

Um guia completo sobre grids para design responsivo

É possível pensar que o papel de criar um Design Responsivo esteja todo nas mãos do time de design (Ai ou UID). Eles tem que ser responsáveis por entregar os PSD's com os layouts em pelo menos três resoluções, para desktop, tablet e mobile? Esqueça isso. Quem faz com que o design seja responsivo ou não, no final das contas, é o desenvolvedor front-end.

Muitas agências ainda terceirizam o desenvolvimento. Quando você não tem o desenvolvedor front-end do seu lado dentro de casa, a comunicação entre os times de design e dev tem que ser impecável, caso contrário, as chances de um resultado abaixo do esperado é grande.

Mas como o time de Design pode resolver esse problema de comunicação e colocar no ar um produto que seja exatamente o que foi pensado? Entendendo do assunto. Lembra que eu disse que o responsável por tornar o Design Responsivo é o desenvolvedor? Então vamos lá...

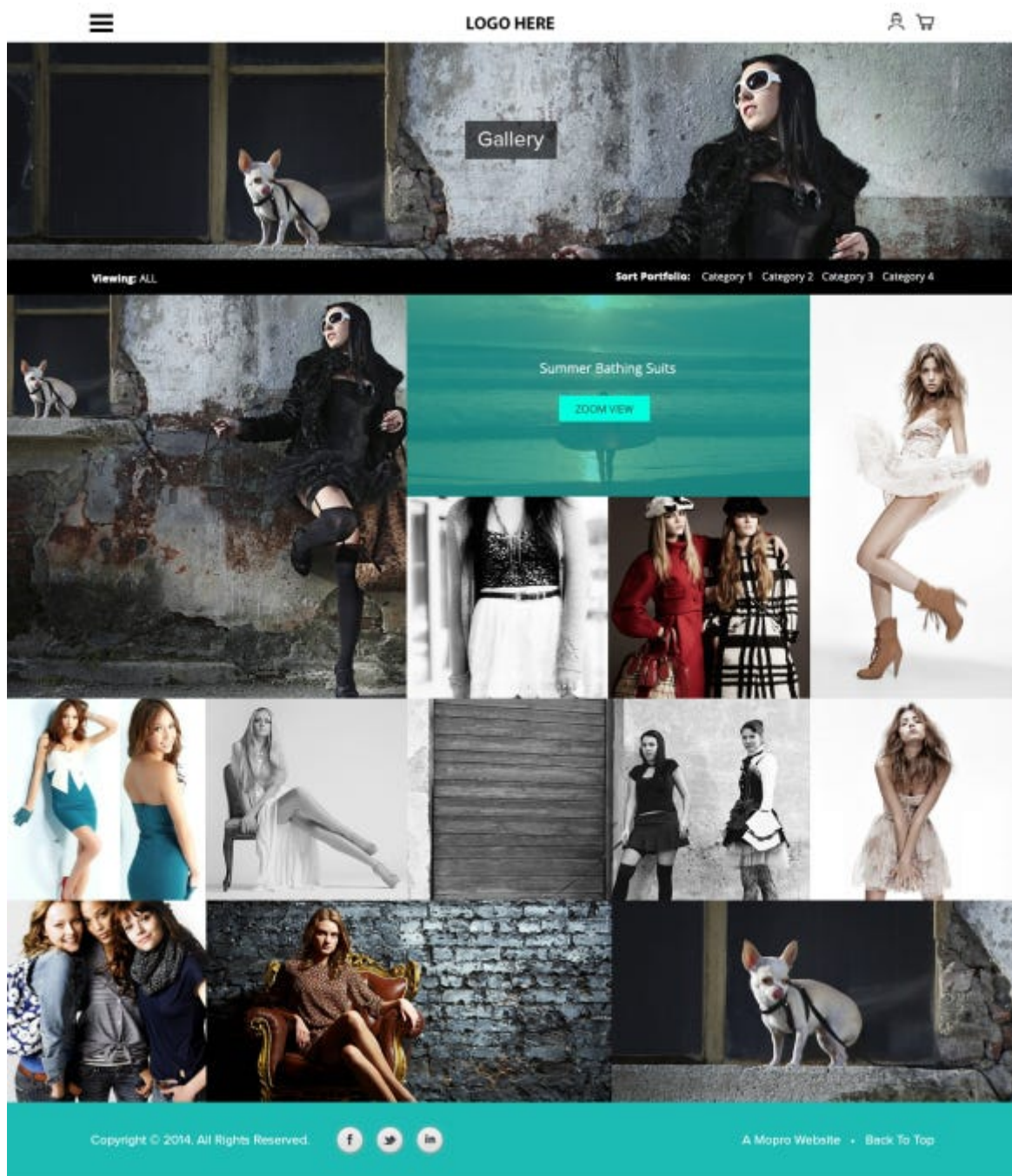
Tecnicamente falando...

