mesma medida em que se crescia o acesso das pessoas aos smartphones e tablets.

Hoje em dia é impossível ao se criar uma interface web não se pensar na responsividade, ou seja, na adaptação do site para várias resoluções, quiçá todas que se pode alcançar.

Este e-book vem para te ajudar nessa questão fundamental na criação de sites que se chama Design Responsivo ou de forma abreviada RWD.

O que você encontrará neste e-book?

A ideia deste e-book é te ajudar a projetar um site responsivo mesmo antes de implementá-lo no código. Os códigos HTML, CSS e Javascripts são essenciais



para o funcionamento do Design Responsivo, entretanto neste e-book não iremos a fundo nessa questão, pois nós do Chief já temos um e-book Fluência em HTML e CSS ([você pode adquiri-lo aqui](#)) onde introduzimos o assunto.

E também porque este e-book ficaria muito longo e fugiria do principal objetivo dele: ajudar você web designer (ou criativo que trabalha com web) a pensar e projetar um site responsivo independente do seu grau de conhecimento em códigos. Então a ideia é te ajudar a começar a criar e projetar um design responsivo já na parte da elaboração do Design.

Muitos web designers criam um site desktop e depois “ajeitam” o site para ser responsivo. Nós não queremos isso. Queremos que pense no funcionamento, na usabilidade, na experiência do usuário desde a criação do site no Photoshop (ou no seu software favorito).

Quando comecei a criar para web criávamos um layout para 800x600, depois para 1024x768...

No momento atual não devemos pensar em qual resolução e sim na melhor experiência que podemos proporcionar aos usuários independentemente do tamanho da tela e que o design seja capaz de atender as características do dispositivo ao qual está sendo aplicado.

Design se trata somente daquilo que se vê, mas também como funciona como um todo.

Então venha comigo e preste atenção nas orientações, dicas e macetes deste e-book. Bele?

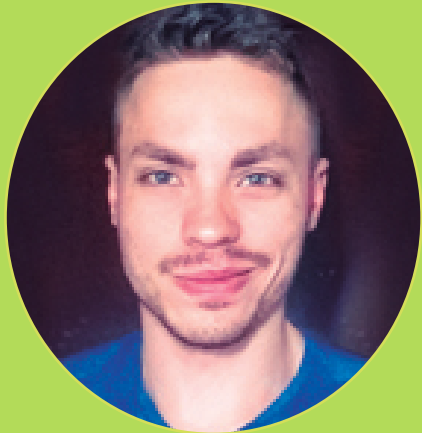
Siga-me os bons! :D



Esta obra está licenciada com uma Licença Creative Commons Atribuição-NãoComercial-Compartilhalqual 4.0 Internacional.



Sobre o Autor



David Arty é o fundador do Blog Chief of Design.

É natural de São Paulo, Brasil.

Graduado em Design Gráfico e formado em Técnico em Multimídia.

Trabalha como Web Designer desde 2009.

Autodidata por natureza, trabalhou em agências, empresas de TI, stúdios, entre outros. Atua também como freelancer e em algumas temporadas ministra treinamentos de Web Design.

Ama comunicação e criação. Encontrou no Design e no Desenvolvimento Front-End formas de unir e atuar com essas duas coisas.

Por isso é um entusiasta por tudo que envolve design e web, e tenta transformar ideias loucas e complexas em peças simples, atrativas, bonitas e funcionais.

Sobre o Blog



O Chief of Design é um blog que fala sobre Web Design, Front-End, Design Gráfico, entre outras coisas.

O intuito do blog é contribuir, principalmente, com quem está iniciando na área e que está buscando por aprendizado e respostas para suas dúvidas em relação as essas áreas.

O blog segue uma linha instrucional, ou seja, o objetivo é compartilhar conhecimento e servir de referência para ajudar profissionais, principalmente os iniciantes, da área criativa a produzirem, criarem, e quem sabe, por que não, e se tornarem "Chief" nesse segmento.

Para saber mais acesse:

www.chiefofdesign.com.br

Como surgiu o Design Responsivo (RWD)?

A internet surgiu em 1990 com o cientista Tim Berners-Lee, inventor da web e fundador do W3C (organização sem fins lucrativos que tem a missão de orientar os desenvolvedores para o uso de boas práticas para que a web seja acessível para todos de forma igual). Naquela época os sites eram todos muito simples em termos visuais e também ainda não tínhamos internet móvel, smartphones, tablets e etc.

Tudo mudou muito rápido. A partir dos anos 2000, com a chegada dos diferentes dispositivos e das diferentes possibilidades de acesso a internet, a necessidade de se criar sites que garantissem uma boa experiência, independente da resolução, foi crescendo e a solução ideal (pelo menos até agora) surgiu no dia 25 de maio de 2010 com o desenvolvedor Ethan Marcotte.



Ethan escreveu um artigo no famoso site Alispart ([veja aqui](#)) que mudaria completamente a forma de desenvolvimento de layouts para a web. O artigo abordava a mudança na forma de uso da web com o crescimento, que só aumentava, de usuários mobile e como isso afetava (em 2010) as empresas que necessitavam que seus sites tivessem uma boa performance e experiência também nos novos dispositivos.

A grande sacada do artigo foi a comparação com um movimento da arquitetura pós-moderno, tendo com um dos principais nomes o arquiteto Nicholas Negroponte, que trabalhava para criar ambientes adaptáveis e sustentáveis as condições já existentes.

Com isso Ethan mostrou que era preciso criar um layout adaptável, inteligente e que funcionasse independente das resoluções e mídias usadas.

Esse artigo mudou a forma de pensar no desenvolvimento de sites em todas as camadas: desde a criação a até o desenvolvimento.



Hoje em dia o RWD é uma realidade e criar uma interface para web sem se pensar na responsividade é cometer um erro crasso.

Mais do que criar um layout responsivo o RWD trouxe a importância de se pensar no visitante garantindo que seu site seja acessível ao maior número de dispositivos e com uma navegação agradável também nos dispositivos menores.

Elementos do Design Responsivo

O Design Responsivo tem alguns elementos fundamentais. Vários desses elementos fazem parte dos códigos Front-End. Não iremos nos aprofundar muito nesses elementos e focaremos mais na criação do layout, entretanto irei apresentá-los para você, até porque você precisa ter noção de que tudo funciona como uma engrenagem e que cada "pecinha" é importante para o funcionamento dessa máquina.

Então mesmo que você não desenvolva os códigos da parte Front-End é importante estar alinhado com o desenvolvedor que irá fazê-lo, pois HTML, CSS e JS são algumas dessas peças fundamentais do RWD.

Então as tecnologias essenciais para o Design Responsivo, são:

- Media queries e CSS3
- HTML5
- Javascript
- Mídias adaptáveis
- Grid Fluido

Media queries: é uma tecnologia CSS (Estilo em Cascata (CSS) que é a linguagem responsável pela formatação do estilo, do design do site) que possibilita criar folhas de estilos baseadas na largura da tela do dispositivo através do que chamamos de "breakpoints".

Portanto podemos criar um estilo para quando o site for renderizado numa resolução de 1200px (pixels), outro para uma resolução de 760px, 480px e assim vai.... Não existem regras específicas para isso. Existem sim alguns tamanhos mais comuns, mas quem realmente deve determinar esses "breakpoints" é o conteúdo do site.



Além das medias queries o CSS3 como um todo e todas as suas possibilidades é um dos pilares do RWD.

HTML: é a tecnologia que usamos para criar sites. Ele é responsável pela estruturação do site. HTML5, a versão mais recente do HTML, é fundamental para estruturação de um site responsivo.

Javascript: é uma linguagem de programação. No RWD ele possibilita, através de [“polyfills”](#) e scripts, que os navegadores mais antigos reconheçam os novos elementos do HTML5 tornando assim o design responsivo mais abrangente e atingindo até navegadores antigos.

