

Projeto de Interfaces para Dispositivos Móveis

Alan Vasconcelos

2018

Projeto de Interfaces para Dispositivos Móveis

Alan Vasconcelos

© Copyright do Instituto de Gestão e Tecnologia da Informação.

Todos os direitos reservados.

Sumário

Introdução.....	7
Objetivos da disciplina.....	7
Tópicos principais.....	8
 Capítulo 1. Princípios de design de interação	9
Princípio 1 - Visibilidade	10
Princípio 2 - Feedback	11
Princípio 3 - Restrições	12
Princípio 4 - Mapeamento	12
Princípio 5 - Consistência.....	13
Princípio 6 - Affordance	14
 Capítulo 2. Paradigmas de interação	16
O paradigma desktop	16
Principais diferenças entre desktop e mobile	17
Padrões de interação mobile.....	20
Gesture	20
Feedback	21
 Capítulo 3. Telas, resoluções, densidades e unidades de medida	24
A diversidade no ecossistema mobile	24
Tamanhos e densidades	26
Como o dp funciona?	29

Desenhando para múltiplas telas	30
Algumas ferramentas úteis.....	35
Capítulo 4. Briefing de projetos mobile	37
Etapas típicas de um projeto mobile	37
Elementos do briefing.....	39
Conclusão	44
Capítulo 5. Modelagem de usuários com Personas	45
Por que modelar usuários?	45
O que é Persona?	46
Como montar a prancha da persona.....	49
Alguns exemplos.....	50
Como obter dados para construir Personas	52
Alguns cuidados.....	53
Personas típicas para projetos mobile	53
Capítulo 6. Modelagem de tarefas com User Story Mapping	55
O que é uma história?.....	56
Por que mapear e não listar?	57
Criando um User Story Mapping	58
Benefícios em desenvolver em releases.....	62
Outros benefícios em se fazer um USM.....	63
Capítulo 7. Arquitetura de informação	65
O que é e para que serve a A.I	67
Padrões de interação. Por que em mobile é diferente?	68

Esquemas típicos de navegação mobile	68
1- Hierarquia	69
2- Hub & spoke	69
3- Nested doll	70
4- <i>Tabbed view</i>	71
5- Dashboard	71
6- Filtered view	72
Algumas considerações	73
Capítulo 8. Prototipagem	74
Benefícios	75
Construindo o protótipo rápido	76
Passo 1	77
Passo 2	77
Passo 3	78
Passo 4	79
Passo 5	79
Passo 6	80
Protótipos funcionais (wireframes)	80
Considerações	81
Capítulo 9. Elementos de design	83
Design Stencils	86
Ferramentas	87

Considerações	88
Considerações finais sobre interfaces para dispositivos móveis.....	89
Referências.....	90

Introdução

Os dispositivos móveis são os PC's do século XXI. Essa é uma realidade inegável e que dificilmente irá se reverter. O crescimento vertiginoso da adoção dos dispositivos móveis pelas pessoas é comparado ao do rádio, TV e internet. Em 2011, os smartphones foram mais vendidos mais que PC's e, em 2013, o número de dispositivos móveis conectados excedeu à população mundial. Além disso, os dispositivos móveis são um negócio bastante lucrativo. Em março de 2012, a Apple atingiu a marca de 25 bilhões de downloads na sua loja de aplicativos. Em 2015, a companhia espera faturar 100 bilhões de dólares. Esse é o cenário que nos espera!

A preocupação com usabilidade e experiência de uso deixou de ser um luxo para se tornar o ponto de sobrevivência no mercado mobile. A extrema concorrência e a variedade de novos aplicativos abrem inúmeras possibilidades de escolhas para os usuários que estão cada vez mais exigentes, não só com aspectos funcionais, mas com design, facilidade de uso e até mesmo estética.

Objetivos da disciplina

Esperamos, com essa disciplina, capacitar o aluno no projeto de interfaces para dispositivos móveis, compreendendo técnicas desde briefing inicial até a entrega do layout para o desenvolvimento de forma alinhada com os princípios de design, interação e experiência do usuário de cada plataforma. Não é o objetivo da disciplina formar designers de interface, uma vez que esse campo de estudo demanda uma abordagem mais ampla e outros conhecimentos correlatos. De toda forma, esperamos que ao final dessa disciplina, o aluno seja capaz compreender e demandar soluções que cercam o projeto de interfaces mobile.

Tópicos principais

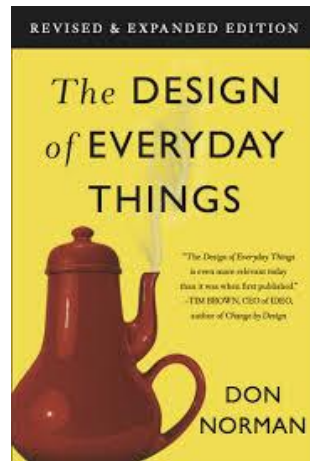
Existem várias metodologias, processos e frameworks destinados ao desenvolvimento de produtos digitais, especialmente, aqueles voltados para os dispositivos móveis. Abordaremos aqui um fluxo de atividades que irá desde o briefing e levantamento de requisitos, passando por prototipação e mockups até o leiaute das telas. Vamos a elas:

1. Princípios de design de interação
2. Paradigmas de interação (Gesture, Scrolling, Feedback, Orientação)
3. Diferenças entre as telas, resoluções, densidades e unidades de medida
4. Briefing para projeto mobile
5. Modelagem de usuários com Personas
6. Modelagem de tarefas (User Story Mapping)
7. Arquitetura de Informação
8. Técnicas de prototipagem
9. UI Guidelines, (Android, iOS)

Capítulo 1. Princípios de design de interação

Donald Norman, autor do livro “O design do dia a dia”, enumerou os seis princípios de design que servem para qualquer produto interativo.

Figura 1 – “O design do dia a dia”.



Esses princípios foram um resultado das teorias da psicologia cognitiva, psicologia das ações e experimentos realizados.

A noção de que grande parte dos objetos não é bem desenhada é que motivou Norman e outros pesquisadores a se debruçarem sobre o tema. O pressuposto é que as pessoas têm grande dificuldade de usar produtos corriqueiros, como retirar o plástico da capa de um CD, usar um aparelho doméstico, entre outros.

Além disso, ainda existe a chamada “síndrome do painel de avião”, em que grande parte das funções de um produto fica subutilizada e ainda torna-se impossível entender para que servem tantas teclas.

Princípio 1 - Visibilidade

O usuário deve saber dizer o estado em que se encontra um dispositivo e as opções de ação a partir daquele estado apenas olhando para ele. O sistema deve mostrar todas as possibilidades de interação naquele contexto.

Figura 2 - Local notification: o usuário consegue saber que existe uma notificação no Facebook.



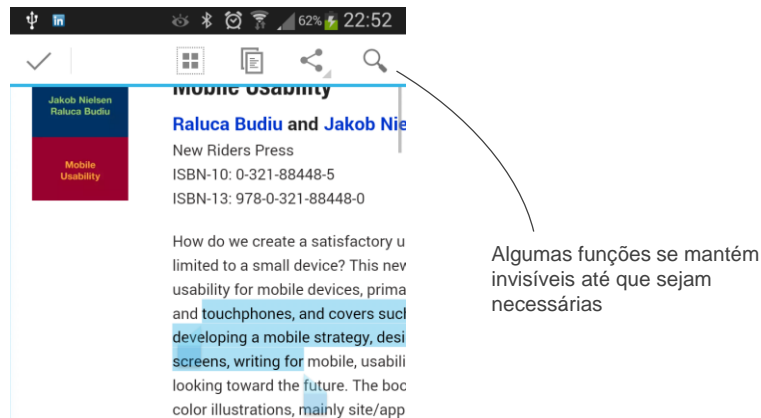
Às vezes não sabemos onde estão os elementos de interação. Os problemas com a Visibilidade aparecem quando “não podemos ver” os elementos de interação do produto.

No exemplo abaixo, temos que adivinhar onde colocar as mãos. Torneiras e botões visíveis foram substituídos por “zonas ativas” que são invisíveis e ambíguas.

Figura 3 - Torneira sem botão visível.



Em alguns casos, um bom exemplo de visibilidade é aquele que oculta algumas funções quando não estão no contexto apropriado.

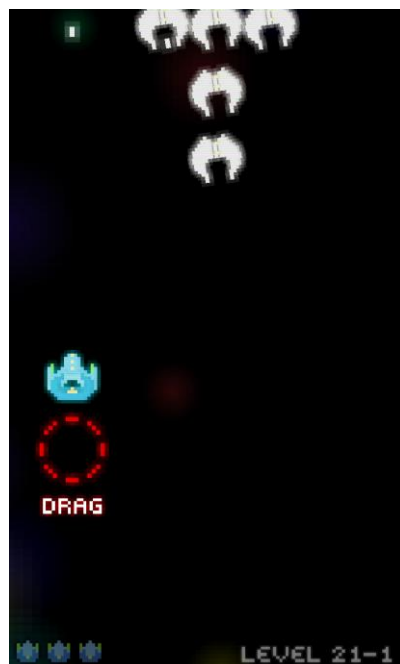


Princípio 2 - Feedback

Na sua interação com o mundo, as pessoas necessitam continuamente perceber, interpretar e avaliar o resultado de suas ações. A interface deve enviar de volta ao usuário uma informação sobre o que foi efetivamente realizado. O usuário deve receber um feedback contínuo e completo sobre os resultados de suas ações.

Toda ação (usuário) requer uma reação (sistema). No jogo da imagem abaixo, ao acertar uma nave inimiga, o sistema emite um feedback sonoro e tátil.

Figura 4 - Voxel Invaders.



Princípio 3 - Restrições

A restrição, neste caso, tem um sentido maior de prover segurança do que limitar as funcionalidades. Isso quer dizer que, em alguns casos, é necessário restringir a interação a fim de não deixar o usuário cometer erros. A máscara nos campos de preenchimento de CPF que impedem a inserção de letras é um ótimo exemplo de restrição.

Neste aplicativo de controle financeiro, a inserção de um lançamento é restringida por uma sequência única de preenchimento.

Figura 5 - Garoupa finanças (Android).

< Lançamento

1 2 3 4 5 6

2. Data

Informe a data do lançamento:

28	Mar	2013
29	Abr	2014
30	Mai	2015

Princípio 4 - Mapeamento

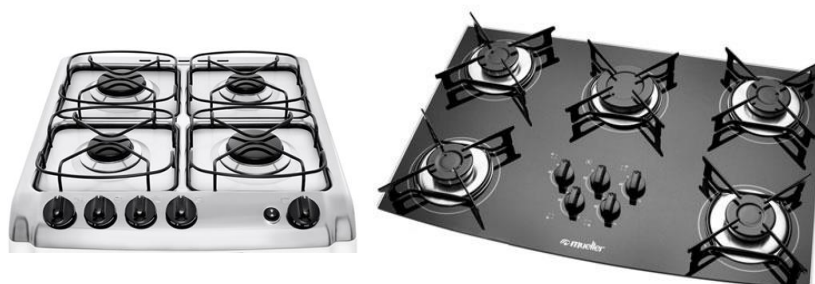
Refere-se à determinação do relacionamento entre as ações e os resultados, entre os controles e seus efeitos e entre o estado do sistema e o que é visível. Bons mapeamentos tiram vantagens de analogias físicas e padrões culturais.

Na figura abaixo, o fogão da direita possui uma melhor solução de mapeamento, já que existe uma correspondência direta entre o controle e a trempe.