Design responsivo é um conceito, não é uma técnica.



Ele representa a adaptação de uma interface a qualquer resolução de tela, adaptando a experiência de navegação de acordo com as necessidades dos usuários e também de cada tipo de mídia. Ele é o melhor amigo da experiência do usuário.

O Design fluído (liquido) é uma técnica de desenvolvimento



Ele utiliza medidas relativas para que o layout de um site possa aumentar ou diminuir de tamanho de acordo com o tamanho da área de visualização.

Ou seja, você pode ter todo seu conteúdo adaptado a uma resolução de mobile, por exemplo, e mesmo assim ele pode não ser responsivo. Por quê?

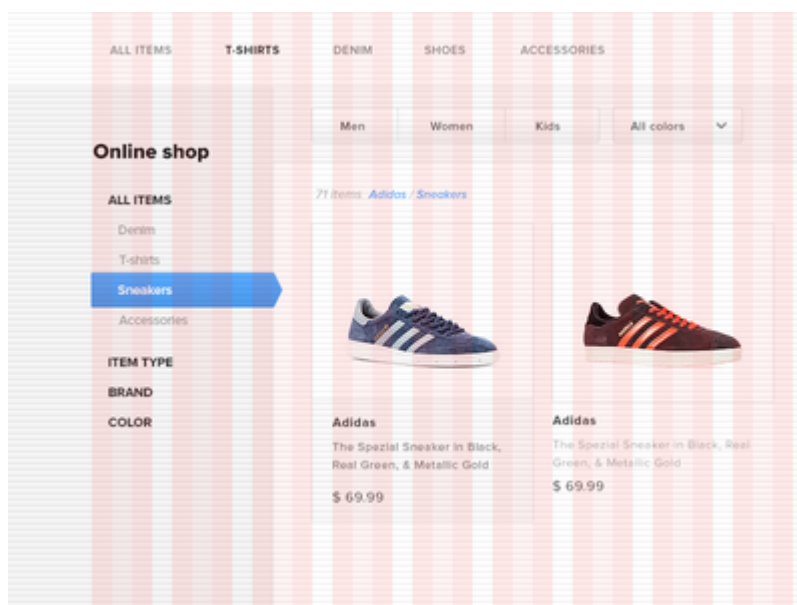
Porque imagine que o Design Fluído de um Blog de Receitas apenas “diminua” o site para que ele caiba em uma resolução mobile. O sidebar desse Blog vai encolher, vai ocupar digamos que 30% da pequena tela do smartphone, e o conteúdo irá ocupar os outros 70%. Temos aí uma péssima experiência de uso.

O que o Design Responsivo faria é adaptar esse conteúdo a uma melhor experiência, transformando o sidebar em um menu de navegação, por exemplo, e liberando espaço para o mais importante: o conteúdo do Blog.

O Design Responsivo não simplesmente encolhe o conteúdo, mas adapta os elementos a fim de entregar ao usuário o melhor conteúdo e navegação para aquele cenário.

Para colocar em prática esse conceito é preciso de técnica, e uma das melhores técnicas é o Design Responsivo Fluído: ele é Responsivo, ou seja, garante a melhor experiência ao usuário, e utiliza a técnica de Design Fluído na construção da sua base em HTML e CSS para garantir que o conteúdo se torne flexível.

