# **Design Responsivo na prática: do rascunho ao digital**

Quer criar um projeto de design responsivo e não sabe por onde começar? Acompanhe passo-a-passo a criação de um layout do wireframe até a apresentação do design com indicações de artigos, dicas e ferramentas para facilitar o seu processo criativo.

**por** [**Dani Guerrato**](https://tableless.com.br/authors/dani-guerrato) 17/04/2013

Você já convenceu o seu chefe ou cliente que [Design Responsivo](https://tableless.com.br/introducao-ao-responsive-web-design/) é a última bolacha de chocolate do pacote, já sabe como trabalhar com media queries e como desenvolver códigos bonitos, semânticos e cheirosos. Mas a dúvida ainda persiste – por onde começar? Você não está sozinho. Esta é uma dúvida bem frequente. Por isto resolvi escrever um artigo básico só com dicas de prototipagem para design responsivo coletando algumas ferramentas úteis, artigos interessantes e comentando também um pouco da minha experiência pessoal ao lidar com o assunto no dia-a-dia gerindo um pequeno estúdio de webdesign.

Neste artigo vamos criar juntos um layout do wireframe ao design final. Assim você poderá acompanhar um processo passo-a-passo e adapta-lo para o seu fluxo de trabalho pessoal. Não tenho a pretensão de criar um guia definitivo, nem comentar técnicas de desenvolvimento mais avançadas. Mas tenho certeza que estas dicas vão ser um bom ponto de partida para você iniciar o seu projeto. Vamos a elas!

## **O conteúdo é o rei**

O primeiro passo para projetar – não apenas um site responsivo, mas qualquer layout para a internet – é o inventório de conteúdo. É ele que vai ditar qual é a melhor estrutura para o layout. Raramente o seu cliente vai possuir os textos, vídeos e imagens finais prontinhos para você diagramar. Mas o ideal é sempre ter ao menos uma noção de qual é o **tipo** de conteúdo que você pretende utilizar. Para isto é importante montar um briefing – sim, mesmo se o site for para você mesmo. Isto pode variar bastante de complexidade, mas o ideal é saber a resposta de algumas perguntas básicas sobre o tipo de conteúdo que você pretende apresentar para o mundo. O que você pretende exibir na página inicial? Notícias? Serviços? Produtos? Imagens? Qual o tamanho médio dos textos? Existirá um espaço destinado a anúncios publicitários? Qual é o formato do logotipo? Com estas respostas em mãos esta na hora de organizar todos estes elementos em uma estrutura lógica.

Vamos então projetar um layout de uma página inicial de uma empresa fictícia que deve conter as seguintes informações:

1. logotipo
2. navegação
3. banner apresentando produtos e serviços
4. blocos com imagens e textos curtos
5. créditos

## **Por onde começar o layout?**

A resposta é bem simples. Pelo papel. Não importa se você vai usar um bloco de notas velho, o verso de um guardanapo, um [template impresso da internet](https://sneakpeekit.com/) ou um [caderno especialmente projetado](https://appsketchbook.com/) para este fim, comece por um rascunho de wireframe no papel. Eu costumo desenhar basicamente duas coisas. Uma teia de navegação (“Que-link-leva-para-onde”) e o esqueleto básico do site. Não precisa ficar bonito. Minha habilidade para desenho se resume em bonecos de palito e partidas de Drawsome com estranhos da internet. Ainda assim é fácil criar um wireframe. Use quadrados para imagens e linhas para texto. O importante é você – e o resto da sua equipe se for o caso – terem um ponto de partida. O ideal é criar ao menos três versões principais: desktops, tablets e smartphones. Estas categorias são meio utópicas já que existem centenas de dispositivos que ficam no meio termo, mas você precisa ter um ponto de partida, certo? Escolher por qual dos três esqueletos começar é uma questão bem pessoal. Alguns advogam firmemente o mobile first, mas aqui por uma questão de didática vou começar pelo desktop.