Caso você não tenha conhecimento nessa tecnologia sugiro que adquira o [e-book solidário em Fluência em HTML e CSS](#).

Mídias adaptáveis: são imagens, vídeos, entre outros, com a capacidade de se expandir ou contrair de acordo com a resolução do dispositivo. Isso pode ser feito através de scripts e/ou sentenças CSS.

As mídias precisam ser fluídas para evitar que seja muito grande em resoluções pequenas ou muito pequena em resoluções grandes.

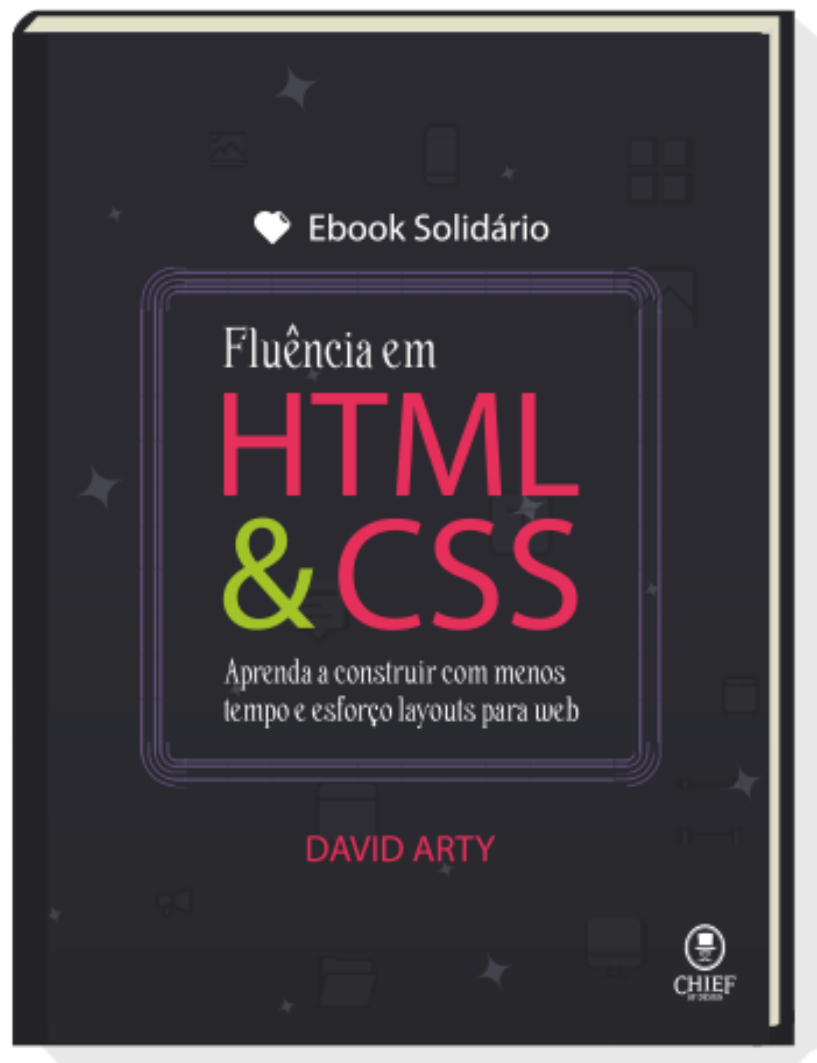
Grid Fluído : é uma técnica de desenvolvimento que utiliza medidas relativas para criar um layout de site de maneira “fluída”, ou seja, que ele possa estender ou se contrair de acordo com o tamanho da tela do dispositivo acessado.

Existem frameworks CSS que nos ajudam a criar um Grid Fluído na prática, visualizando direto no navegador, mas usaremos tal conceito para pensarmos no passo anterior a implementação do código, ou seja, a parte da criação.



Para o Web Designer responsável pelo design do projeto web é importantíssimo que ele conheça e use o conceito de Grid Fluído na hora da criação do layout no photoshop (ou no seu software favorito).

Focaremos nesse conceito e veremos a seguir como usá-lo para criar um layout responsivo. E caso você tenha interesse nos outros tópicos citados o primeiro passo que você pode dar é adquirir o [E-book de HTML e CSS do Chief of Design neste link!](#)



Resoluções Comuns

Com as inúmeras resoluções é impossível atender a todas com total perfeição, porém é importante saber dos tamanhos mais comuns “somente” para termos uma base para construirmos o layout no software gráfico.

1200 pixels de largura –
Desktops com monitores widescreen.

1024 pixels de largura –
Tablets em formato paisagem
e monitores antigos.

768 pixels de largura –
Tablets em formato retrato.

600 pixels de largura –
Tablets pequenos

480 pixels de largura –
Smartphones em formato paisagem.

320 pixels de largura –
Smartphones em formato retrato

Pensar em “breakpoints”, ou pontos de quebra, já não é tão eficaz devido as inúmeras resoluções. Portanto devemos fazer um site de qualidade que atenda resoluções entre 1200px e 300px.

A seguir veremos o porquê não devemos nos prender aos breakpoints.



Conteúdo “First”

Um conceito usado no desenvolvimento de sites responsivos é o “mobile first”, criado em 2009 pelo desenvolvedor Luke Wroblewski no site lukew ([confira aqui](#)). Este conceito defende que devemos começar o desenvolvimento de um site primeiro por sua versão mobile. Indo do menor para o maior (do mobile até desktop). Com isso se economiza e reaproveita códigos evitando coisas desnecessárias e refações durante o projeto.

Focando somente na concepção de um layout, podemos fazer uma analogia e usarmos o Conteúdo First.

Portanto é o conteúdo que vai determinar qual é a melhor estrutura para o layout. Ter definido o conteúdo que o site vai apresentar é essencial. E quando falo conteúdo não digo necessariamente o “conteúdo real”, ou seja, aquele texto bonito e bem escrito que fala do produto, mas sim o objetivo do site, do tipo de conteúdo, quais os tipos de mídias que o site terá, ter uma estimativa do tamanho do texto corrido de cara área e etc.

Para saber dessas informações é importante estar alinhado com quem fornece as informações do projeto e, claro, ter um briefing bem estruturado ([caso tenha interesse em saber mais sobre briefing leia esse artigo](#)).