O Grid



É comum que a função de utilizar o Grid acabe sendo exclusiva do Designer de Interface? Parece que sim. Mas já faz algum tempo que tenho observado a quantidade enorme de UI Designers que não utilizam Grid em seus trabalhos (usar uma linha guia aqui e ali não é a mesma coisa). Sendo assim, confiar no UI Designer para que ele aplique e estude um Grid que seja adequado para o projeto pode se tornar um problema.

Definir Grid pode e, para mim, deve ser uma tarefa do AI. Muitos designers passam um tempo olhando para uma tela em branco, tentando descobrir onde os elementos devem ser posicionados, mas, se você tem um Grid Flexível, muitos desses problemas já estão resolvidos, deixando o designer com mais tempo para trabalhar em design gráfico e outros componentes da página.

Em vez de tomar decisões arbitrariamente, os Grids permitem criar uma linguagem visual comum em toda a página. Os componentes se relacionam entre si, o que torna mais fácil para os usuários scanarem conteúdo e desenvolver familiaridade com ele.

Grid é uma malha que divide a tela em partes proporcionais e possibilita a distribuição do conteúdo de forma uniforme, proporcionando equilíbrio visual e estrutural. Construir um Design Responsivo se torna muito mais fácil se ele for apoiado em um Grid Flexível.

Benefícios do Grid...para o usuário

- Facilita a leitura e o scaneamento dos elementos.
- Melhora a experiência visual.
- Menos esforço cognitivo para interpretar o conteúdo.
- O olho humano percebe a distribuição proporcional dos elementos e associa as informações com maior facilidade.

Benefícios do Grid...para o designer

- Rapidez para adicionar, alterar e mover elementos.
- Padrões de espaçamento e alinhamento proporcionam uma linha de trabalho uniforme.
- Linguagem visual comum em toda a página.
- Mantém a consistência do trabalho caso outro designer mexa no arquivo.

Benefícios do Grid... para o desenvolvimento

- É o caminho para que o layout que irá ao ar fique o mais pixel perfect possível.
- Atualização do layout de maneira consistente.
- Rapidez na construção do front-end do site.

Grid Responsivo

Se você está criando um layout responsivo a primeira coisa que tem que saber é que tudo funciona com base em porcentagens(%). O Grid Fluído, como o nome diz, vai sempre dar a impressão de ocupar toda a tela.

Qual melhor formato?

O famoso formato 960gs foi bastante usado durante muitos anos, quando a resolução mais popular era 1024x768px. Nos dias de hoje, 13,28% dos monitores utilizam 1024x768px como resolução. A grande maioria utiliza 1366x768px como resolução. Ou seja, utilizar o Grid 960g pode tornar o design limitado e preso a uma grade ultrapassada. O mais comum, vem sendo utilizar uma largura de **1140px**, para que no Grid para que se adapte bem as maiores resoluções de monitores.



Joaquin Phoenix: I'm Still Here

Tickets from \$12.00
Special Tickets from \$90.00

★★★★☆

[Get Tickets Now](#)



All Events


Concerts

Sports


Basketball (1093)
Boxing & Wrestling (588)
Ice Hockey (368)
Football (858)
Handball (51)
Motorsports (4037)
Equitation (73)
More (1256)

Classical
Comedy
Musical & Show


Top Events (25)




Justin Timberlake
Tickets from \$45.00
[Tickets](#)




Alicia Keys
Tickets from \$25.00
[Tickets](#)




Deshaun Martin
Tickets from \$52.00
[Tickets](#)




X Games
Tickets from \$10.00
[Tickets](#)




Bond of Heroes
Tickets from \$35.00
[Tickets](#)




The Fray
Tickets from \$38.00
[Tickets](#)




Aerosmith
Tickets from \$30.00
[Tickets](#)



Phantom of the Opera
Tickets from \$35.00
[Tickets](#)




TED Talks
Tickets from \$10.00
[Tickets](#)




Psy
Tickets from \$35.00
[Tickets](#)


Best Selling (20)




Disney's Tron
Tickets from \$45.00
[Tickets](#)




Engine Strikes Back
Tickets from \$25.00
[Tickets](#)




Into the Wild
Tickets from \$52.00
[Tickets](#)




Bruno Mars
Tickets from \$10.00
[Tickets](#)