Figura 6 - Fogões e soluções de mapeamento

Figura 7 - Fogões e soluções de mapeamento

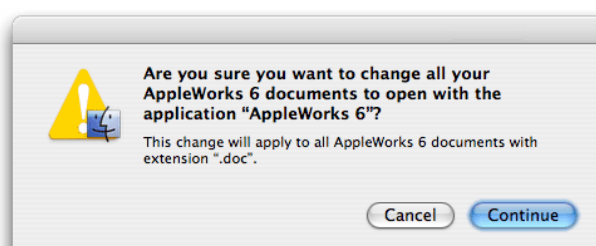
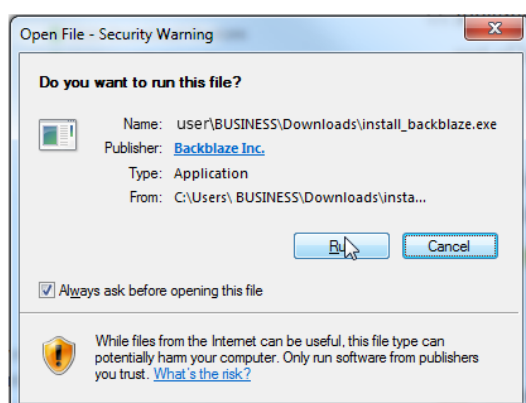
Figura 6 - Fogões e soluções de mapeamento.



Princípio 5 - Consistência

É essencial que um mesmo sistema mantenha uma consistência, seja em termos funcionais ou estéticos. Assim, o usuário não dispensará tanto tempo com a aprendizagem do seu funcionamento. As possibilidades de erro são menores e, conseqüentemente o uso do sistema se torna mais eficaz.

Figura 8 - Consistência interna. Em sistemas Windows, o botão cancelar sempre fica à direita, enquanto no MacOS esse botão fica à esquerda.



Há ainda situações de consistência externa, em que vários sistemas apresentam mecanismos de funcionamento idênticos, como no exemplo abaixo.

Figura 9 - Consistência externa. Diferentes aplicações usam um mesmo padrão de menu de ferramentas.



Princípio 6 - Affordance

Os elementos de interação devem “falar por si” sobre como utilizá-los. São propriedades nos elementos que nos permitem perceber como é que se usa, como por exemplo, um botão que tem “cara de botão” como se nos convidasse a clicá-lo. Ou uma maçaneta horizontal que nos mostra como abrir a porta, sem mesmo colocarmos a mão nela.

Figura 10 - Bom exemplo de affordance. Tanto a maçaneta quanto a tesoura nos dão dicas sobre como usá-las mesmo só olhando



Figura 11 - Bom exemplo de affordance à esquerda, onde a interface já indica a possibilidade do drag and drop. Na direita, um exemplo ruim de affordance pois os labels dos campos se parecem com botões.



Capítulo 2. Paradigmas de interação

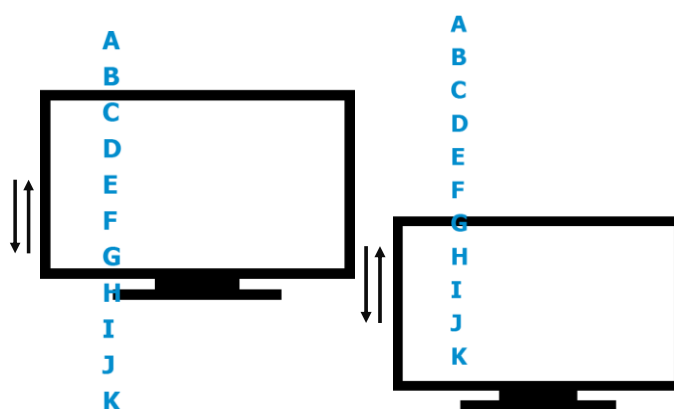
Como dito, os dispositivos móveis são os PCs do século XXI. Mas não dá para negar que o paradigma de uso anterior – o desktop – ainda é muito forte. Haja visto que grande parte dos projetistas de interface para dispositivos móveis tem uma tendência em transpor esse paradigma para as novas plataformas.

O paradigma desktop

Em 1984, Jakob Nielsen, considerado o “pai da usabilidade”, realizou um estudo na IBM, para ver qual paradigma se encaixava melhor no modelo mental das pessoas que iriam utilizar os terminais 3270 (terminais burros, como eram conhecidos aqui).

Nessa altura você deve estar imaginando que a resposta foi “seta para baixo”. Exato! É o padrão usado até hoje.

Figura 12 - Em desktop, nós movemos a tela em direção ao conteúdo.



No desktop, nós movemos a tela em direção ao conteúdo. Recentemente, a Apple alterou um pouco esse modelo nos Mac OS. Se usarmos o scroll do Magic Mouse da Apple, o conteúdo se moverá na direção oposta à do Windows. Em

mobile, o paradigma é o oposto, ou seja, é o conteúdo que se move dentro da tela (como no Magic Mouse da Apple).

E quando se fala em mobile, devemos entender que não se trata apenas de smartphones, mas também tablets, games consoles, totens e outros. Pode parecer óbvio, mas esse é um fator constantemente negligenciado pelos projetistas de interfaces mobile.

Figura 13 - Em mobile, o paradigma é o oposto: É o conteúdo que é descolado dentro da tela.



Principais diferenças entre desktop e mobile

Muitos esquecem que não há mouse com scroll, nem teclado físico na maioria dos dispositivos móveis e, por isso, deixam de reservar os espaços necessários para o toque, o scroll e outros movimentos. Por causa disso, muitas vezes os usuário acabam, sem querer, acionando um comando quando queria apenas subir o conteúdo. A seguir, veremos outras diferenças cruciais.

Não há como prever onde os usuários irão interagir com seu aplicativo

Os notebooks impulsionaram a história da computação móvel, mas, agora, as pessoas levam a computação na palma de suas mãos e esperam que tudo funcione no lugar e na hora que elas desejam.

Nem todo mundo está numa grande metrópole

No Brasil, mais de 15% da população vive na área rural. Sem contar as cidades de médio porte. Ou seja, nem sempre as pessoas estarão dentro de um metrô ou naquelas ruas modernas e bem pavimentadas como nos comerciais de TV. Então, não se assuste ao saber que seu aplicativo foi acessado em cima de um trator.

Sua interface deve atender às necessidades dos usuários que mudam dependendo de onde estão

Um clássico estudo feito pelo Google descobriu que um mesmo usuário pode ter até 3 perfis diferentes dependendo do contexto:

- a) “O repetitivo” – checa a mesma informação várias vezes como status do Facebook, Instagram etc.
- b) “O entediado”: Geralmente é quando precisa de um passa-tempo, como esperando uma conexão em um aeroporto, dentro de uma viagem longa, etc. Esse perfil é muito semelhante ao do internauta comum.
- c) “O apressado”: Quer encontrar o que precisa com urgência, como uma agência bancária ou o trajeto para o aeroporto. Nesse caso, utilizar as APIs de georreferenciamento são bastante úteis.

As interfaces clássicas de Web podem não funcionar bem

Lembre-se: não só a tela é menor, mas a conexão e o contexto de uso são bem diferentes em mobile. Por isso, privilegie a informação relevante e faça com que o layout seja adaptado ao dispositivo. Para isso existem as técnicas de Responsive Design. Para saber mais, acesse: <http://goo.gl/OJv50o>.

Não existe interface padrão para **todos os dispositivos mobile**

Symbian, BlackBerry, Firefox OS, Windows Mobile, iOS e Android são bastante distintos entre si em termos de interface. Cada um deles tem sua particularidade e seria um erro criar a mesma interface para todos.

Não há muito espaço para expor sua marca

Apesar de a maioria dos clientes caírem na tentação de pedir para aumentar o logotipo, em mobile, isso deve ser feito com bastante cuidado. Nesse caso, o melhor é seguir a identidade visual da empresa ou produto.

Tamanho é documento

Se em laptops e desktops você pode usar barras laterais, banners e menus extensos, lembre-se que, agora, você deve dar importância à tarefa a ser executada.

Dê preferência a uma tarefa-chave por tela. Pergunte a si mesmo: “Quais são as ações e os comandos essenciais para a execução de uma tarefa, e quais deles eu posso deixar para uma tela secundária?”

Não deixe que sua interface exija muita digitação de texto

Digitar em smartphones já é difícil. Ainda mais quando estamos em movimento. Por isso, faça com que sua interface possua comandos frequentes à vista. Seguem abaixo algumas dicas que podem ser úteis:

- Adote padrões inteligentes e orientados ao objetivo da tarefa
- Deixe algumas configurações como default para o usuário não ter que configurar tudo.
- Ofereça ao usuário uma lista de opções de escolha em vez de ter que digitá-las

Desenhar para diferentes telas e densidades pode ser doloroso

Lembre-se que não só os tamanhos são diferentes, mas a densidade de pixels também varia. Dessa forma, um quadrado de 80px de largura pode aparecer maior ou menor dependendo do aparelho. Densidade é a quantidade de pixels em uma polegada linear de tela. Por exemplo, o iPad Air e o Galaxy Tab 3 possuem dimensões semelhantes. Mas o primeiro tem uma densidade de 264 ppi (pontos por polegada) enquanto o segundo tem apenas 149 ppi. Isso significa que o quadrado acima visto nesses dois aparelhos terá tamanhos diferentes, ou seja, no iPad o quadrado terá quase a metade do tamanho que no Galaxy. Neste link você encontra uma tabela completa de tamanhos e densidades de vários aparelhos: <http://goo.gl/OThRIV>.

Usuários mobile não são tão fiéis

Devido a imensidão de opções disponíveis, usuários desistem facilmente de um aplicativo que não lhe agrada em algum aspecto, seja funcional, estético ou outro qualquer que afete a sua experiência de uso. Por isso, tome cuidado em fazer uma interface que agradável, e com uma experiência rica.

Padrões de interação mobile

Veja a seguir alguns padrões de interação que são típicos de dispositivos mobile, sobretudo os de tela sensível ao toque (*touch screen*).

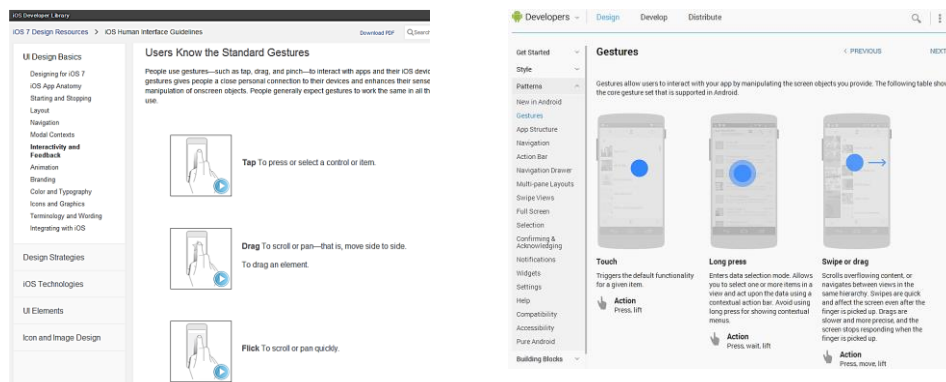
Gesture

Um gesto é, basicamente, um movimento de um dedo, mão ou outra parte do corpo para expressar uma ideia, desejo, ou acionar um comando.

Inicialmente, alguns dispositivos só podiam processar o toque de um dígito por vez, mas hoje em dia, os dispositivos multitoque podem perceber gestos feitos por um dedo ou mais simultaneamente.

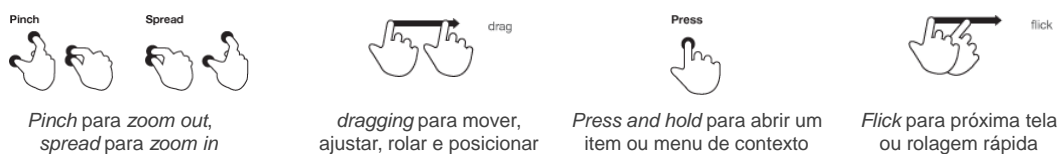
Cada gesture recebe um nome específico. O problema é que esse nome varia em diferentes plataformas (iOS versus Android versus Windows Phone versus Palma versus Take seu pick). Por isso, é recomendável ler a documentação de Design Pattern de cada plataforma.