## **Grids são seus amigos**

*Design responsivo é basicamente como montar um quebra cabeça onde você pode esticar, encolher, quebrar e dobrar estruturas. Realizar esta tarefa será muito mais fácil se você construir um layout sustentado por um grid.*

Não é absolutamente necessário utilizar um grid em seu CSS (embora seja uma prática recomendável), mas se o seu wireframe básico estiver organizado neste formato ajuda bastante na hora de projetar o design de maneira mais fluida, simétrica e organizada. Isto por que você pode simplesmente re-arranjar os blocos e colunas do seu layout de maneira mais lógica e matemática, o que vai refletir em uma maior coesão do design final.

Esta parte do fluxo de trabalho é bem parecida com criar um layout para design “normal”. O primeiro passo portanto é criar o tal do grid, basicamente um conjunto de linhas “invisiveis” que vão sustentar o seu design. Pense que você vai ter que quebrar esta estrutura em pedaços menores e, para manter a simetria o ideal é escolher um número par que possa ser divisível por 2, 3 e/ou 4… Por isto grids de 12, 16, 18 ou 24 colunas são bem comuns. (Você pode escolher o número que quiser. Eu mesma já trabalhei com 14 colunas. Deu mais trabalho. Não diga que não avisei… ). Não se esqueça do espaço das margens entre as colunas.

## **O wireframe**

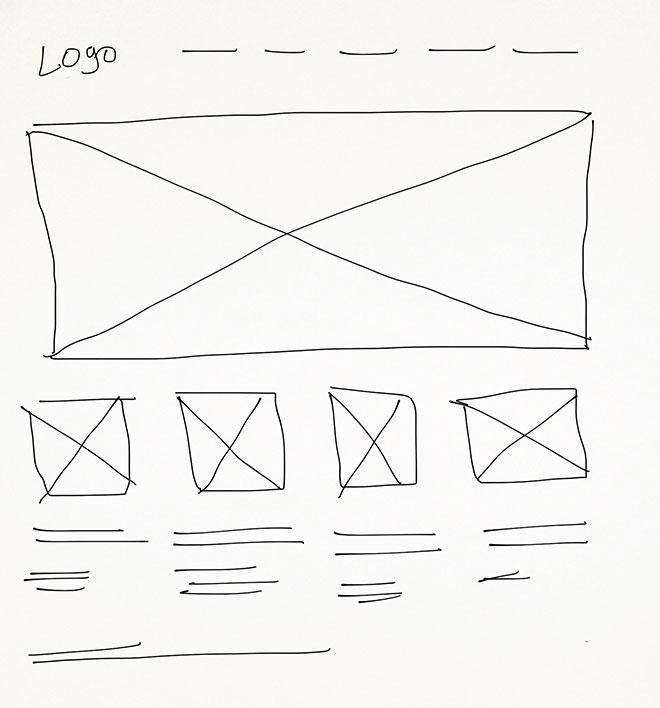
Se você precisar reescrever completamente todos os elementos do CSS você está fazendo design adaptativo, o que tem seus métodos mas não é responsivo.

*Design responsivo é focado na economia. Economia de tempo, economia de peso de arquivos, economia de código. Pense em escrever estruturas que, embora sofram adaptações, possam ser re-aproveitadas.*