M83
Tickets from \$35.00
[Tickets](#)




Imagine Dragons
Tickets from \$20.00
[Tickets](#)




Everyday With Jay-Z
Tickets from \$30.00
[Tickets](#)



Black Swan
Tickets from \$35.00
[Tickets](#)



Foster the People
Tickets from \$10.00
[Tickets](#)



Huel
Tickets from \$35.00
[Tickets](#)

Local Events in San Francisco, CA:

[Change Location](#)

Upcoming (25)

Apr 23

Sacramento Kings vs. Los Angeles Clippers
AT&T Park
[Tickets](#)

Apr 24

Vampire Weekend
Piedmont Hall
[Tickets](#)

Apr 27

SON JUVI Because We Can - The Tour
San Mateo Center
[Tickets](#)

Apr 30

NASCAR Sprint Cup
Sprint Raceway
[Tickets](#)

May 1

Charles Schwab Cup Championship
TPC Harding Park
[Tickets](#)

[See More](#)

Recommended (70)

Apr 21

San Francisco Giants vs. Oakland A's
AT&T Park
[Tickets](#)

Apr 22

The Rolling Stones
Oracle Arena
[Tickets](#)

Apr 27

PINK: The Truth About Love Tour
Oakland Arena
[Tickets](#)

Apr 28

Jane Krakowski
Comcast Center SF
[Tickets](#)

May 1

Charles Schwab Cup Championship
TPC Harding Park
[Tickets](#)

[See More](#)

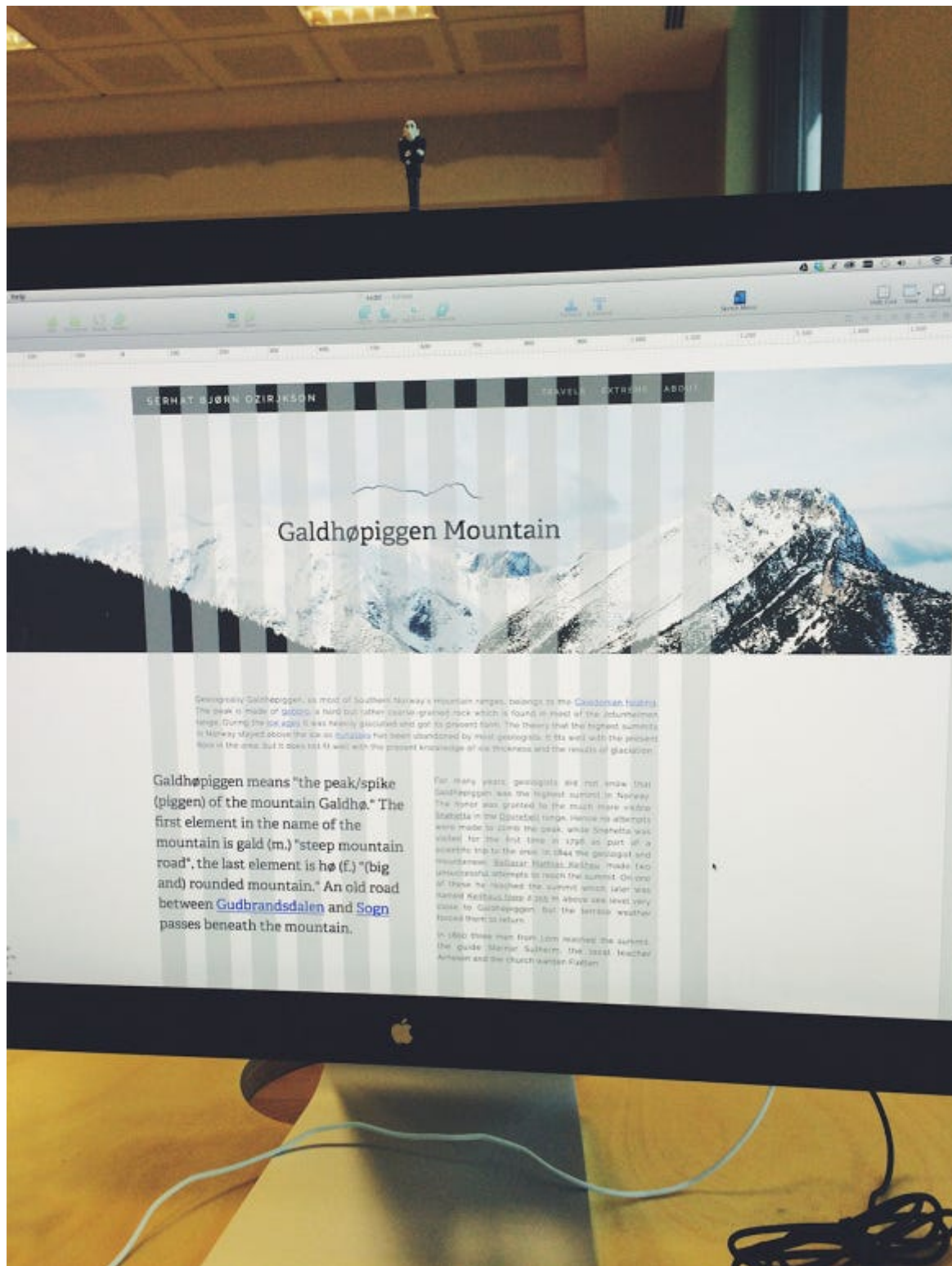
Ticket Alarm (5)