Figura 14 - Documentação sobre gestures disponibilizada pelo iOS e Android em seus respectivos sites de desenvolvimento.



Luke Wroblewski criou um excelente guia de referência de Gestures que está disponível em PDF no link abaixo: <http://goo.gl/QZWfql>. Em seu site ele ilustra, explica gestos fundamentais, bem como a forma como eles diferem em todas as plataformas. Veja alguns exemplos:

Figura 15 - Alguns gestos (gestures) ilustrados por Luke Wroblewski



Feedback

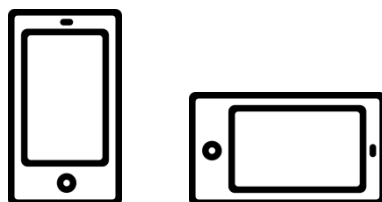
São ações e métodos que atraem a atenção do usuário para uma informação importante. Algumas orientações:

- a) Alertas e demais mensagens devem ser usadas com moderação para que o usuário saiba que se trata de uma informação crítica. Exagerar nesse quesito faz com que o usuário não dê importância aos alertas.
- b) As mensagens devem ser claras e objetivas explicando o que aconteceu, a causa e oferecer uma saída com comandos e botões claros.
- c) As notificações devem ser sucintas e informativas sem interferir no que o usuário está fazendo de modo que ele possa responder à notificação ou descartá-la com facilidade.
- d) Não quebre o fluxo de interação do usuário para exibir alertas.

Orientação:

Trata-se da posição do dispositivo em relação ao solo. Pode ser retrato ou paisagem. Essa posição é detectada pelo acelerômetro, disponível em quase todos os aparelhos atuais.

Figura 16 - Orientações: Retrato (Portrait) à esquerda e Paisagem (Landscape) à direita.



No projeto de sua interface, considere as diferenças de orientação a fim de propor novas possibilidades de interação. Em sites responsivos, é possível detectar com CSS qual é a orientação do dispositivo.

Figura 17 - Trecho de CSS para detectar a orientação do dispositivo.

```

1  /* Portrait */
2  @media screen and (orientation:portrait) {
3      /* Portrait styles */
4  }
5  /* Landscape */
6  @media screen and (orientation:landscape) {
7      /* Landscape styles */
8  }

```

Tente manter a localização do usuário na tela para que ele não tenha que rolar até o ponto em que estava. Se o contexto permitir, adicione conteúdos diferentes ou novas funcionalidades, como no caso do aplicativo da seguradora ING:

Figura 18 - Iniciativa da ING em prover conteúdo adicional em diferentes orientações.



Capítulo 3. Telas, resoluções, densidades e unidades de medida

A partir de 2008, com a chegada do Android da Google, diversos fabricantes no mundo inteiro, principalmente no oriente, apostaram suas fichas em um sistema estável, endossado por uma grande companhia (Google) e o principal, Opensource. Desde então, o Android proporcionou a popularização de “smart devices” em todo mundo, e com preços a partir de 50 dólares.

A diversidade no ecossistema mobile

O sucesso do Android trouxe um novo desafio para os desenvolvedores de aplicativos e de Web, sobretudo para os smartphones. E o mais surpreendente é que, apenas um ano atrás, a Apple havia lançado uma tendência que é seguida por muitos até hoje: o iPhone. Mas essa diversidade é muito maior do que pensamos. Veja os exemplos a seguir.

Figura 19 - Smartphone de duas telas criado pela operadora de telefonia russa Yota.

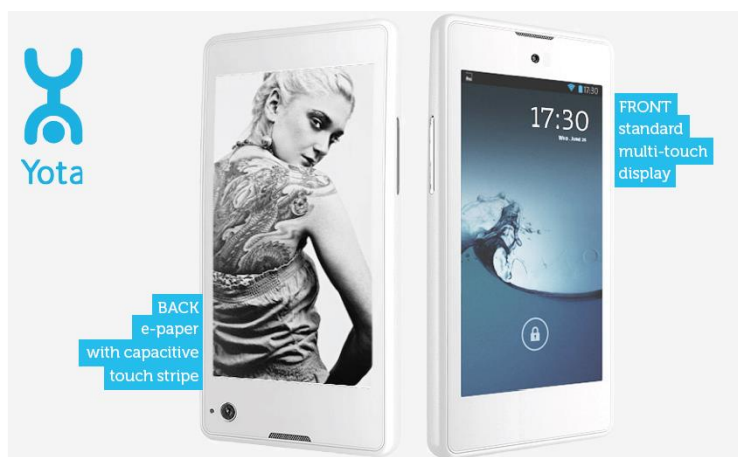
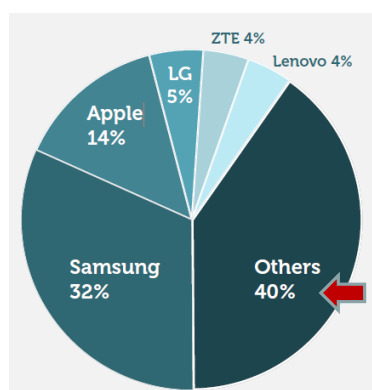


Figura 20 - Tablet da marca Nabi criado para fins educativos.



Embora muitos consumidores ainda comprem dispositivos projetados, desenvolvidos e comercializados por grandes empresas, isso pode não ser o caso daqui para frente. Se os padrões atuais continuarem, consumidores ao redor do mundo irão cada vez mais ter a mesma probabilidade de adquirir dispositivos desenvolvidos por "outros" fabricantes.

Figura 21 - Venda de Smartphones para usuários finais no mundo por fabricante em 2013¹.



¹ <http://www.gartner.com/newsroom/id/2573415>

Com essa diversidade podemos esperar uma vasta gama de tamanhos e densidades diferentes. O diretor da empresa Open Signal disse uma vez:

“Em poucos meses de 2013, 11.868 diferentes aparelhos baixaram nosso aplicativo. Em 2012 eram apenas 3.997²”.

Tamanhos e densidades

Recapitulando, densidade de tela é a quantidade de pixels existente em um determinado comprimento de tela física.

A medida padrão de densidade é o **ppi** = *pixel per inch* ou pixel por polegada.

Na grande maioria dos computadores desktop, a densidade dos monitores é de 72 ppi. Ou seja, em uma polegada do seu monitor, cabem 72 pixels.

Nos computadores da Apple munidos com a tecnologia Retina Display, a densidade chega a 288 ppi.

Há uma tendência por parte da indústria de smartphones, tablets e TVs em aumentar a densidade das telas para valores maiores que **300 ppi**. Isso se deve ao fato de que, num futuro próximo, o padrão **full-HD** cuja resolução máxima é de 1920 por 1080 pixels seja substituído pelo padrão **4K** que chega a ter 4 vezes mais resolução que o anterior. O iPhone 4 e o Samsung Galaxy SIII foram um dos primeiros a ultrapassarem a marca dos 300 ppi.

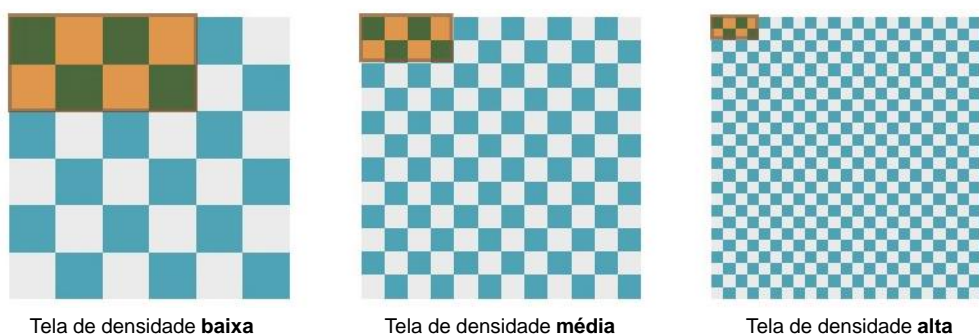
² Open Signal Device Report - <http://opensignal.com/reports/fragmentation-2013/>

Figura 22 - Diferenças de densidades de tela entre os modelos da linha iPhone.

				
iPhone	iPhone 3G	iPhone 3GS	iPhone 4	iPhone 5
320x480 pixels 3.5-inch	320x480 pixels 3.5-inch	320x480 pixels 3.5-inch	640x960 pixels 3.5-inch Retina	640x960 pixels 4-inch Retina
160 ppi	160 ppi	160 ppi	320 ppi	320 ppi

Essa diferença entre densidades e tamanhos de tela pode ser problemática para nós desenvolvedores. Em telas com diferentes densidades, um elemento visual com a mesma medida em pixels, terá tamanhos “físicos” diferentes. Veja o exemplo abaixo.

Figura 23 - Diagrama que mostra a densidade de tela.



O botão amarelo possui a mesma medida em pixels (4 x 2), mas aparece em tamanhos diferentes, em telas com outras densidades maiores.

Como vimos anteriormente, a grande diversidade de dispositivos Android torna essa tarefa mais difícil. No caso do iPhone, por ser uma tecnologia fechada, esse problema é minimizado.

Em sete anos no mercado, a Apple lançou apenas cinco modelos do seu smartphone, sendo que apenas na penúltima versão é que a densidade foi aumentada.

Figura 24 - Cinco modelos e apenas duas densidades diferentes.

				
iPhone	iPhone 3G	iPhone 3GS	iPhone 4	iPhone 5
320x480 pixels 3.5-inch	320x480 pixels 3.5-inch	320x480 pixels 3.5-inch	640x960 pixels 3.5-inch Retina	640x960 pixels 4-inch Retina
160 ppi	160 ppi	160 ppi	320 ppi	320 ppi

No caso da família iPad, apenas a última geração do iPad Retina e do iPad mini é que possuem 320 ppi. As demais, apenas 130 ppi.

Figura 25 - A tela Retina é que possui densidade maior que 300 ppi.

		
iPad mini	iPad 2	iPad with Retina display
320 ppi	130 ppi	320 ppi

Mas quando o assunto é Android, o desafio aumenta.

Figura 26 - Variedade de dispositivos Android.



Por causa da grande quantidade de fabricantes de telas e dispositivos com Android, o Google criou um mecanismo que permite com que os desenvolvedores possam criar layouts usando uma nova medida que independe da densidade da tela - o **dp** (density independent pixel).

O dp é uma espécie de “pixel virtual” para designar dimensões e posicionamento.

Como o dp funciona?

Elementos gráficos em **dp** são escalonados automaticamente para o tamanho apropriado, independentemente do tamanho ou da densidade da tela. Veja abaixo o mesmo botão mostrado anteriormente, só que dessa vez, com as medidas em **dp** e não mais em pixels.

Figura 27 - Dica: Em Android, 1dp corresponde a 1px em uma tela de 160ppi.



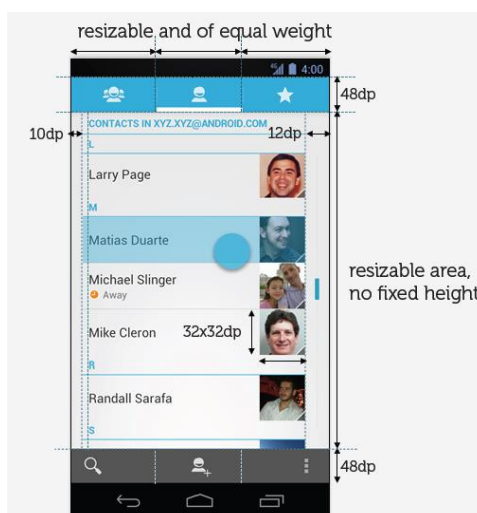
Desenhando para múltiplas telas

A seguir, você verá algumas orientações que o auxiliarão no desenho para múltiplas telas com diferentes densidades, fato que é muito mais desafiador quando se desenvolve para Android.