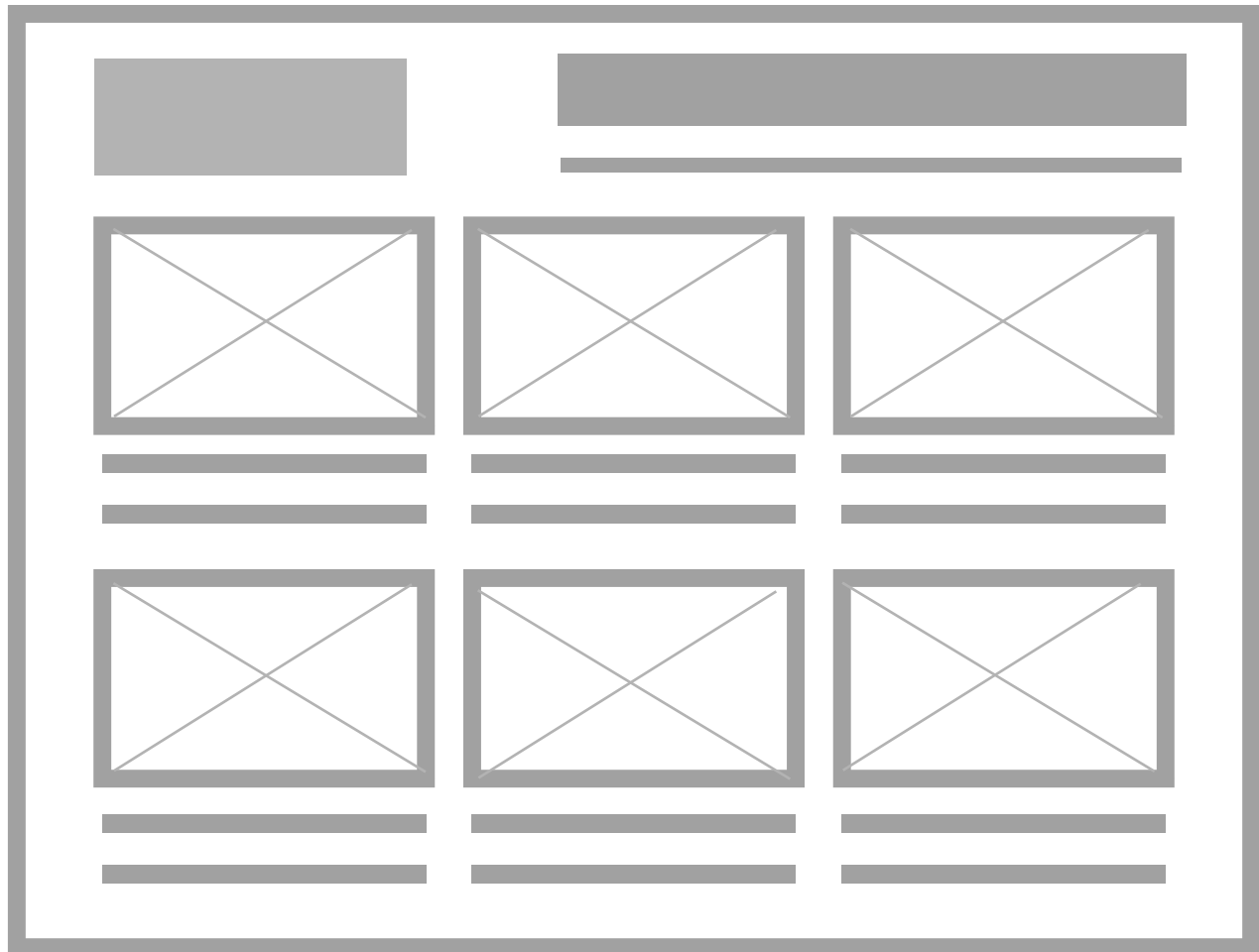
Com o conteúdo em mãos, podemos pensar em como organizá-lo e apresentá-lo da melhor forma possível nas mais várias resoluções. E será o conteúdo a estrela do layout. Faremos o design responsivo para que ele brilhe.

Então vamos começar a estruturar o site responsivo pelo wireframe.



Wireframe para começar

Wireframe é um esboço: uma simulação de como a interface deverá ser. Nele nós prevemos como o conteúdo será exposto de forma organizada.



Podemos construir um wireframe no papel, no computador ou nos dois.

A ideia é organizar o conteúdo para aplicarmos o design depois. Não precisa ser estiloso e nem bonito.



Caso queira usar um template específico para RWD você pode encontrar alguns na internet, como este por exemplo: <http://sneakpeekit.com/>

E caso queira saber mais sobre como criar um Wireframe, você pode conferir [esse artigo sobre criação de wireframes \(clique aqui\)](#).

No primeiro momento, na hora da criação do layout, não temos como prever com exatidão todas as resoluções. Conforme o projeto avança e você vai colocando o site no código HTML (ou alguém da sua equipe) é que você poderá ver onde o conteúdo “quebra” e aplicar as soluções necessárias.

Então, para começo e para que você entenda mais facilmente, iremos abordar somente três tipos básicos: desktops, tablets e smartphones.

Vamos nessa! :D

Grids para organizar e adaptar



Vamos utilizar o que fizemos no wireframe só que agora de uma forma estruturada e organizada, através dos grids.

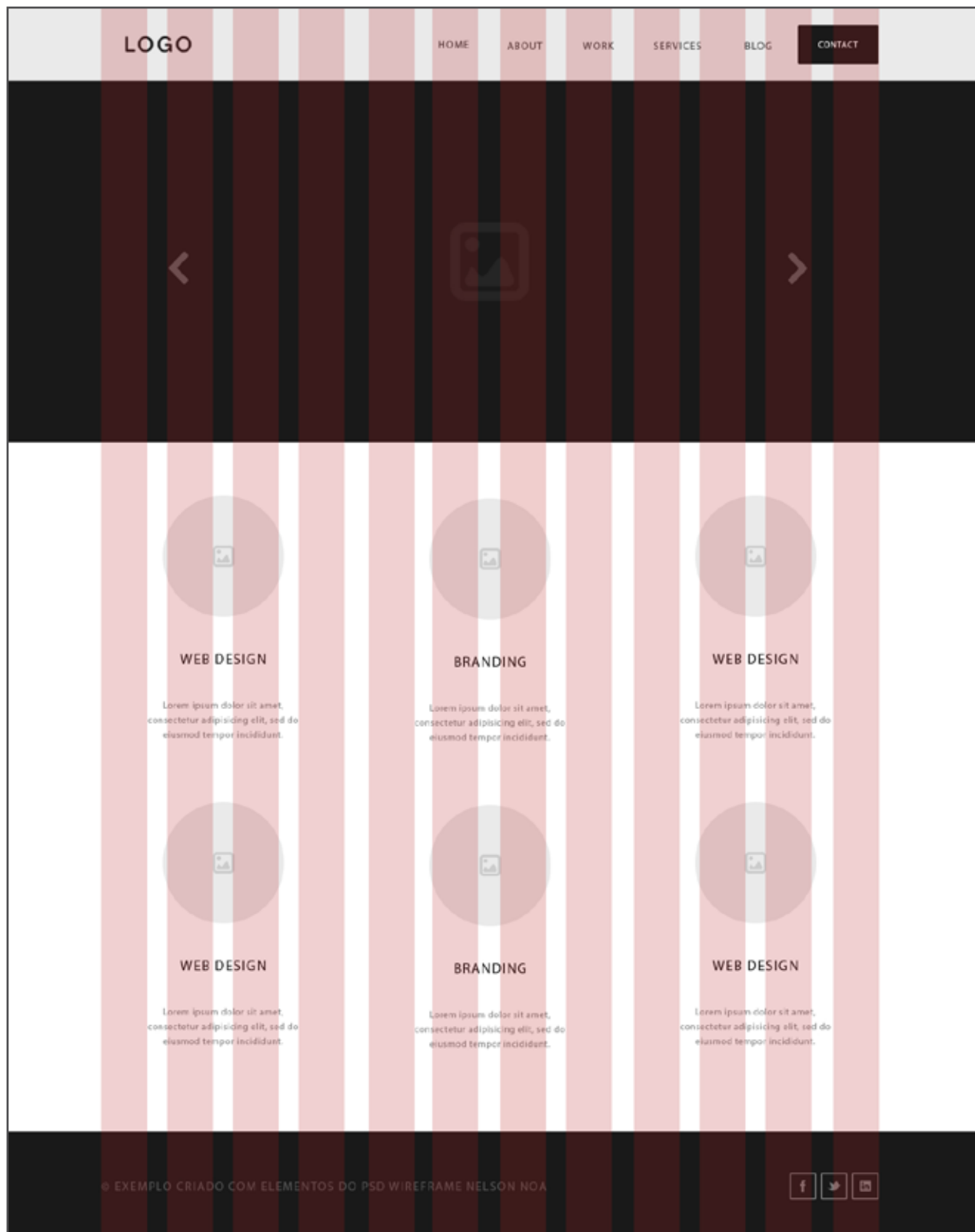
Grids são um conjunto de linhas verticais e horizontais que nos ajudam a organizar os elementos de um layout dentro do espaço determinado. Caso queira saber mais sobre grids [você pode ler este artigo \(clique aqui\)](#).