San Francisco Giants vs. LA Dodgers
AT&T Park
[Tickets](#)

Step Into The Liquid Movie Debut
The Market
[Tickets](#)

Star Wars Adventures Tour
Lucas Ranch
[Tickets](#)

Como escolher o número de colunas?



Isso vai depender de você e do seu projeto. A razão para trabalhar com 12 colunas, que é o formato mais utilizado, é por causa da flexibilidade proporcionada e por ser capaz de dividir o layout uniformemente. Ou seja, você pode dividir 12 por 2,3,4 e ter uma serie de modulações disponíveis (2 x 6 , 3 x 4) e também em várias outras combinações. Se você dividir 12 colunas por 2, vai perceber que consegue ter 6 conjuntos de 2 colunas sem que sobre nenhuma coluna.

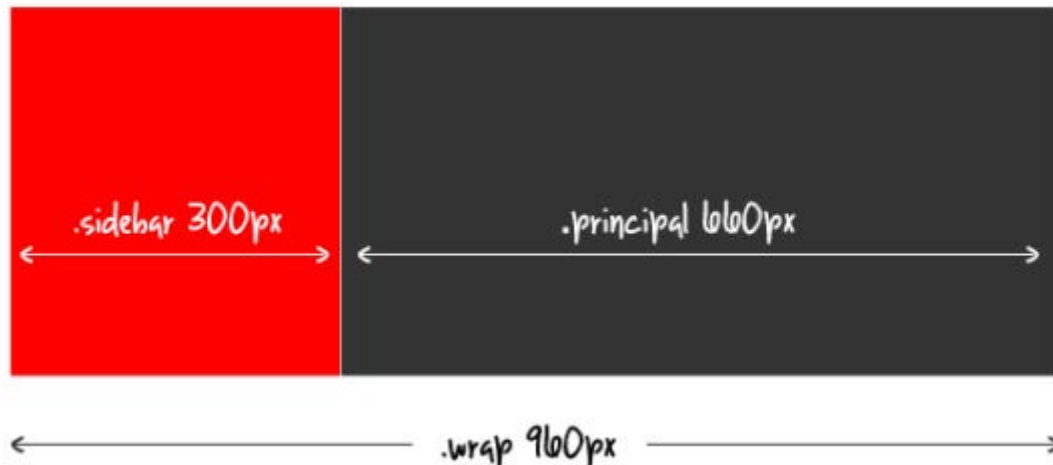
Mas o número de colunas pode variar de projeto para projeto. Basicamente, quanto maior o número de colunas, (14 ou 21, por ex.) maior o quantidade de possibilidades

para dividir o layout. Geralmente esse tipo de Grid é utilizado quando se tem muito conteúdo editorial e o design é projetado com uma largura de 1140px ou mais.

Calculando as porcentagens

Os desenvolvedores da PopUp Design explicam de um jeito bem fácil como calcular a porcentagem de um elemento.