Priorize a flexibilidade. Assim, como em interfaces Web, não se deve especificar dimensões para elementos que são redimensionados. O Android permite que você defina como os elementos são escalonados e quais são fixos.

Veja abaixo o que recomenda a documentação do Android Developers.

Figura 28 - Definição dos elementos que têm dimensões fixas e variáveis.³

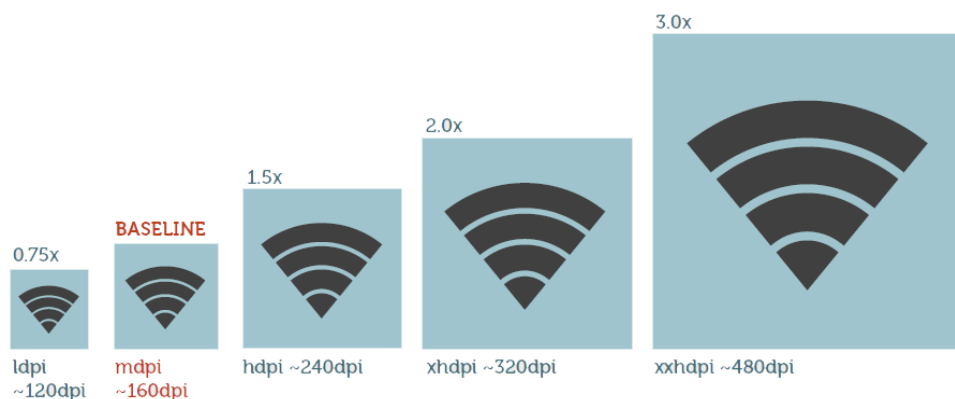


E quanto às imagens?

Na hora de fazer seu layout para Android, lembre-se que, cada imagem (ícone, botão, símbolo) deverá ser fornecida em tamanhos diferentes a fim de suprir todas as densidades possíveis. Veja o exemplo do ícone abaixo.

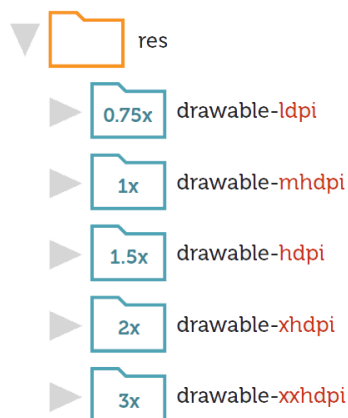
³ <http://developer.android.com/>

Figura 29 - O mesmo ícone em diferentes tamanhos em um mesmo aplicativo.



Cada imagem (de preferência em PNG) deverá ficar na sua respectiva pasta de acordo com seu tamanho. Dessa forma, depois de compilado, o aplicativo usa a imagem apropriada, dependendo do aparelho.

Figura 30 - Cada ícone deverá ser colocado em sua respectiva pasta, definida pelo seu tamanho.

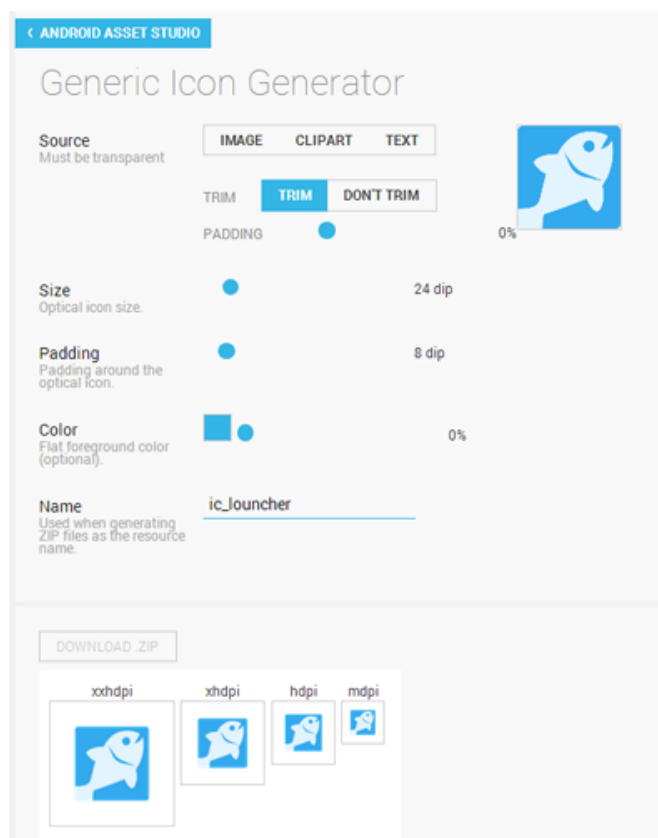


Mas isso não dá muito trabalho? Dá sim.

Para ajudar nesse processo, o Android criou uma ferramenta Web chamada **Android Asset Studio**.⁴

Com ela, basta você fazer o upload da sua imagem na resolução mais alta e, em seguida, baixar um arquivo Zip com todas as outras imagens recortadas e com as pastas já criadas e nomeadas.

Figura 31 - Snapshot da tela do Android Asset Studio.



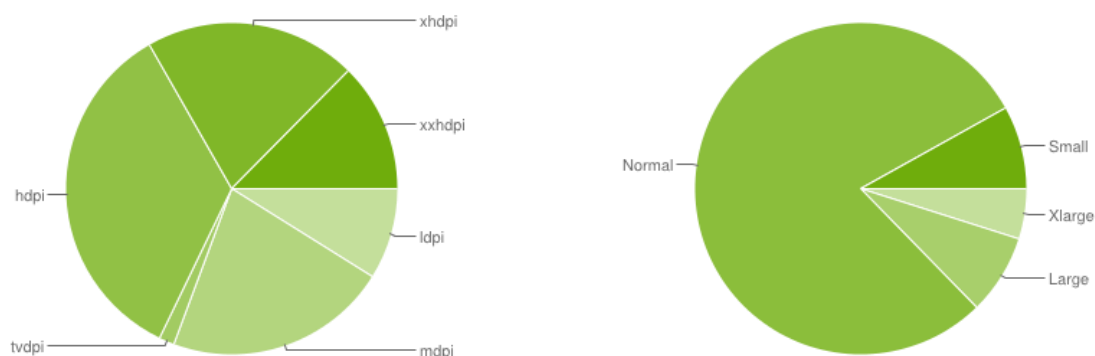
Mas eu devo atender a todos os requisitos de cada fabricante? Depende.

Frequentemente, o Android publica algumas estatísticas sobre vários aspectos, entre eles, os tamanhos e a densidade das telas. Por isso, é importante

⁴ <http://android-ui-utils.googlecode.com/hg/asset-studio/dist/index.html>

levar em consideração esses dados a fim de tomar a melhor decisão. Os gráficos abaixo são de 2014⁵.

Figura 32 - Estatísticas sobre telas e densidades em dispositivos Android.



E para websites e aplicativos web?

Se você está desenvolvendo um website adaptado para dispositivos móveis, certifique-se de que, não só as imagens, mas o layout e o conteúdo estejam adaptados. Veja o diagrama abaixo:

Figura 33 - Diagrama que se adapta à orientação e tamanho de tela.

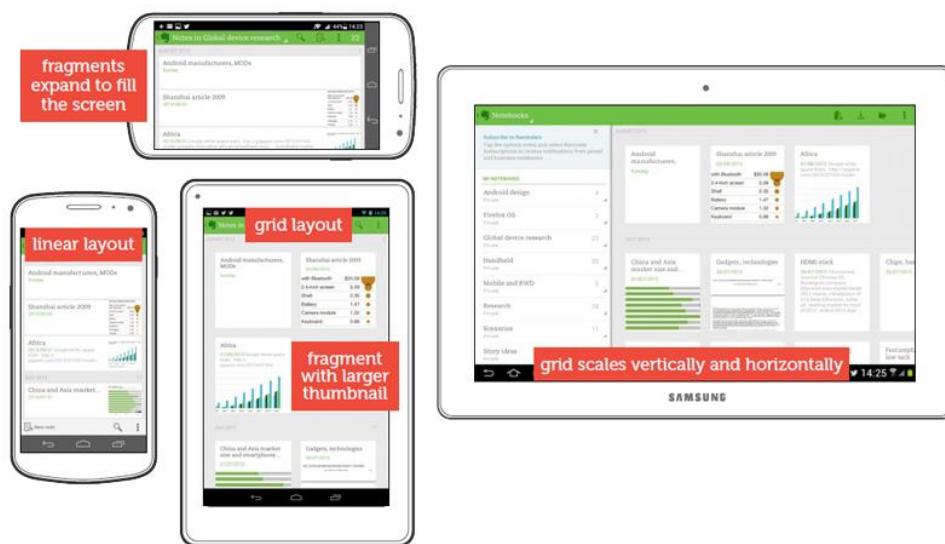


Diferentes tamanhos de tela podem ser uma oportunidade para combinar diferentes porções de conteúdo. Para isso, basta pensar na melhor maneira de adaptar o seu layout para cada tamanho.

⁵ <http://developer.android.com/about/dashboards/index.html>

Diferentes tamanhos de tela podem ser uma oportunidade para combinar diferentes porções de conteúdo. Para isso, basta pensar na melhor maneira de adaptar o seu layout para cada tamanho.

Figura 34 - O Evernote utiliza bastante as adaptações de layout para diferentes tamanhos.



Em páginas web, o uso de imagens deve receber um tratamento especial. Lembre-se que:

- As imagens devem ser também responsivas e, de preferência, ocupar toda a largura da tela
- Fazer um tamanho único de imagem em resolução maior e deixar que o dispositivo adapte-a a seu tamanho não funciona muito bem, por dois motivos:
 1. Esse escalonamento nem sempre se dá de forma apropriada e os pixels da imagem ficam truncados;
 2. O browser mobile carrega **todas as imagens antes mesmo do HTML e CSS**. Por isso, usar imagens maiores acaba por penalizar os usuário de conexões 3G.

E o que fazer neste caso?

Para isso, existem scripts que cuidam de carregar as imagens corretas, em tamanhos diferentes, de acordo com o dispositivo. Um deles é o **HISRC** – Um script que depende da biblioteca jQuery e faz o trabalho de carregar a imagem correta. Basta que as diferentes imagens sejam referenciadas no HTML conforme a figura abaixo:

```

```

Para saber mais, acesse: <https://github.com/teleject/hisrc>

Algumas ferramentas úteis

[Android] Devices and Displays

Uma seção do portal Android Developers que explica as variações de telas e densidades dos dispositivos Android, além de conteúdos relacionados.

<http://developer.android.com/design/style/devices-displays.html>

[iOS] LiveView for iPhone & iPad

LiveView é uma ferramenta de visualização remota da tela para ajudar os designers a visualizarem seu layout diretamente no dispositivo.

<http://zambetti.com/projects/liveview/>

[Todos] Lista de todas as telas e suas densidades

É uma lista baseada em Wiki com todas as telas e densidades separadas por fabricante, modelo e versão.

http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density

[Android] Device Frame Generator

Ferramenta útil para ver como o seu layout aparece em alguns aparelhos. Basta salvar seu layout em PNG e arrastar o arquivo em cima do aparelho desejado para ver como fica.

<http://android-ui-utils.googlecode.com/hg/asset-studio/dist/device-frames.html>

[WEB Mobile] **HISRC**

Script para carregar imagens responsivas.