Por muito tempo usamos o grid 960. Mas isso ficou no passado...

Hoje em dia existem vários templates de Grids na web melhores que o 960. Também temos a ferramenta guideguide para Photoshop que facilita bastante a criação de grids.

Então primeiro você deve determinar o número de colunas do seu grid. Normalmente para interfaces web usa-se entre 12 a 24 colunas em versões desktop, mas não existe uma regra específica.

Começaremos fazendo primeiro a versão desktop. No exemplo a seguir usaremos 12 colunas. Veja:

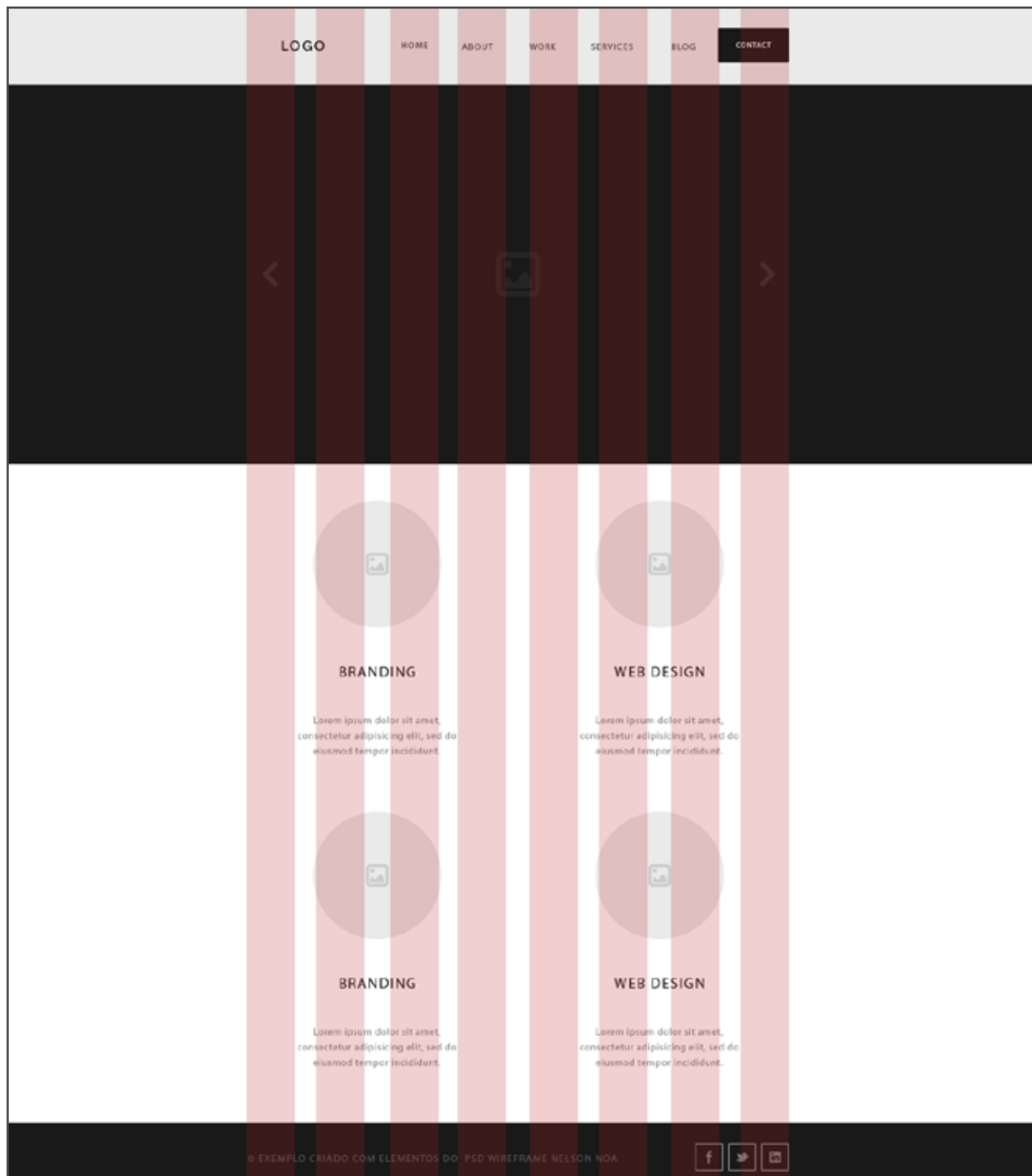


A técnica para adaptar a outras versões é bem simples, basta diminuir o número de colunas no grid. Logo, no nosso exemplo, se na versão maior (desktop) temos 12 colunas, na versão para tablet teremos 8 colunas e para smartphone 4 colunas.

Essa técnica de subtrair 4 colunas não é algo determinado, pois quem determina isso é o conteúdo do seu layout e a forma como você quer apresentá-lo. O conteúdo deve-se adaptar a estrutura menor, mas sem perder a identidade, sem excluir conteúdos e proporcionando uma boa experiência.

Então no tablet reorganizaremos o conteúdo passando de 3 conteúdos por fileira para 2 conteúdos por fileira. Além disso outros ajustes nos espaçamentos e nos tamanhos dos elementos do site, como por exemplo, logo, menu, imagens, devem ser realizados para garantir tanto a harmonia do layout quanto o funcionamento e também não prejudicar a área de toque no dispositivo “touch screens”. Veja a seguir:





* No exemplo acima temos apenas 4 itens na parte de serviços, pois a imagem ficaria muito grande e prejudicaria no visual do ebook. O template usado é apenas para fins didáticos. Caso fosse um projeto real todos os elementos deveriam estar presentes.

Por fim a versão mobile que terá apenas 4 colunas e que na prática o usuário verá apenas uma. Para mobile a atenção para a área de toque de ser redobrada.

O ideal é que o visitante consiga navegar sempre precisar dar “zoom” no site. Por isso cuidado com links pequenos e/ ou muitos próximos, pois assim eles serão difíceis de selecionar através do toque.

E um exemplo disso é o menu!