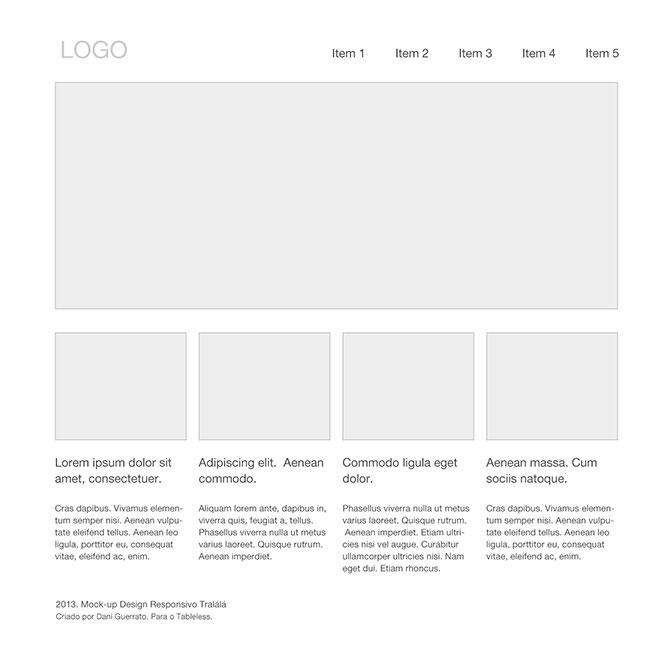
### **Desktops**

Este é o exemplo de wireframe que vamos criar passo-a-passo para o nosso exemplo. A estrutura é bem simples: um logotipo no canto superior esquerdo, um menu no topo a direita, um banner, 4 destaques com texto e foto e um rodapé.

Este é o meu rascunho inicial:

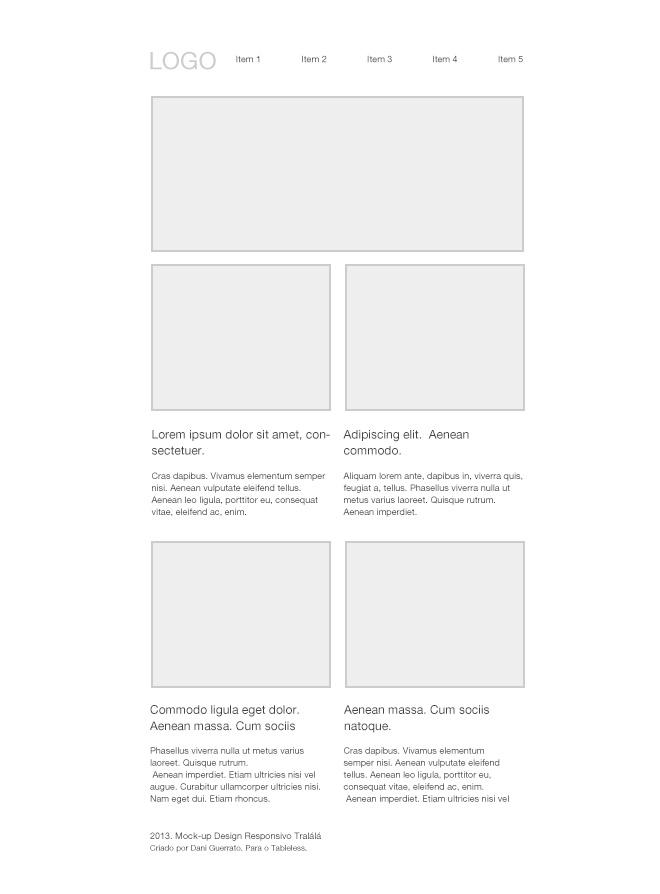


E agora o mesmo modelo recriado no Photoshop:



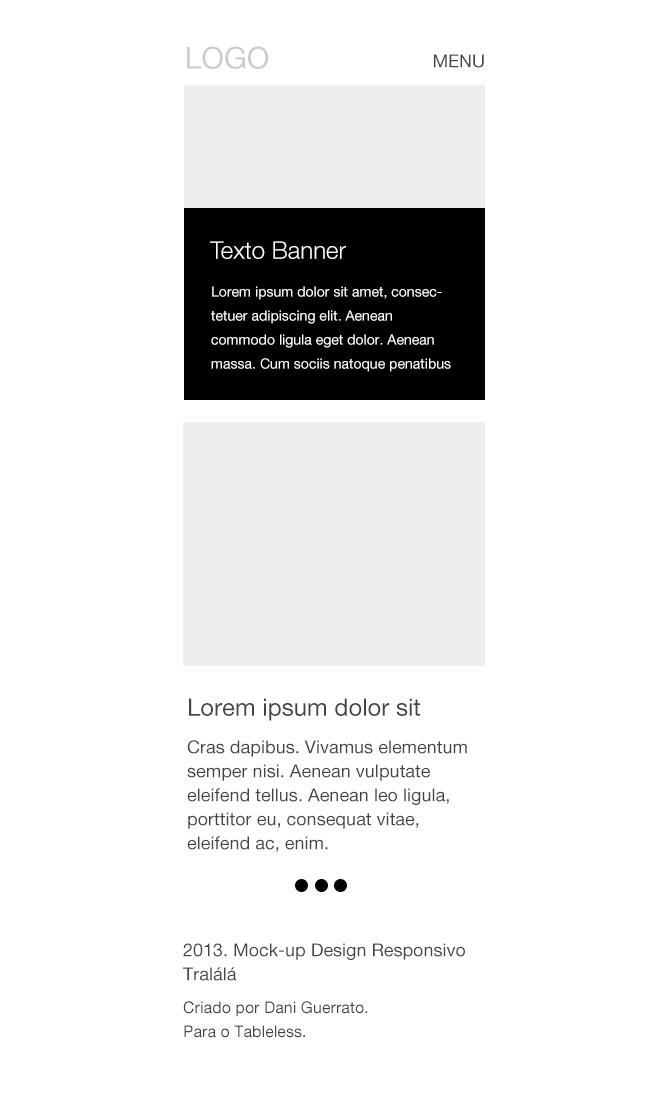
### **Tablets**

A técnica para adaptar esta estrutura para as outras versões é simples. Diminua o número de colunas no grid. Se inicialmente tinhamos 16 colunas no desktop, teremos 10 no tablet e 4 no smartphone, por exemplo. O conteúdo deve se re-arranjar para caber nesta estrutura menor. Então no tablet ao invés de 4 destaques lado-a-lado temos 2 fileiras com 2 destaques cada. Para adequar-se a estas mudanças as imagens ficaram maiores. Outra modificação foi um ajuste no tamanho do texto do menu.



### **Smartphones**

Como a tela dos smartphones é ainda menor, é necessário re-arranjar as estruturas novamente e fazer alguns outros ajustes. Isto não significa necessariamente diminuir os elementos de tamanho. Lembre-se que a maior parte dos dispositivos móveis utilizam touch screens. Você deve adaptar os elementos considerando esta área de toque. Links muito pequenos e juntinhos são difíceis selecionar. O ideal é que o usuário possa navegar no site sem precisar dar zoom. Por isto optei por colapsar os elementos em um menu drop-down. Os destaques agora ocupam o espaço total do wrap. Optei por juntar os 4 destaques em uma navegação estilo slider (navegáveis através de seletores em forma de “bolinhas”). Outra mudança estetica significativa foi o banner. Como pretendo incluir texto em HTML com os serviços principais da empresa optei por deixa-lo em uma caixa preta abaixo da foto. Desta maneira as imagens ficam mais visíveis e o formato não prejudica a leitura do texto.



## **Resoluções**

Existem dezenas de resoluções diferentes e, embora este seja o objetivo final, é bem difícil ter um layout que vai ficar perfeito a cada ponto de quebra (difícil, não impossível). O ideal, portanto, é ter em mente quais são os formatos mais comuns e focalizar para que ao menos nestes estágios o design esteja funcionando perfeitamente. Considere portanto estas resoluções básicas:

* 1200 pixels – Desktops com monitores widescreen.
* 960 pixels – Tablets em formato paisagem e monitores antigos.
* 768 pixels – Tablets em formato retrato.
* 480 pixels – Smartphones em formato paisagem.
* 320 pixels – Smartphones em formato retrato.

Para testar o seu HTML final em diferentes resoluções você pode utilizar algum add-on ou [bookmarklet](https://seesparkbox.com/foundry/media_query_bookmarklet) para o seu browser.