<https://github.com/teleject/hisrc>

[WEB Mobile] **ADAPTIVE IMAGES**

Script **server side** para carregar imagens responsivas. Depende de um servidor Apache e PHP 5+

<http://adaptive-images.com/>

Capítulo 4. Briefing de projetos mobile

É bastante comum sermos procurados por uma pessoa ou empresa dizendo:

“Olha, eu tenho uma ideia para um aplicativo. Quanto custa e quanto tempo leva pra fazer?”

Responder a essa pergunta instintivamente pode ser uma grande armadilha. Basear-se em estimativas para “chutar” um valor e um prazo pode levá-lo ao prejuízo. Portanto, é extremamente importante que você faça um planejamento baseando-se nas etapas que você ou sua empresa adotam. Uma das melhores maneiras de começar esse planejamento é criando um documento chamado **briefing**. Trata-se de um documento que contempla, de forma abrangente, todas as características de um projeto.

O briefing ainda ajuda a dimensionar melhor o projeto e seus custos, além de esclarecer todas as nuances tanto para a equipe, quanto para o cliente. Este, por sua vez, poderá ter uma noção muito mais fiel de quais são os custos, etapas e o que mais envolve a criação e desenvolvimento de um produto mobile.

A construção de um briefing é um evento que vai esclarecendo passo a passo, para ambas as partes, quão complexo é o produto.

Etapas típicas de um projeto mobile

A seguir, um fluxo de etapas de um típico projeto mobile. Existem várias outras abordagens no mercado e na literatura, com os quais você pode estar mais familiarizado, mas essas são apenas algumas etapas mais típicas e que podem ser adaptadas em qualquer cenário.

1- Descoberta:

Aqui se faz um levantamento dos objetivos gerais do projeto, usuários, stakeholders e métricas de sucesso.

2- Planejamento estratégico:

Aqui nós identificamos os usuários finais, suas necessidades para conduzir o projeto centrado no usuário. Recomenda-se fazer uma análise SWOT a fim de determinar se a estratégia está correta. Por fim, um cronograma com as estimativas de tempo e entrega.

3- Design:

Aqui começam os trabalhos de prototipagem, testes com usuários e layout. Uma vez que o design e a identidade visual estão concluídos, começa o desenvolvimento.

4- Desenvolvimento:

Esta é a fase em que são feitas as implementações nas linguagens e plataformas escolhidas. Hoje em dia, os processos ágeis dominam o mercado e é essa etapa que é subdividida em *sprints* com entregas iterativas.

5- Lançamento:

Com o desenvolvimento e os testes concluídos, é hora de lançar o produto. Esse lançamento pode acontecer em um ambiente interno ou para usuários testadores.

Elementos do briefing

1- Introdução:

Aqui você deve perguntar ao cliente sobre o que o motivou a criar um aplicativo. Tente descobrir se é apenas uma maneira que entrar no mercado ou se é para atender a uma necessidade real e registre o que descobriu. Uma boa maneira de se descobrir isso é fazendo uma análise SWOT (um gráfico em que são mapeadas as forças, fraquezas, oportunidades e ameaças para saber mais sobre isso acesse: <http://goo.gl/a0E2jW>).

2- Objetivos:

É, basicamente, mapear o problema e os objetivos de uso.

O que os futuros usuários alcançarão com seu produto? Que vantagens eles terão? O produto resolve algum problema específico? Você consegue descrevê-lo?

Veja se os objetivos são: a) específicos b) mensuráveis c) atingíveis e d) oportunos.

Isso irá ajudar o planejamento de marketing quando do lançamento do produto, já que nenhum aplicativo se “vende” sozinho.

3- Contexto de uso:

Baseando-se nos objetivos, você consegue identificar como e onde as pessoas utilizarão seu produto?

Por exemplo, se for um aplicativo para corridas, é presumível que este será usado na rua, sob o sol e em movimento. Se for um aplicativo para gestão de clínicas, podemos pensar que este será usado em ambientes internos.

Com este contexto mapeado, teremos uma ótima fonte de consulta tanto para o design, quanto para o planejamento dos testes.

4- MVP - Mínimo produto viável (*Minimal Viable Product*):

É comum que, durante a reunião de briefing, você e o cliente se empolguem em listar várias e várias funcionalidades diferentes para seu produto.

Entretanto, é importante definir quais são as funcionalidades que deverão **atingir os objetivos principais** e **resolver os problemas-chave** definidos anteriormente **na data de lançamento**.

Deixe o MVP bem esclarecido para todos a fim de não gerar falsas expectativas no cliente e nem desmotivar a equipe com o acréscimo de trabalho durante o projeto.

As demais funcionalidades podem ser adicionadas em versões posteriores. E, por incrível que pareça, usuários gostam de perceber que o aplicativo está em evolução e com novas funcionalidades sempre surgindo. Mas é importante definir com o cliente como essas novas funcionalidades serão implementadas.

5- Dispositivos, plataformas e pessoas:

A escolha da plataforma influencia bastante no custo e no prazo de desenvolvimento.

No mercado atual, o que podemos ver é basicamente uma corrida com apenas 2 carros: Apple e Android.

Mesmo que você opte por construir “apenas” para essas duas plataformas, fique atento para não cair em um erro muito comum: planejar o lançamento do aplicativo nas duas plataformas ao mesmo tempo. O ideal é que o produto seja desenvolvido para uma plataforma por vez, por dois motivos:

- a. Quaisquer problemas que surgiram desde o desenvolvimento da primeira plataforma serão evitados para o segundo.
- b. Os feedbacks dos usuários da primeira plataforma podem ser úteis para o desenvolvimento da segunda que poderá ir ao ar com atualizações decorrentes da plataforma anterior.

- c. O risco é menor já que todas as experimentações já foram feitas anteriormente.

Então qual plataforma adotar primeiro? Pergunte às pessoas! Faça pesquisas online usando redes sociais, ferramentas de entrevistas como Survey Monkey ou PollDaddy e descubra informações relevantes que poderão nortear sua estratégia de negócio.

Já escolhi a plataforma, agora, qual versão adotar? Converse com seu desenvolvedor.

Ele saberá melhor quais recursos estão disponíveis em cada versão. Por exemplo: seu aplicativo iOS precisa detectar movimento dos usuários ou o número de passos que eles dão? Se assim for, isso só é possível na 2ª geração do iPhone 5s/iPad Air e iPad Mini, que utiliza novo coprocessador de movimento M7 da Apple. No caso do Android você precisa atender ao público do Android 2.3 ou é melhor começar do 4.0 (veja as estatísticas em <http://developer.android.com/about/dashboards/index.html>)?

Considerar iOS6 e acima ou você vai apenas iOS7? Essas considerações são importantes e afetam custos e prazos. Lembre-se que a partir de 01 de fevereiro de 2014, todos os aplicativos iOS devem ser otimizados para iOS7.

6- Conteúdo:

Este é outro fator muitas vezes negligenciado. Em caso de um web app, ou outro aplicativo de exibição de conteúdo, você deve descobrir quem irá prover o conteúdo editorial? Você, a equipe de marketing do cliente ou um terceiro?

Outra questão importante é se o conteúdo será atualizado. Se sim, como? Com que frequência?

Esse conteúdo será “puxado” de algum site ou webservice?

7- Layouts preliminares:

Não pense que basta mudar a orientação do aparelho de retrato para paisagem e que tudo irá automaticamente se encaixar de forma perfeita. É preciso desenhar e especificar essa parte também.

Duas das ferramentas mais eficazes para esta etapa são **papel e lápis**. Rabisque algumas ideias no papel e vá aprimorando-as depois.

Não se preocupe em fazer um belo desenho ou um protótipo de alta fidelidade. A ideia aqui é gerar esboços preliminares que irão ajudar a equipe de design.

Se quiser, você poderá usar uma ferramenta bastante útil chamada POP (*Prototype On Paper*). É um aplicativo (Android e iOS) no qual você pode escanear seus desenhos e colocar links em cada elemento desenhado e até transições entre telas diferentes.

8- Banco de dados e hospedagem:

Essa é uma questão crítica que, se não for tratada com cuidado, poderá arruinar todo o projeto.

Na medida do possível, tente estruturar o modelo de banco de dados para seu produto ou inclua amostras do banco de dados atual que seu cliente já dispõe para que seus desenvolvedores possam analisá-lo.

Dependendo do estado e da qualidade da estrutura do banco, pode ser que você tenha que incluir os custos da remodelagem ou mesmo terceirizar essa parte.

Deixe claro de quem é a propriedade do banco. Você, o cliente ou a empresa terceirizada.

Caso seu aplicativo precise puxar ou alimentar um banco externo, especifique se a hospedagem (*hosting*) contratada que irá abrigar o banco de dados.

Mesmo a hospedagem sendo própria (fornecida por você) ou contratada, anote todas as informações técnicas e repasse-as para sua equipe de desenvolvimento.

9- Política de suporte:

Se você ficar responsável por prestar suporte para os usuários, certifique-se de que você conta com uma estrutura para

- a. Aceitar feedbacks através de determinados canais (telefone, chat, e-mail, etc.)
- b. Ouvir e responder aos feedbacks

Lembre-se que toda e qualquer definição nesse quesito deverá entrar na sua planilha de custos.

Itens de suporte ainda incluem correção de bugs, hospedagem e compatibilidade com novas versões de software.

10- Lançamento:

Como dito, aplicativos não se promovem sozinhos. Foi-se o tempo em que os usuários ficavam ansiosos para baixar um aplicativo novo assim que a aparecia na loja. Fazer o marketing do aplicativo não é uma tarefa trivial. É preciso uma estratégia de curto, médio e longo prazo para que o marketing realmente funcione.

Dessa forma, é preciso que o briefing mostre quais são os canais de marketing que serão usados para anunciar seu aplicativo.

Esse é um ótimo exercício para ver como criar um produto sustentável e de sucesso.

11- Prazos:

Leve em consideração todos os requisitos técnicos, escopo e equipe antes de especificar um prazo.

Não planeje o fim do projeto para uma data específica determinada por outros fatores como um grande evento ou uma festa de abertura.

Caso o prazo final seja inflexível, nesse caso, é melhor renegociar o escopo, pautando-se na sua capacidade produtiva.

Conclusão

O documento de briefing não acaba aqui.

Uma das melhores coisas sobre esse artefato é que ele é o principal ponto de partida para novas discussões e descobertas.

Uma relação saudável entre cliente e desenvolvedor começa com um briefing bem escrito e com todas as funcionalidades descritas e programadas (MVP e atualizações).

O briefing bem feito é uma ótima fonte de referência que serve como base para a equipe de desenvolvimento, evitando surpresas ao longo do projeto.

Capítulo 5. Modelagem de usuários com Personas

Antes de falarmos em modelagem, vamos definir o que são modelos: são representações estruturadas de fenômenos e abstrações complexas. Podem ser usados nas ciências naturais e sociais.

Os economistas se valem de modelos para entender o comportamento dos mercados, os físicos, para entender as partículas.

Descobrimos que pesquisar e criar modelos descritivos de nossos usuários é uma ferramenta útil e poderosa.

Alan Cooper, About Face - 2007⁶

Modelos ainda servem para otimizar a compreensão, visibilidade e comunicação de informações. Um bom modelo é aquele que **evidencia as características mais relevantes**.

Em se tratando de usuários, podemos sugerir modelos que sejam representativos para um determinado grupo, que, por sua vez, está inserido em uma população maior.

Em design de interação, a técnica mais adotada para fazer essa modelagem é chamada de “Personas”.

Por que modelar usuários?