Digamos que você tem as dimensões abaixo no Photoshop:



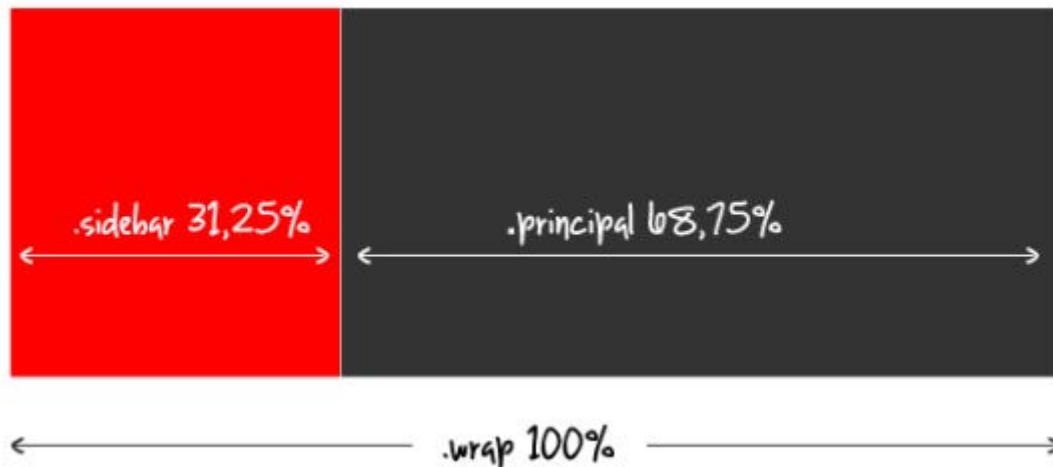
Ou seja, temos um Sidebar de 300px e uma área de conteúdo de 660px. O wrap, ou seja, o repositório que envolta todos os elementos, tem 960px.

Para calcular esses valores em porcentagens (aliás, qualquer valor de Pixel para Porcentagem), é só utilizar a fórmula:

$$\text{objeto} \div \text{contexto} = \text{resultado.}$$

O wrap nesse caso é o Contexto. Para calcular o sidebar (nosso Objeto), fazemos assim: sidebar \div wrap = resultado. Para descobrir a porcentagem é necessário multiplicar o resultado por 100. No caso do sidebar temos $300 \div 960 = 0.3125$, e

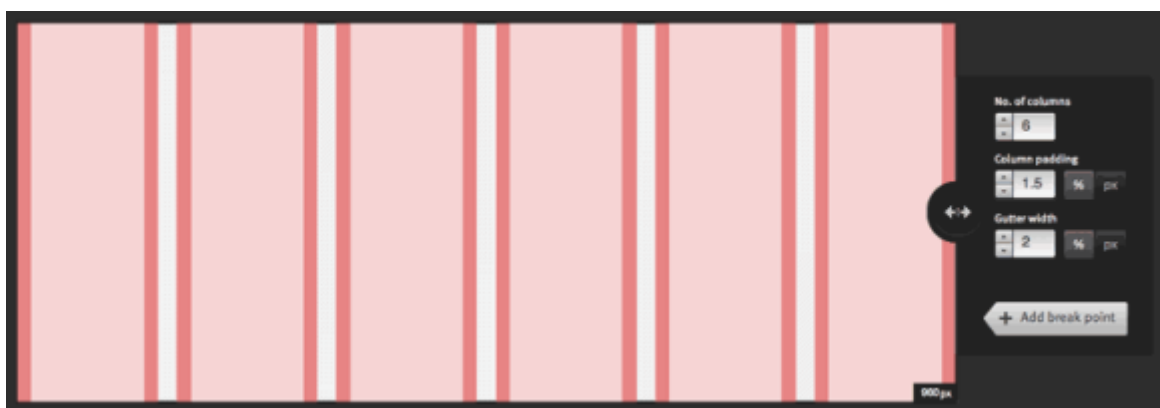
multiplicando o resultado por 100, temos finalmente nossa porcentagem, que é 31.25%.