## **O mock-up**

Hora de criar esta estrutura no seu programa gráfico favorito. Você pode montar o seu próprio grid utilizando linhas guias ou baixar um modelo pronto como o [Responsive Grid PSD](https://dribbble.com/shots/410635-Responsive-Grid-PSD/) ou o [Frameless](https://framelessgrid.com/). Para este tutorial utilizei uma versão modificada do Frameless que você pode [baixar aqui](https://raw.githubusercontent.com/diegoeis/tableless-static-images/master/2013/04/grid.zip).

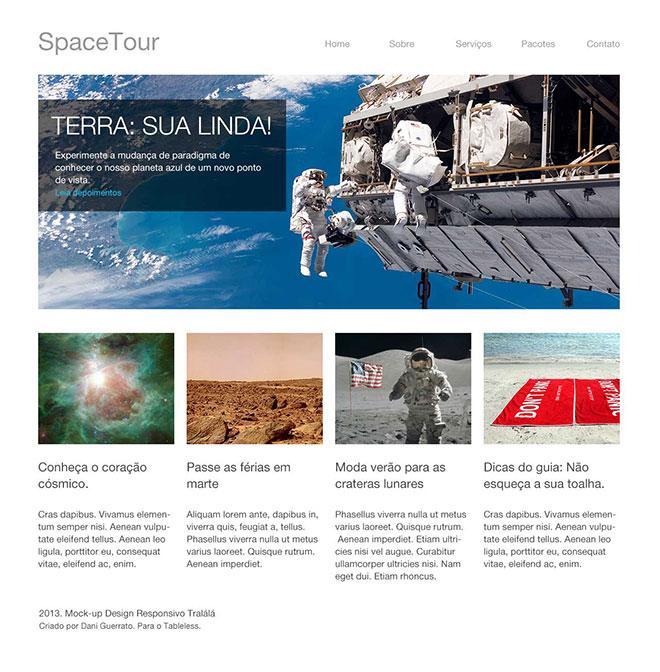
### **Nota sobre retina display**

É importante ter em mente que alguns aparelhos tem a densidade de pixels superior a 1. É interessante, em termos de qualidade de apresentação do layout, criar um arquivo com o dobro da resolução nestes casos. Então dobre o tamanho do seu PSD na versão para smartphones (o que significa um mockup de 960x640px). Assim, na hora de exportar os elementos, você pode criar imagens em alta resolução para dispositivos retina.