Em seu projeto mobile, você pode até aplicar todas aquelas funcionalidades “super legais” que o gerente de marketing sugeriu, como QR Code, Realidade

⁶ Alan Cooper, Robert Reimann, David Cronin, “About Face 3: The Essentials of Interaction Design” Wiley | ISBN 0470084111 | May 7, 2007.

Aumentada, NFC, Face detect e outras tantas. Mas, na realidade, você deve se perguntar:

Quais são as ferramentas, estratégias e funcionalidades que realmente atendem meu público-alvo da melhor maneira?

É muito comum que nós (desenvolvedores, designers e gerentes de marketing) fiquemos tentados a encher nosso produto de novas ferramentas, estratégias ousadas e funcionalidades avançadas.

Mas essas decisões devem ser pautadas em informações concretas e não em “caprichos” da equipe.

E você deve estar muito bem embasado para convencer seu chefe, o gerente de marketing e sua equipe a abandonar algumas dessas coisas “super legais” que foram propostas.

Para isso, pesquise fundo quem é o seu público alvo e como ele se comporta e quais são seus perfis. Em outras palavras, **crie personas!**

O que é Persona?

É uma técnica de modelagem de usuários que utiliza pessoas fictícias para representar grupos/perfis de usuários de um produto.

A técnica é considerada barata, fácil e divertida para a equipe de desenvolvimento.

A persona é como uma ficha de personagem de RPG do usuário-modelo do sistema, criada a partir de dados reais. Contém, entre outros, o nome, gostos, hábitos, necessidades e habilidades dos usuários.

Foi mencionada pela primeira vez no livro **The Inmates Are Running the Asylum** de Alan Cooper.

Lembre-se: “fictício” não significa “falso”.

Uma persona é um arquétipo de usuário que ajuda a direcionar decisões sobre funcionalidades, navegação, interatividade e elementos visuais na interface.

Kim Goodwin, Cooper

Personas são perfis representativos de comportamentos e atividades que são contextualizadas e específicas a um determinado produto, software ou aplicação.

São representadas graficamente por meio de uma “prancha” que pode ser uma folha de papel colada na parede.

O ideal é que as pranchas de personas estejam em locais visíveis não só na sala de desenvolvimento, mas também, na da gerência e do marketing.

Figura 35 - Exemplo de uma prancha de persona.

Marcelo Santos

“Ainda tenho muita coisa pra fazer na vida”
Deficiente físico, Engenheiro Eletricista, 45 anos.



Marcelo graduou-se em Engenharia Elétrica aos 25 anos e trabalha desde os 18, com projetos que envolvem eletricidade e tecnologia. Aos 30 anos, passou em um concurso para trabalhar numa multinacional do ramo automobilístico.

Sendo fã de Motocross, Marcelo costumava sair em trilha com seus amigos quase todo fim-de-semana. Porém, aos 33 anos, Marcelo sofreu um grave acidente de moto em uma dessas trilhas, que o levou a ter que amputar o braço direito.

Sendo destro, Marcelo teve que reaprender a manusear os objetos com a mão esquerda.

“Calma, gente. Eu só perdi um braço! Não me sinto deficiente por causa disso. Realizo meu trabalho com competência e ainda sobra tempo pra me divertir com minhas filhas, esposa e meus amigos.”

Marcelo ainda pensa em montar uma banda para tocar Beatles e Beach Boys, suas bandas favoritas.
“Mas com apenas uma mão, acho meio complicado, né? (risos) Mas ainda não desisti da idéia!”

- Computador: 7
- Internet: 9
- Espontâneo, otimista.
- Meio desorganizado e não planeja as coisas com muita antecedência.

Personas são também um meio muito eficaz de comunicação interna da equipe. Quando uma descoberta importante é feita sobre o projeto, é muito mais fácil comunicar a equipe toda.

Por exemplo, utilizar:

"Marcelo Santos não está conseguindo usar nossa ferramenta de busca"

É melhor que:

"uma quantidade representativa dos participantes dos testes de usabilidade teve problemas com a ferramenta de busca".

Para que servem as Personas?

Para acabar com o mito do usuário padrão.

- Um dos mitos mais difíceis de dizimar é o tal mito do **usuário médio ou padrão**.
- Repita em voz alta: **"Não existe usuário padrão"**.
- Os usuários **não são iguais a nós**.

Para otimizar o processo de desenvolvimento

- Engaja e conscientiza a equipe de projeto
- Chega-se a um consenso dos interesses do usuário
- Mantém o foco no usuário durante todo o projeto
- Agiliza a tomada de decisões, pois não é preciso consultar usuários reais a cada etapa

Personas são uma forma de dar ao usuário um assento na mesa de desenvolvimento.

Alan Cooper

Como montar a prancha da persona

Uma boa prancha de persona deve conter:

- **Nome e foto**
- **Frase de efeito e características gerais**
- **Tipo.** Ex.: "adolescente do ensino médio" ou "aposentada doméstica".
- **Função.** Ex.: "usuário administrador" ou "editor de conteúdo" ou "usuário final".
- **Motivações**
- **Objetivos**
- **Necessidades**
- **Dificuldades.** Ex.: "Nunca usou um dispositivo touch na vida" ou "usa versão antiga do S.O em smartphone de baixo custo e baixa resolução" ou "não possui plano de dados 3G".
- **Comportamentos e habilidades.** É uma escala visual que mede certa habilidade ou comportamento da persona como “proficiência em informática: alta - baixa” ou “tendência de uso do seu smartphone: trabalho – diversão”.

Para entender melhor, veja um exemplo a seguir⁷:


Figura 36 – Template para criação de Persona

Figura 37 – Template para criação de Persona

⁷ Você poderá baixar este template em: <http://goo.gl/HlvXiG>

Figura 35 – Template para criação de Persona.

<Nome da persona>



“Insira aqui uma citação que defina bem a persona em uma frase”

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Nome <nome>
Tipo <tipo>
Função <função se houver>

Motivações

- Excepteur sint occaecat cupidatat non.
- Sunt in culpa qui officia deserunt mollit anim id est laborum.
- Excepteur sint occaecat cupidatat non.

Objetivos

- Excepteur sint occaecat cupidatat non.
- Sunt in culpa qui officia deserunt mollit anim id est laborum.
- Excepteur sint occaecat cupidatat non.

Necessidades

- Excepteur sint occaecat cupidatat non.
- Sunt in culpa qui officia deserunt mollit anim id est laborum.
- Excepteur sint occaecat cupidatat non.

Dificuldades

- Excepteur sint occaecat cupidatat non.
- Sunt in culpa qui officia deserunt mollit anim id est laborum.
- Excepteur sint occaecat cupidatat non.

Comportamentos e habilidades

Descrição da variável
 Uma característica ————— Característica oposta

Uso de redes sociais
 No smartphone ————— No PC

O smartphone é uma ferramenta para
 Trabalho ————— Diversão

Informática e internet
 Pouca experiência ————— Muita experiência

Dispositivos touch
 Pouca habilidade ————— Muita habilidade

Conhecimento do domínio (compra e venda de imóveis)
 Baixo conhecimento ————— Alto conhecimento

ThoughtWorks®

Alguns exemplos

Figura 38 - As pranchas ficam visíveis para toda a equipe

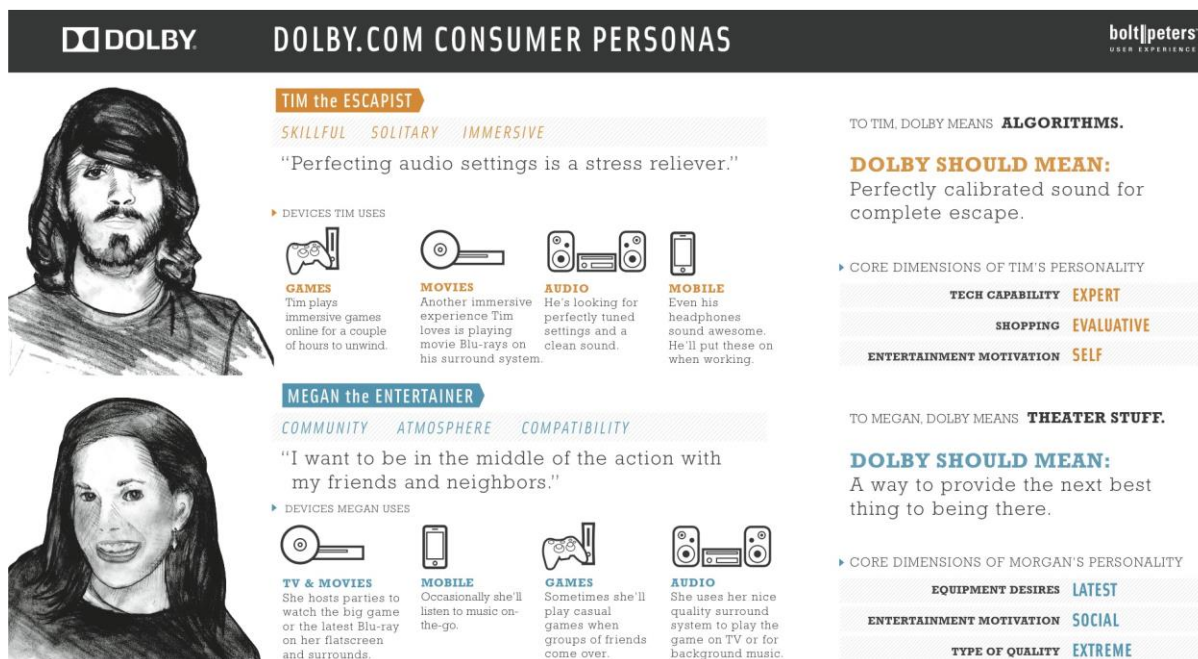


Figura 37 - Grande parte dos produtos interativos demandam mais de duas personas.



As pranchas podem ser adaptadas à realidade e ao contexto do produto, como foi o caso da Dolby – fabricante de equipamentos de áudio e vídeo.

Figura 39 - Exemplo de persona da Dolby



Como obter dados para construir Personas

Existem várias maneiras de obter dados relevantes para a construção das personas. Você pode coletar dados...

- **Analisando estatísticas de acesso** (como Google Analytics)
- **Entrevistando com equipes de suporte** (que, por sinal, sempre dão ótimas ideias sobre os diferentes perfis de usuário e suas demandas).
- **Verificando com a equipe de marketing se já não existe alguma pesquisa feita sobre o público-alvo** (mesmo que a intenção seja descobrir consumidores e não usuários)
- **Criando questionários online** (com perguntas-chave que irão revelar como são os perfis de usuário).

Alguns cuidados

Ao trabalhar com personas, podemos cometer alguns enganos típicos que poderão prejudicar todo o planejamento estratégico.

Personas não são segmentos de mercado definidos pela equipe de marketing (lembre-se: uma persona é um modelo de usuário e não de comprador).

Personas não são atores de sistema (se você é desenvolvedor, cuidado para não cair neste erro. Personas são modelos de usuário sendo que um ator pode ser até uma máquina ou outro sistema).

Personas não são papéis de usuários (papéis de usuários como “usuário administrador” ou “cliente comprador” são perfis de uso orientados à **tarefa** enquanto que as personas são orientadas a **objetivos e comportamentos**).

Não crie segmentações demais (muitos segmentos de usuários torna sua compreensão mais difícil, além de não ser muito representativo).

Nas pesquisas, não pergunte informações pessoais em exagero (pergunte a si mesmo: “será que preferência musical, religião ou inclinação política são mesmo relevantes para o projeto da interface?”).

Personas típicas para projetos mobile

Bruce Hershey, diretor sênior de estratégias mobile da Merkle, criou três personas baseando-se em dados de pesquisa do instituto Forrester Research, além de pesquisas com clientes de grandes marcas do mercado.