O Grid dos designers

Existem muitas maneiras de criar um Grid de marcação (aquele que você sobrepõe sobre seu design no Photoshop ou qualquer outra ferramenta visual, afinal, você não é programador e não vai criar um Grid em CSS e HTML), você pode criar manualmente, utilizar Actions no Photoshop entre outras maneiras. Hoje eu vou mostrar para vocês o que para mim é o mais fácil e pode ajudar vocês a construir um Grid Flexível.

Calma, você não precisa fazer mil e uma contas para conseguir fazer seu Grid Responsivo. Sabe aquele ditado que “fiz design para se livrar dos números”? É mais ou menos assim. Quando se desenvolve um layout responsivo, a parte mais casca grossa acaba ficando para os desenvolvedores front-end. É preciso converter pixels em porcentagens e EM's e fazer algumas continhas bem chatas. Mas mesmo assim é importante entender o processo do desenvolvimento front-end para que você e seu time de Design consigam passar para o time de desenvolvimento exatamente o que foi pensado.



[Gridpak](#) é um excelente gerador de Grid Flexível que pode te salvar de ter que fazer alguns cálculos. É só digitar a quantidade de colunas que você quer, o valor do

padding entre as colunas e o valor da calha. Defina um breakpoint e exporte seu Grid em PNG, CSS entre outros formatos.

O número de colunas de um Grid costuma diminuir a medida que a resolução de tela diminui, isso acontece porque se você diminui o tamanho da tela, você tem que aumentar o tamanho dos elementos, e um Grid com poucas colunas acaba aumentando a proporção dos elementos.

O mais comum, é utilizar sempre um número par para as colunas. Sendo assim, para Tablets o número de colunas costuma ser de 8 col. e para smartphones 6 col. 8 e 6 colunas permitem uma serie maior de modulação disponível.

Mas eu sou designer, porque preciso entender sobre porcentagens, EM's e códigos?

Da próxima vez que você for falar com o desenvolvedor do seu Design Responsivo, vai poder ficar de olho para ver se ele está usando porcentagens e EM's na construção do CSS (é isso que vai garantir que seu design fique Fluído).

Além disso, é importante que no seu documento de especificações você deixe bem claro o Grid utilizado, a quantidade de colunas, largura das colunas e o padding e gutter do seu Grid, todos esses dados, claro, em porcentagens.

Assim vai ficar muito mais fácil acompanhar o desenvolvimento do seu design e entender o por quê das coisas.

Entendi, mas porque não usar apenas Break Points?

Digamos que você e/ou seu desenvolvedor decidem não tornar o design flexível, ou vocês não tem tempo ou verba para isso. Então fica decidido que o design será construído em cima de Break Points que irão indicar quando a mudança do design deve ser feita. Bom, como você já deve saber, existem inúmeros dispositivos por aí, e a cada ano algo novo é lançado (que o diga o novo iPhone 6 e iPhone 6s). Fica impossível prever qual a nova resolução que será adotada pelo mercado. Se você for tentar satisfazer a todos, vai terminar tendo que adaptar seu layout a algumas dezenas de breakpoints e vai acabar louco fazendo isso. Ok, você pode adaptar seu layout aos breakpoints mais “famosos” como 1200 px (Desktops widescreen), 960px (Monitores antigos), 768 px (Tablets), 480 px (Smartphones em formato paisagem) e 320 px (Smartphones em formato retrato) e pode até ser que seu design fique agradável em um iPhone4s, mas quem irá garantir que ele ficará agradável no novo iPhone6s ou um modelo de Android?

Utilizar um Design Adaptável (que só utiliza BreakPoints), pode ajudar se você precisar resolver um problema a curto prazo, mas definitivamente não é uma solução permanente.

Mas o Grid não vai limitar a minha criatividade?

De jeito nenhum, pelo contrário. Além de trazer organização e clareza para o design, nada o impede de criar áreas no seu layout que podem ser digamos... “Grid Free”, espaços para aplicar uma ilustração ou algo do tipo. O Grid também pode ser usado em qualquer formato de design, seja ele horizontal, vertical, não importa. O Grid não deve limitar sua criatividade ele é apenas uma ferramenta para melhorar o seu trabalho.