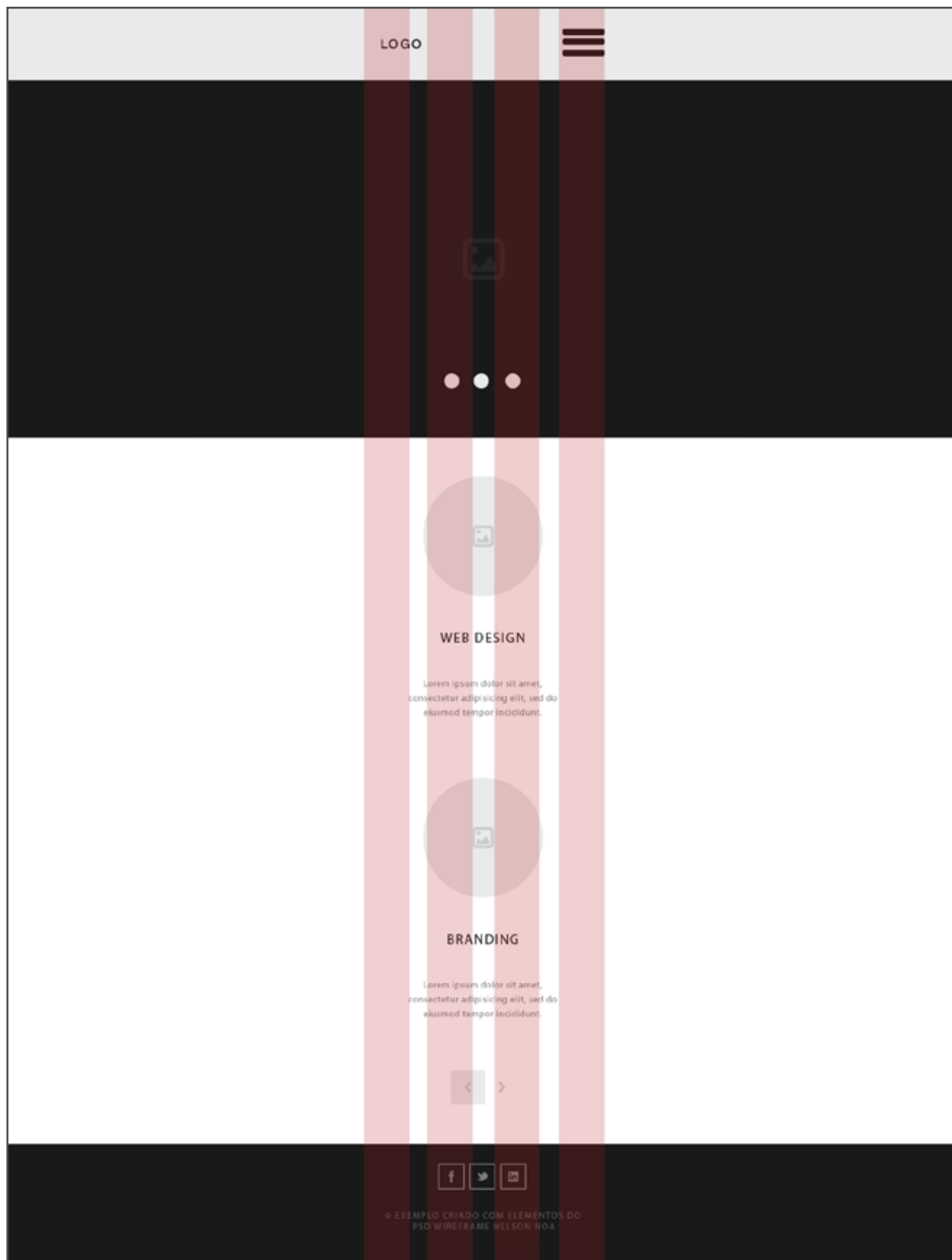
O menu como estava na versão desktop ficaria muito pequeno na versão mobile, então opta-se, comumente, em colapsar os elementos em um menu “drop-down”. No caso desse site usaremos o menu com estilo conhecido como hamburguer! (e que você já deve ter visto ao navegar aí no seu celular).

E também os elementos em destaque ficam um abaixo do outro. Neste exemplo usei uma navegação em “slide” para os destaques e deixei visível apenas dois deles.

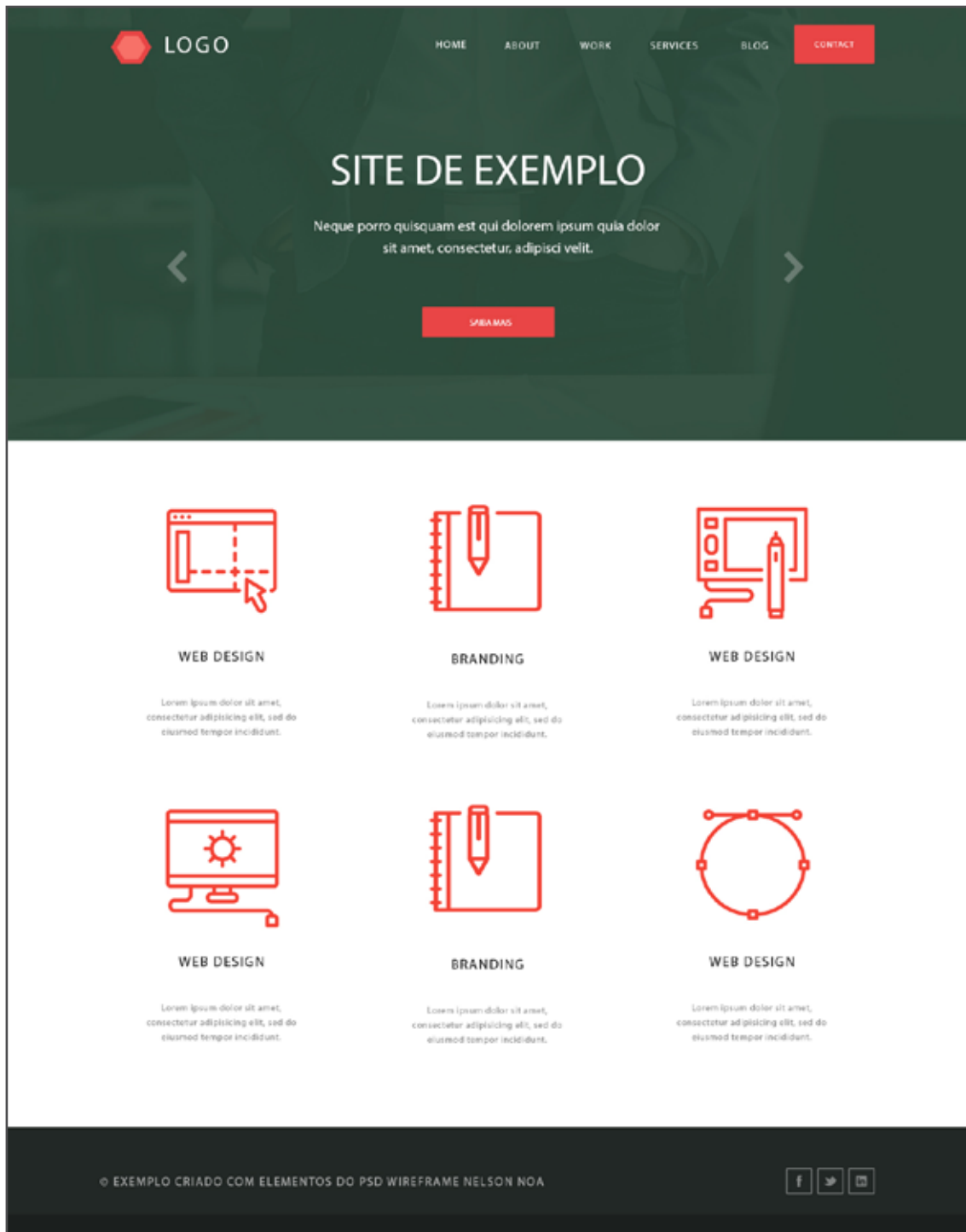
Como são apenas seis destaques isso foi uma opção e eu poderia deixar os seis expostos que provavelmente não iriam prejudicar a usabilidade, porém caso o site tivessem muitos destaques (uns 20, por exemplo) aí a opção por um slide em que a pessoa ao passar o dedo para o lado muda de destaque certamente seria essencial.