As pesquisas são a melhor forma de conhecer seu público e cabe a você segmenta-lo da melhor forma possível. De toda forma, as três personas típicas de Bruce Hershey podem ser um grande ponto de partida.

Touch screeners (geração touchscreen)

- 18 a 34 anos de idade (Geração Y/Millennials)
- Acompanham as últimas tendências
- Provavelmente, possuem smartphones
- Provavelmente, baixam aplicativos, acessam seu banco via smartphone, escaneiam QR Codes, assistem vídeos e pesquisam produtos

Green thumbs (“jardineiros” ou “do lar”)

- 24 a 44 anos de idade (geração x e y que provavelmente já possuem filhos)
- Inclui mães jovens que estão aumentando as atividades online via mobile para poupar tempo
- Participam de promoções e concursos

Straight talkers (seriam os “sem gríria” ou que falam sem os trejeitos dos adolescentes)

- 35 a 65 anos de idade (Geração X e Boomers com filhos adolescentes)
- Gasta a maior parte do tempo livre com a família
- Notável aumento do uso de mobile para pesquisa, mensagens de texto e muito esporadicamente pesquisa por produtos

Personas não duram para sempre

- O mundo, as pessoas, a tecnologia e o mercado mudam... Sempre!
- Por isso, pesquise sempre e identifique novas personas, se seu produto sofrer evoluções em um período longo. Até lá, o cenário pode ser outro.
- Ao terminar a criação das personas, lembre-se que **“e o produto que se adapta às pessoas e não o contrário”**.

Capítulo 6. Modelagem de tarefas com User Story Mapping

Antes de falarmos sobre User Story Mapping, vamos falar sobre análise de necessidades em um produto de software. Considere-se um sistema de apontamento de horas de trabalho de empregados relacionadas com a realização de tarefas. Quais são os interesses do empregador e do empregado em relação ao sistema?

Você há de concordar que se fizermos uma entrevista, poderemos detectar interesses explícitos de ambas as partes, mas aqueles não declarados podem ser tão importantes quanto.

Veja o quadro abaixo e responda:

Em termos de interface, o que esse sistema deveria ter para atender a estas duas personas?

Interesses do empresário	Interesses dos funcionários
<ul style="list-style-type: none"> • Evitar problemas com auditores • Aumentar lucros • Obter o máximo dos empregados • Obter melhores dados para futuros planejamentos • Melhorar a organização da empresa 	<ul style="list-style-type: none"> • Manter seu emprego • Fazer seu trabalho para ir embora para casa na hora certa • Fazer o patrão feliz para ser promovido • Não se esquecer de anotar para evitar desconto de pagamento.

Por isso é importante mapearmos as necessidades dos usuários e relacioná-las com o valor de negócio.

Pois em 2005 Jeff Patton, um guru da área de processos ágeis criou uma técnica chamada **User Story Mapping**.

Trata-se de uma técnica colaborativa, cuja missão é auxiliar na priorização e planejamento das funcionalidades (estórias) que constarão no produto iterativo.

Esse mapeamento ajuda a definir os releases (lançamentos) do produto.

O que é uma história?

Uma história não é mais do que a descrição de uma pequena funcionalidade que o cliente pretende ver desenvolvida no sistema.

O termo em inglês é *story* (história – conto) e não *history* (história - relato de fatos).

O que são:

- Uma **breve** descrição de uma funcionalidade que foi discutida
- São tradicionalmente escritas em cartões ou post-its

O que não são:

- Documentos de implementação (classes, modelagem...)
- Imutáveis: Podem sofrer alterações e negociações ao longo do projeto
- Casos de uso: Este último se refere à narrativa de funções de forma impessoal, ou seja, independente do usuário

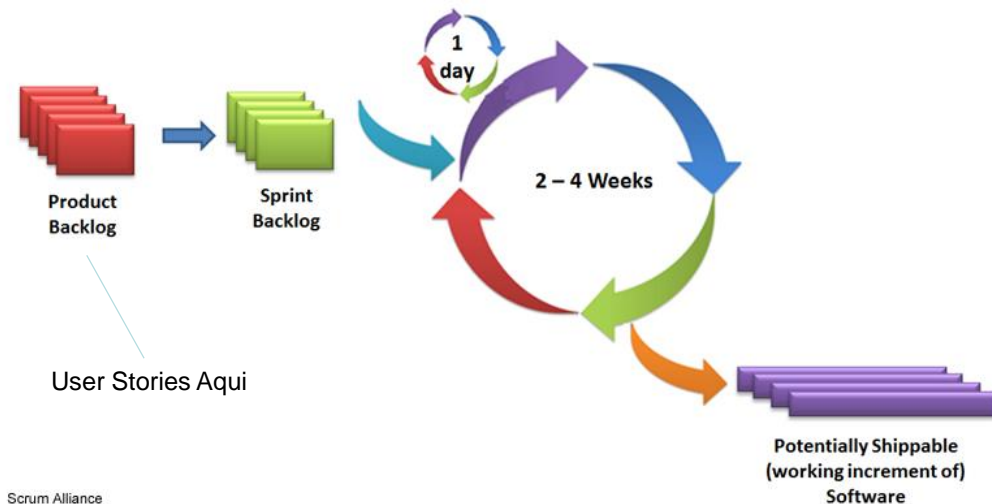
Características

- Focam nos objetivos do usuário e como o sistema alcança esses objetivos.
- Fracionam os requisitos para que seja possível (e mais fácil) estimar o esforço para realizar aquele objetivo. Resumindo, *User Stories* são descrições simples que descrevem uma funcionalidade e é recomendável que sejam escritas **segundo o ponto de vista do usuário**.
- Devem ser curtas, simples e claras. Devemos conseguir escrevê-las em um simples e pequeno cartão (conhecidos como *User Index Cards*). Se não há espaço para escrevê-la em um cartão é por que devemos refiná-la mais, e as dividir em outras *User Stories*.
- *Stakeholders* escrevem *User Stories*, não os desenvolvedores. *User Stories* são simples o suficiente para que as pessoas possam aprender a escrevê-las em alguns minutos.

Por que mapear e não listar?

Quem já trabalha com métodos ágeis, já deve estar acostumado com esse diagrama.

Figura 40 - Diagrama de atividades do processo SCRUM.

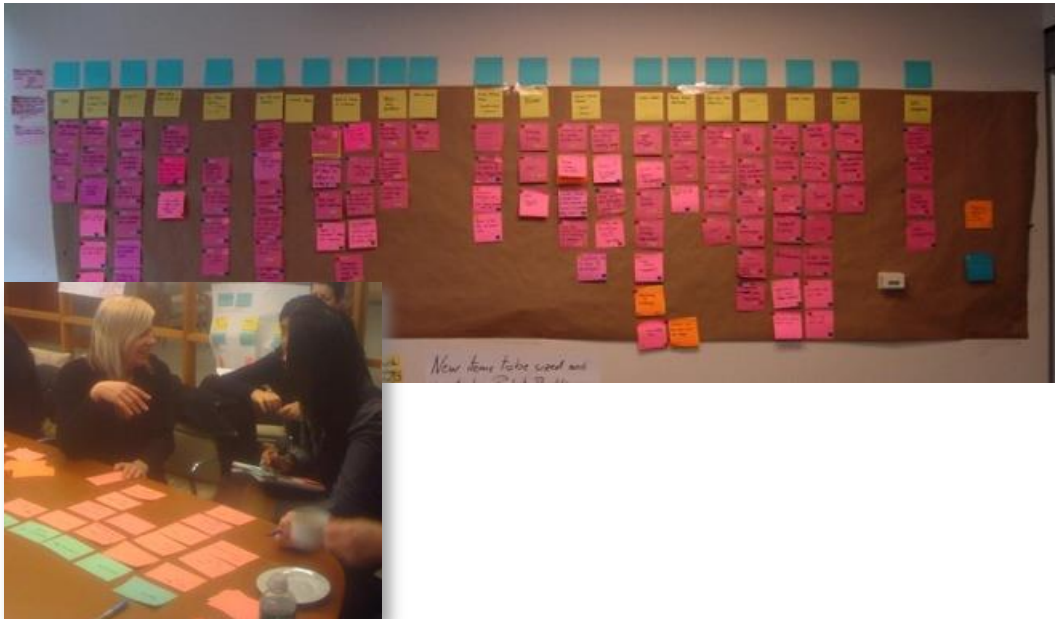


Trata-se do diagrama de fluxo de trabalho do método chamado SCRUM. Mas aqui os requisitos são listados ou “pilhados” em um backlog que é esvaziado a cada ciclo.

Diferentemente do backlog típico do Scrum, o USM:

- Torna mais visível o fluxo de atividades e a cadeia de valor de cada estória.
- Mostra os relacionamentos entre estórias dentro de uma atividade maior.
- Ajuda a checar a completude do backlog.
- Revela um contexto mais amplo do que deve ser priorizado.
- Ajuda a planejar os releases com mais valor (de negócio) agregado.

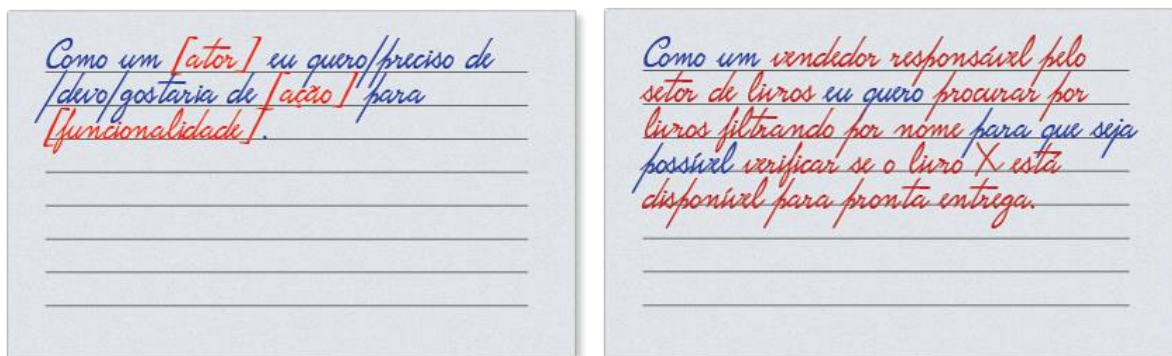
Figura 41 - Mapeando histórias e gerando discussões.



Criando um User Story Mapping

Cada história de usuário possui uma sintaxe como no exemplo abaixo:

Figura 42 - Exemplo de sintaxe para a criação de uma história.



Formato

Ator – O proprietário da User Story. De forma simplista é o usuário, o interessado naquela funcionalidade. Mas é recomendado descrever de forma específica quem é o ator para ser mais fácil identificar o contexto da história dentro do sistema.

Ação – É o que o ator quer fazer. Utilizando aquela ação ele espera alcançar seu objetivo dentro do sistema.

Objetivo/Funcionalidade – É o que o ator espera que aconteça ao realizar a ação. Ou seja, é o resultado de executar a ação segundo a ótica do ator. Também pode ser visto como justificativa.

Mas este é apenas um modelo de base. Você pode criar estórias de acordo com sua necessidade ou realidade.

Escrevendo boas histórias

É importante que as estórias tenham as seguintes características

- **Independentes:** Estórias devem ser independentes uma das outras. Ex.: usuário pode entrar com o nome do meio, primeiro nome e último nome.
- **Negociáveis:** Estórias não são contratos, mas lembretes para discussões. (Por questões de escopo, orçamento ou complexidade, as histórias podem ser removidas ou alteradas.)
- **De valor agregado:** Estórias devem agregar valor para o usuário/cliente; Ex.: dizer que o sistema será feito em Java e MySQL não é relevante para o usuário/cliente).

Escrevendo boas histórias

- **Estimáveis:** Os desenvolvedores devem ser capazes de estimar o tamanho das histórias; (se não puder ser estimada, não será usada no sprint. Se for muito complexa, dividir em histórias menores.)