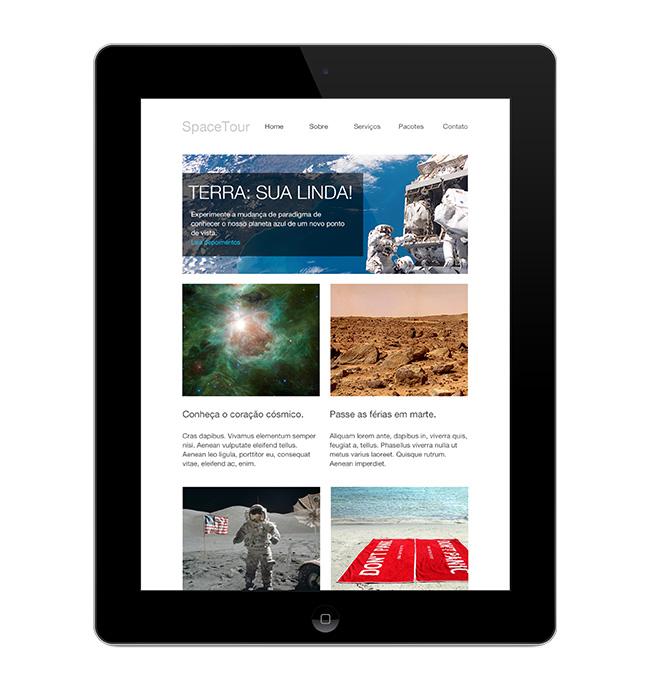
### **Apresentação**

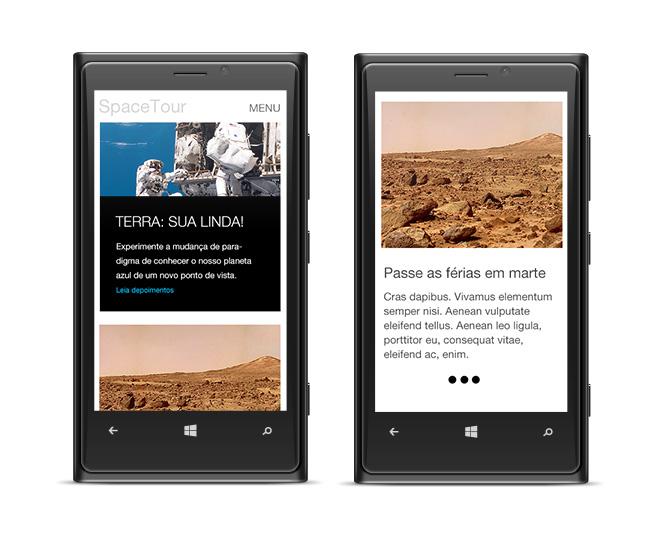
Na minha empresa criamos modelos estáticos para serem aprovados pelos clientes antes de partirmos para etapa de desenvolvimento do HTML/CSS. Na minha experiência, mesmo se nenhum conteúdo for fornecido previamente, fazer um esforcinho a mais e utilizar imagens e textos parecidas com as finais ajudar o seu layout a ser aprovado. Isto acontece por que, ao contrário de nós designers e desenvolvedores, as pessoas “normais” possuem naturalmente uma certa dificuldade de abstrair que aqueles quadrados e caixinhas são um site. Adicionar imagens e textos a um tema pode dar mais trabalho, mas é acrescentar contexto e propósito ao seu design, o que por sua vez causa muito mais empatia e identificação.

Para o nosso exemplo criei a empresa fictícia Space Tour.



Mesmo com um conteúdo falso aplicado ainda é difícil de imaginar como o produto final vai se comportar online, principalmente no caso de dispositivos móveis. Para que o cliente possa visualizar como o layout vai se comportar é interessante utilizar mock-ups de hardwares reais. Basta procurar no Google que existem diversos recursos gratuitos para este fim. O site [PSD Covers](https://www.psdcovers.com/), por exemplo, possui actions e templates de Photoshop em alta resolução que podem ajudar bastante na apresentação final. O][10](https://dribbble.com/) também pode ser uma boa fonte para recursos gratuitos.





## **Um processo em evolução**

Com o passar do tempo este fluxo de trabalho torna-se natural. Ao observar um layout você consegue mentalmente criar os pontos-de-quebra e algumas fases deste processo podem ser puladas. Você pode, por exemplo, trabalhar com apenas dois formatos de wireframe: um para computadores/tablets e um para smartphones. Existem pessoas que preferem ainda queimar completamente estas etapas e desenvolver diretamente no CSS. Cada um tem o seu modo de trabalho e nenhum é necessariamente melhor ou pior que o outro. Vale a pena testar diferentes abordagens até encontrar o que funciona melhor para você, sua equipe e seus clientes.

## **Conclusão**

Bem, agora você já sabe como criar um protótipo de design responsivo. Com o layout aprovado entra a segunda etapa: colocar ele para funcionar. Para isto existem diversos [artigos sobre Design Responsivo](https://tableless.com.br/?s=design+responsivo) aqui no Tableless que podem te ajudar como . Ou você pode ainda consultar a [série de artigos do meu blog](https://blog.popupdesign.com.br/?s=Design+responsivo) . Quer mais? O repositório [Responsive Resources](https://bradfrost.github.io/this-is-responsive/resources.html) possui uma lista gigante de artigos, recursos e ferramentas úteis sobre o tema. Bons estudos!