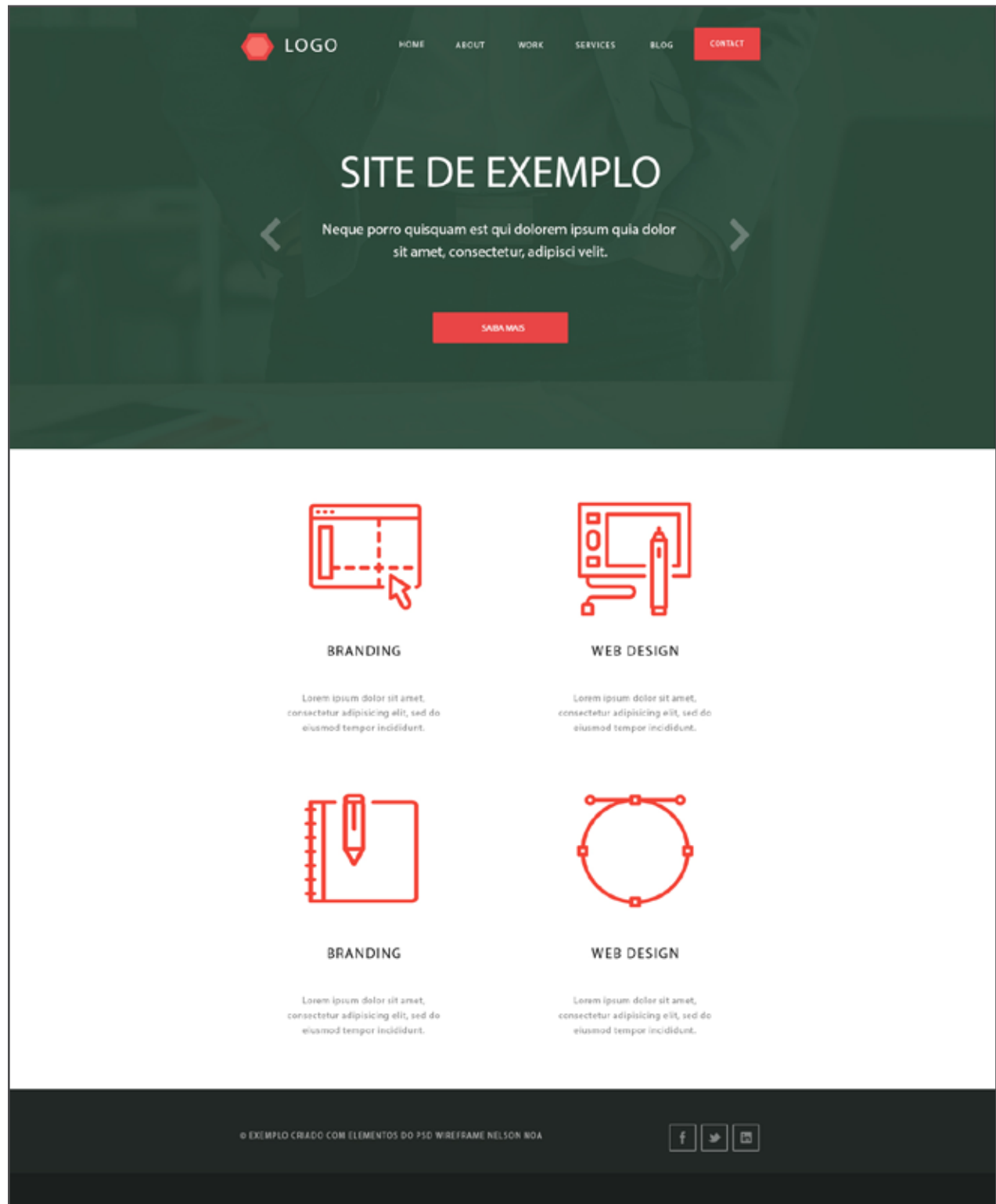
Veja:

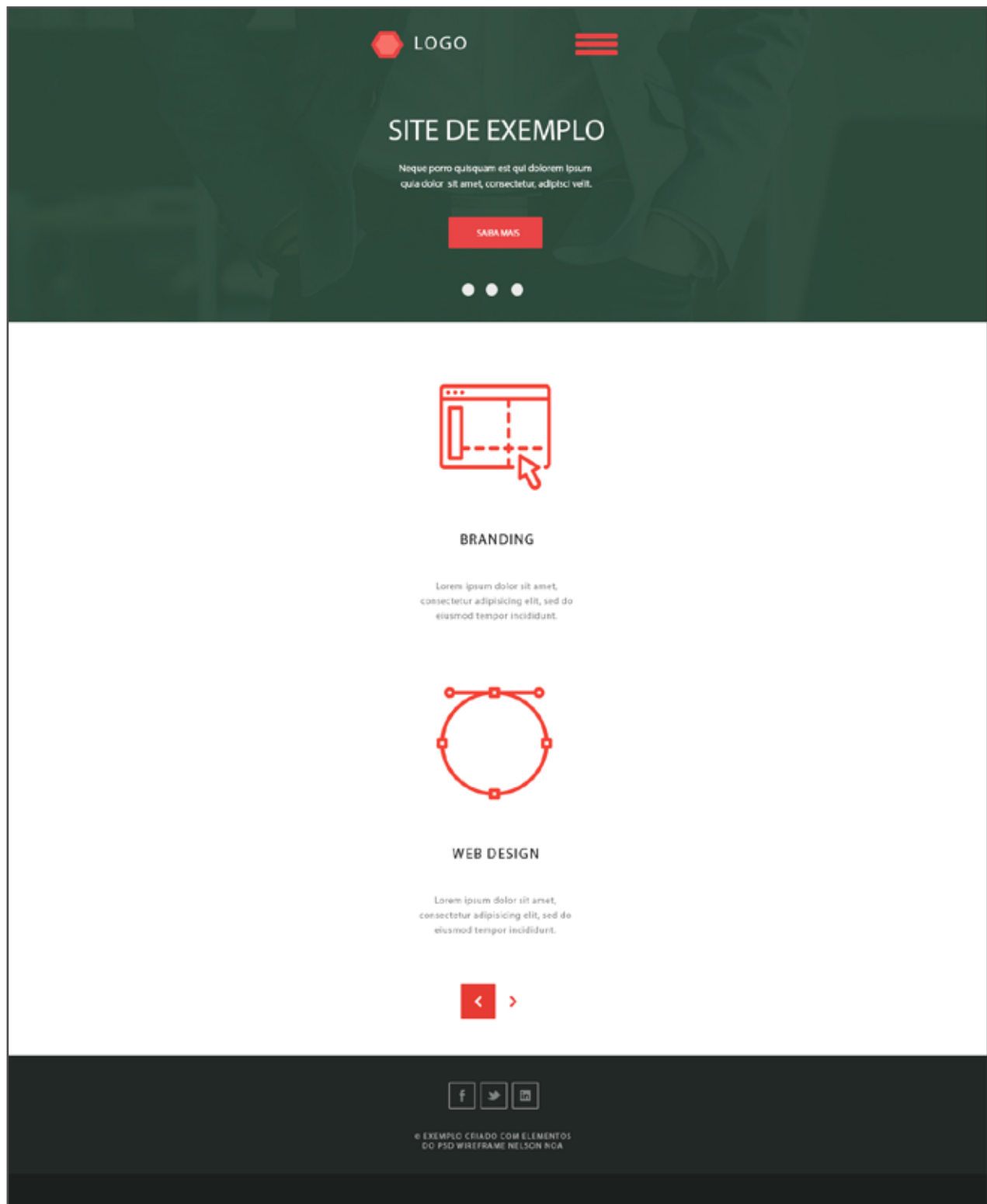




Agora depois de estruturado, aplica-se ao layout elementos estéticos formais. Veja o resultado:







Veja a seguir uma simulação de como ficaria o layout na prática.



Após você criar o design do seu site seguindo as técnicas e conceitos do RWD ele estará pronto para ser implementado no código. Neste e-book não iremos a fundo neste tópico, porém seguem algumas dicas importantes.

Prefira Ícones, CSS e fontes a imagens

Prefira trabalhar com gráficos vetoriais ao invés de imagens.
Use SVG, Icon Fonts e efeitos CSS3 (sombras, gradientes, cores, etc).

Tudo isso ajuda na manutenção do site e funciona muito bem em qualquer tipo de resolução. E também ajuda na performance do site deixando-o mais leve.

Por mais que você faça apenas o design do site, oriente e dê suporte a pessoa que estiver fazendo os códigos do site para utilizar tais técnicas.



Otimize as suas imagens

Fique atento as imagens do seu site. Lembre-se que por mais que você adapte o tamanho de uma imagem ao dispositivo ela sempre terá o tamanho inicial. Por exemplo:

Se você utilizar uma imagem com 300 kb no seu site, por mais que você use técnicas para redimensioná-la isso não irá diminuir o peso dela. Ela terá sempre 300kb independente do modo que ela está sendo apresentada.

Por isso, otimize as imagens do seu site. Você pode fazer isso de várias maneiras e existem vários serviços online que diminuem o peso das imagens sem perda de qualidade como:

- [Kraken.io](https://kraken.io)
- [TinyPNG](https://tinypng.com)
- [Jpgemini](https://jpgemini.com)
- [Compressor.io](https://compressor.io)
- [JpegOptimizer](https://jpegoptimizer.com)



Dispositivos com Tela de Retina

Tela Retina ou Retina Display é um termo criado pela Apple para nomear as suas telas de alta resolução de cristal líquido. Essas telas de retina tem uma densidade de pixels DPI (quantidade de pixels por polegada) maior do que os monitores comuns, logo as imagens ficam muito mais nítidas.

Então para telas assim é ideal criar imagens com o dobro de tamanho.
Por exemplo:



Se você vai criar uma imagem para um dispositivo de retina com o tamanho 320 X 240px, a imagem deve ser feita com 640 X 48px.

O padrão para nomear essas imagens para dispositivos de retina é o “@2X”, que é um padrão que a própria Apple criou para nomear as imagens de alta resolução.

Portanto, o nome do arquivo ficaria assim: minha-imagem@2x.jpg.

E você deve estar se perguntando?

“Mas como vou determinar que o dispositivo com tela de retina use a imagem @2x?”

Bom... Isso se consegue através de CSS, javascripts ou funções php, por exemplo.

Cabe a pessoa que estiver cuidando dessa parte pesquisar e encontrar o melhor caminho para a sua situação.

Use medidas relativas ao invés de absolutas

Mesmo que não seja você o responsável por fazer a parte do código é importante saber que para o Design Responsivo é sempre melhor usar unidades relativas ao invés de unidades absolutas.

Unidades absolutas (cm, mm, in, pt e pc) são medidas expressas que não dependem de valores de referência. Então, por exemplo, um box com uma largura de 800cm terá 800cm em todas as versões, desde desktop até mobile, independentemente do tamanho da tela.



Unidades relativas (% , em, ex, rem, ch, etc) são medidas que se baseiam em um valor de referência definido anteriormente. Portanto essas unidades para o RWD são mais úteis, porque podem mudar de acordo com o tamanho da tela se adaptando ao contexto em que estão inseridas.

A unidade px (pixel) pode ser considerada como híbrida, pois apesar de ser fixa (se você colocar 50px sempre será 50px) ela é calculada com base na resolução da tela onde o documento é visualizado, logo poderá variar de tela para tela.

No RWD é conveniente sempre preferir medidas relativas. Logo, prefira larguras em porcentagem à larguras fixas com pixels, prefira tamanhos de fontes com “unidades em” ao invés de “pixels” ou “pts”.

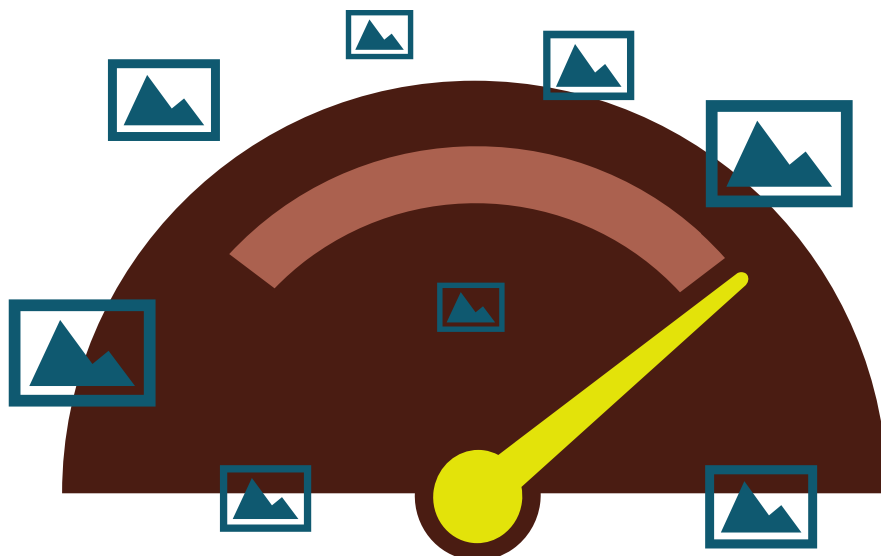
Utilize somente o necessário

Quando falamos de imagens no RWD devemos pensar em dois tipos: imagens de conteúdo e imagens decorativas.

Muitas das imagens que usamos nos websites tem apenas a função decorativa. Esse tipo de imagem caso seja retirada do site não vai interferir no entendimento do conteúdo. Elas podem ajudar a deixar o site mais bonito, etc, para versões desktop, porém para resoluções menores talvez elas não sejam fundamentais. E ainda podem prejudicar a performance do site.

Quanto mais imagens tem um site mais tempo ele leva para carregar. Na versão desktop, com uma conexão de internet estável, talvez isso não seja problema, porém em um aparelho mobile com menos memória de processamento e uma conexão que nem sempre é estável, esse tipo de imagem pode atrapalhar a velocidade de carregamento do site.

Nessas situações é preciso avaliar se essas imagens realmente necessitam aparecer em resoluções menores. Numa resolução menor o espaço que temos para trabalhar é reduzido e devemos aproveitá-lo ao máximo com o que realmente é importante para o visitante ver.



Art Direction

Essa técnica consiste cortar partes superficiais da imagem e focar no tema principal. Com isso pode-se reduzir o peso e tamanho da imagem sem perder o significado.

Então, cria-se versões da imagem que serão aplicadas de acordo com uma determinada faixa de resolução. Sendo assim quanto maior a resolução, mais completa é a imagem com menos foco no tema central, quanto menor é a resolução maior o foco no tema principal. Veja o exemplo a seguir:



Note que quanto maior é a tela mais detalhes do fundo aparecem, quanto menor é a tela menos detalhes do fundo e maior foco na figura principal da foto, o skatista.

Caso a imagem tenha sido otimizada e comprimida, em várias situações, o peso fica pequeno o bastante e uma troca de imagens não se faz necessário. Por outro lado, caso a imagem seja pesada para versões de telas maiores, você pode fazer verificações das telas dos dispositivos através de códigos (CSS, js, php, etc) e com isso criar imagens para cada tipo de resolução.

Use mockups para apresentar para o cliente

Mockup é uma representação de forma mais realista, ou seja, como o projeto ficará depois de pronto.

Na hora de apresentar um design responsivo para seu cliente use mockups, pois assim ele terá uma noção maior de como será o projeto depois de pronto.



Design Responsivo x Design Fluido

Muita gente se confunde ao ver um site com design fluído e logo pensa quem aquele mesmo é um site responsivo, porém são coisas diferentes.

O Design Responsivo pensa muito mais na usabilidade e na experiência do usuário. É muito mais que uma técnica e sim um conceito. Ele adapta a experiência de navegação de acordo com o dispositivo do usuário, com isso procura-se atender as necessidades dos visitantes de forma concisa. A estrutura pode não ser necessariamente a mesma em todos os dispositivos.

Já o Design Fluído somente aumenta ou diminui de tamanho de acordo com a área de visualização do dispositivo. É como se fosse uma sanfona, estica e contrai, sem pensar na usabilidade ou como o usuário daquele dispositivo acessa a interface.

Veja o exemplo a seguir e note as diferenças na versão mobile:



Design Responsivo (RWD) ou Design Adaptativo (AWD)

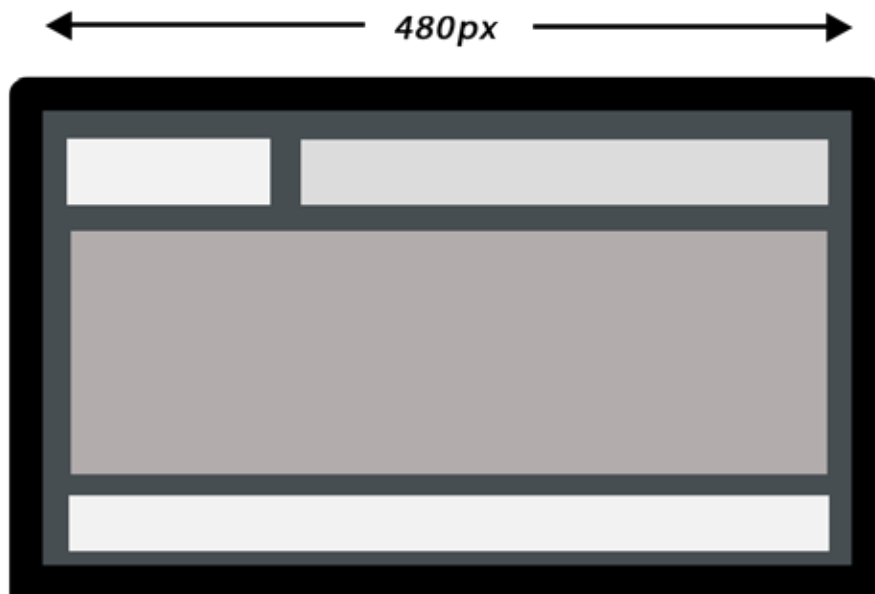
Muito provavelmente você já deve ter visto um design adaptativo e pensou que era Responsivo. Hoje em dia usa-se muito o termo Design Responsivo para sites com Design Adaptativos. Os dois são muito parecidos e tem o mesmo fim: fazer o site se adaptar aos diferentes formatos de tela. Mas mesmo assim eles são coisas distintas.

O conceito do Design Responsivo é acompanhar exatamente a resolução do dispositivo, já o Design Adaptativo trabalha com medidas fixas para cada resolução, como:

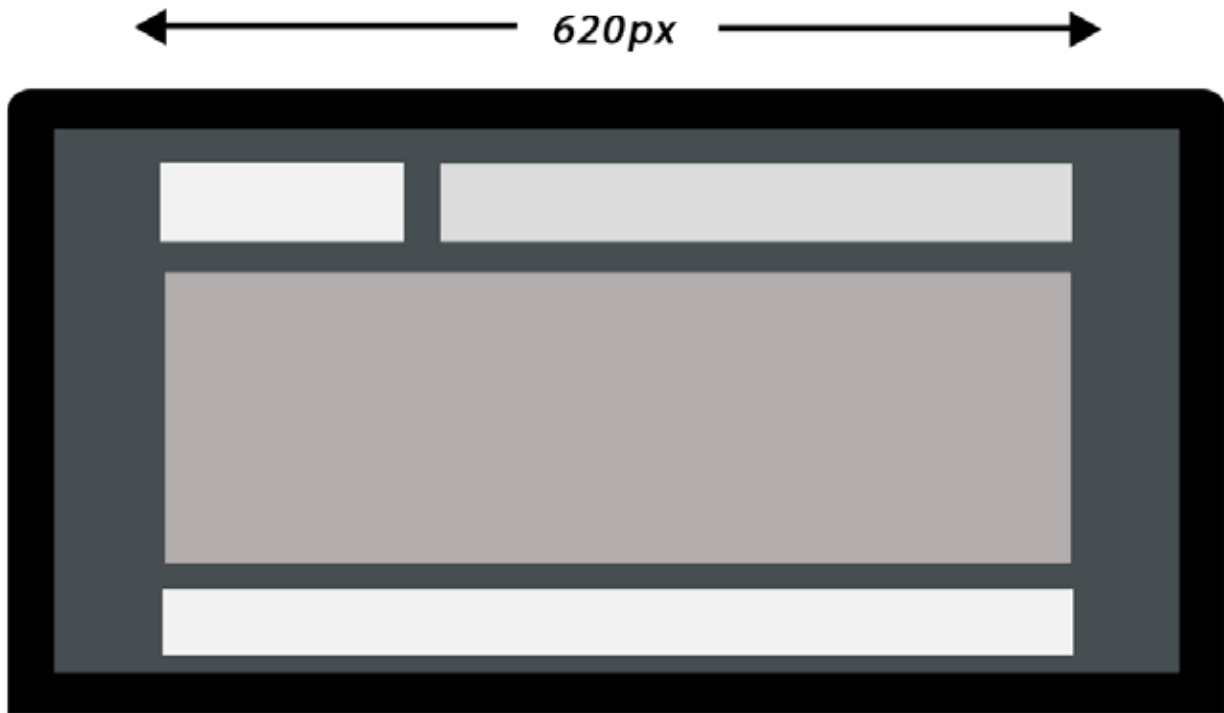
1200px, 1024px, 768px, 480px.

Irei usar a seguir essas medidas para exemplificar o AWD, mais especificamente a medida 480px.

Imagine um site AWD onde um dos pontos de quebra foi definido em 480px.



Agora imagine acessar esse mesmo site AWD numa resolução de 620px.
Veja o que aconteceria:



Note que o visitante teria acesso a versão do site para 480px mesmo acessando de uma tela com 620px.

Isso acontece porquê os breakpoints para as menores resoluções, do nosso exemplo, são 480px e 768px . Não existe um breakpoint para o 620px, portanto o site assume as definições determinadas no breakpoint 480px (já que 620px é menor que 768px).

Então o RWD, como já vimos aqui neste e-book, é mais flexível, enquanto que o AWD é mais restritivo.

E sabe qual delas é a melhor técnica? Eu te respondo a seguir :)

Não existe melhor ou pior!

O que determinará escolher um ou outro é a necessidade do projeto.

Em situações em que se precisa:

- adaptar um site já existente as resoluções menores;
- o layout do site é muito complexo;
- o projeto precisa passar experiências únicas e diferentes para cada dispositivo.

Talvez seja melhor usar o AWD.

Por outro lado, em casos em que:

- o projeto precisa funcionar bem em qualquer tipo de dispositivo;
- que o layout é menos complexo;
- que o projeto tem o foco maior na experiência do usuário.

Aplicar o RWD é o indicado.

O mais importante nessa questão não é quem é melhor ou pior e sim garantir que o site seja acessível ao maior número de pessoas independentemente do tamanho da tela.

Avalie o projeto e veja suas possibilidades e objetivos.



Considerações Finais

Web Design Responsivo não é mais luxo: é uma necessidade!

Hoje o acesso à internet via dispositivos móveis é muito superior aos desktops. Estamos conectados a todo momento.

Por isso nós desenvolvedores e designers não devemos somente pensar se o layout vai “encaixar” no dispositivo e sim em como podemos contribuir para que o visitante possa navegar no site da maneira mais tranquila e prazerosa possível.

Existem muito mais coisas para estudar sobre o RWD que esse singelo e-book não seria capaz de conter, como por exemplo a parte do código.

Prefiri não adentrar no assunto neste ebook, pois não era esse o foco. Mas saiba que essa parte é essencial para se construir interfaces responsivas.

Por isso, não termine seus estudos por aqui.

Espero que tenha sido útil para você. Muito obrigado por ler!

Guarde o Manual de Web Design Responsivo em um local fácil de lembrar e de acessar, para que você possa consultar sempre que precisar.

Qualquer coisa pode entrar em contato comigo, bele?

Até mais!

Forte abraço!



Referências

Livro:

Web Design Responsivo do Maurício Samy Silva (Maujor), editora Novatec.

Sites:

tableless.com

blog.popupdesign.com.br

smashingmagazine.com



PORQUE QUALQUER UM ACHA QUE
PODE DAR PITACOS, NÉ?



CHIEF
OF DESIGN

Siga o Chief nas redes