- **Curtas:** histórias grandes dificultam as estimativas. Bem como histórias muito pequenas. Quebre ou agrupe dependendo do caso. Grandes histórias (épicas) são difíceis de estimar e difíceis de planejar, elas não se encaixam bem em uma única iteração.
- **Testáveis:** Histórias devem ser possíveis de serem testadas. Em vez de: “Não fazer o usuário esperar muito em cada tela”, use: “As telas não deverão demorar mais que 2 segundos para abrir”.

Veja abaixo o processo de montagem de um USM⁸

1- Organize os cartões horizontalmente em uma sequência lógica de operação



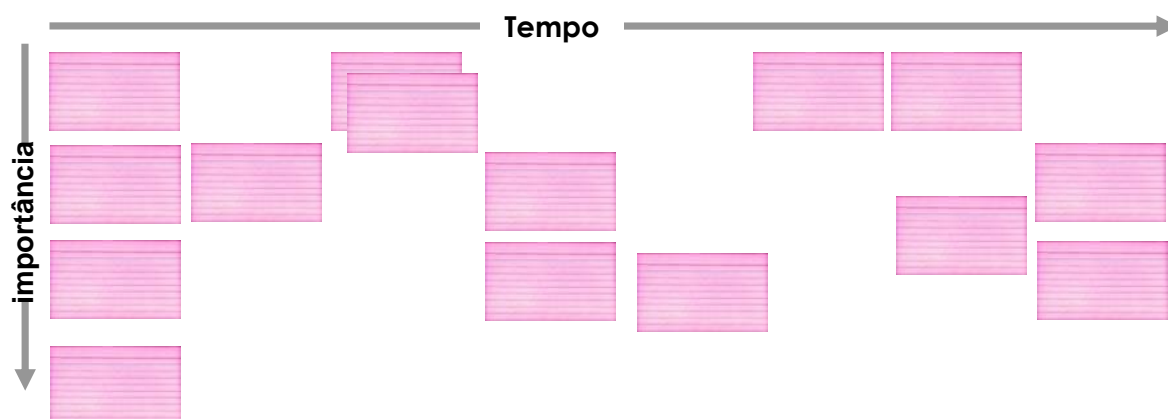
2- Empilhe os cartões que possuem atividades simultâneas



3- Ordene verticalmente os cartões em ordem de importância ou criticidade

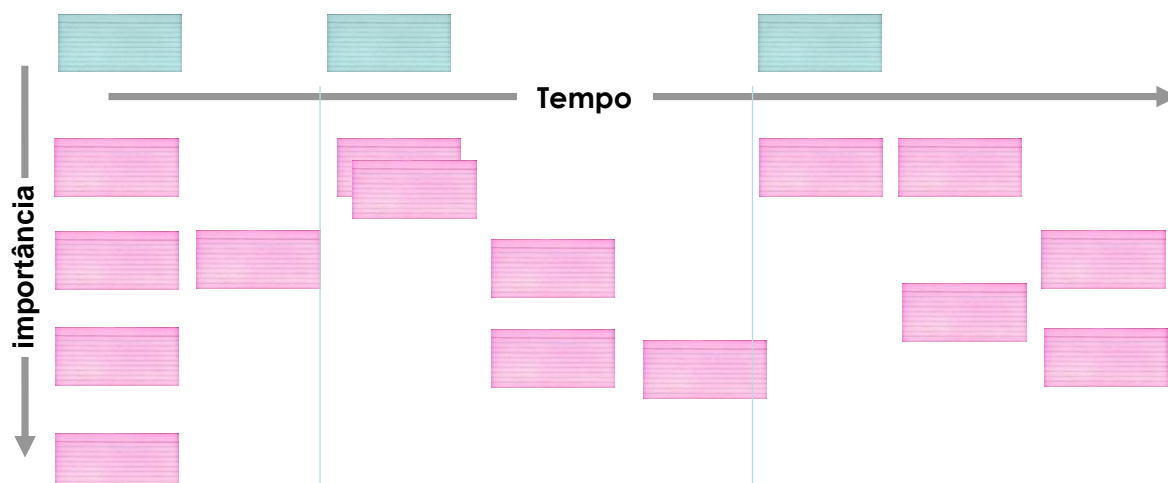
(frequência de uso, ROI, valor do negócio, etc.)

⁸ © Jeff Patton, all rights reserved, www.AgileProductDesign.com

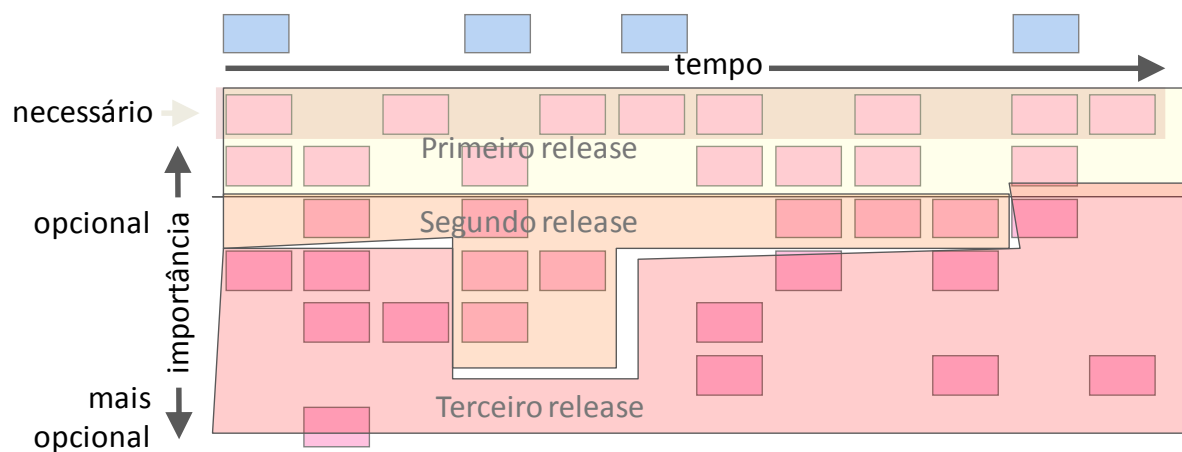


4- Observe os agrupamentos que se formam e discuta com a equipe.

Um conjunto de tarefas, forma uma **atividade**:



5- Discuta e escolha uma das atividades que será implementada primeiro. Peça para a equipe de desenvolvimento estimar a implementação de cada cartão. Essa deve ser uma decisão estratégica!



Benefícios em desenvolver em releases

Iteração é diferente de incrementação

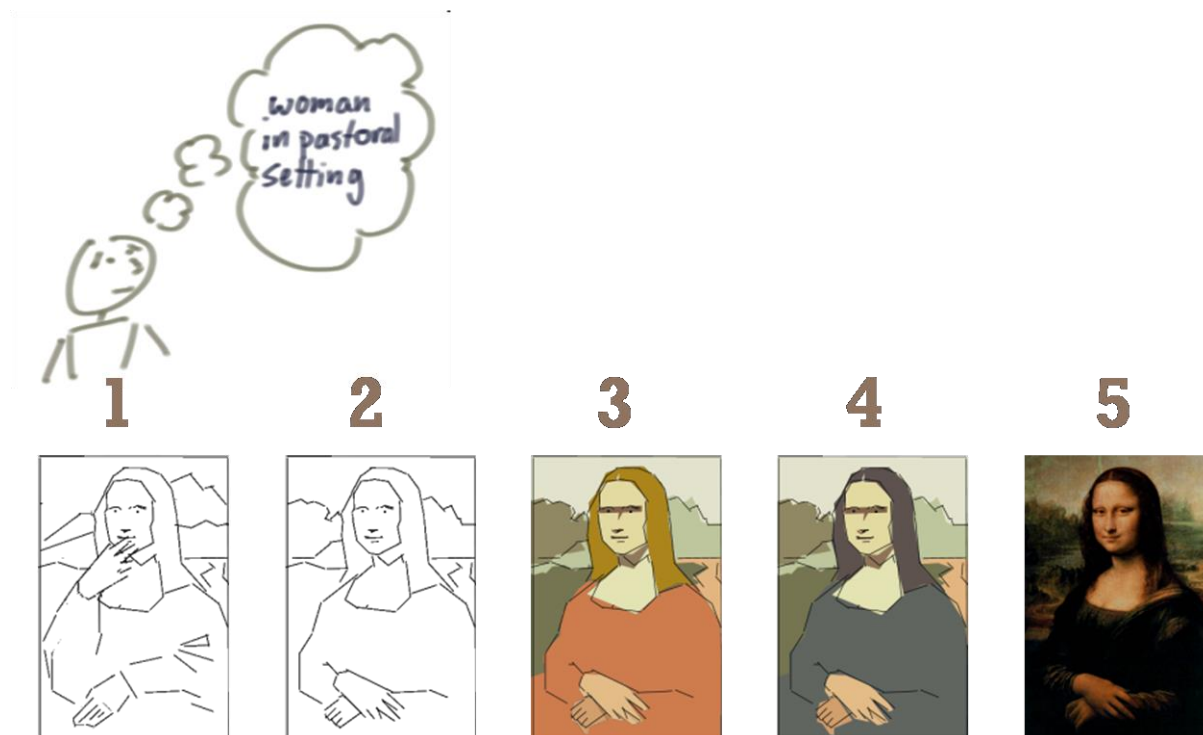
Incrementação presume que já se tenha uma ideia completa e imutável do que será construído. Além disso, terminar o projeto no prazo estimado requer uma estimativa ultra precisa.

Figura 43 - Exemplo de um processo incremental.



O processo iterativo parte de um rascunho e, gradativamente vai aperfeiçoando a qualidade. Além disso, permite criar produtos a partir de uma mera ideia sendo que as correções, validações e ajustes vão acontecendo ao longo do caminho. Portanto, iteração é entregar o produto em partes funcionais ou releases.

Figura 44 - Exemplo de um processo iterativo (entregue em releases).



Outros benefícios em se fazer um USM

1. Ajuda a entender, como um todo, o que o sistema faz.
2. Ajuda a entender o que está sendo construindo.
3. Ajuda a fazer um planejamento e entendimento do release.
4. Visão mais clara do que é preciso fazer antes e o valor agregado.
5. Por ser uma técnica colaborativa, engaja a equipe.
6. Prioriza os requisitos (estórias) de uma forma ágil.

7. Preenche o abismo entre processo de design e o processo ágil de desenvolvimento.
8. Evita a síndrome do canivete suíço (*feature creep*) já que as histórias são bem discutidas.

Capítulo 7. Arquitetura de informação

Em nossa vida, nós temos a necessidade de organizar, classificar e categorizar todo que conhecemos.

Figura 45 - Todo mundo categoriza/classifica/organiza.



E essa é uma prática que aperfeiçoamos a cada dia para tirar proveito da informação organizada.

Figura 46 - Em um cardápio, a informação é organizada pelos sabores e tipos.



Todo mundo categoriza/classifica/organiza. O que varia são os esquemas de organização. Vejamos alguns esquemas e suas características.

Esquemas de organização ambíguos

Por assunto

Organiza a informação em temas.

Ex.: cardápios, editoriais do jornal, supermercado.

Por tarefa

Organiza a informação em sequência de ações.

Ex.: Menu aplicativos Windows (editar, exibir, formatar).

Por audiência

Organiza a informação para diferentes públicos.

Ex.: Lojas de departamento (classifica seus produtos em masculino, feminino e infantil).

Esquemas de organização exatos

Alfabeto

Indicado para organizar informação extensa e de público variado.

Ex.: dicionários, enciclopédias, listas telefônicas.

Tempo

Indicado para mostrar a ordem cronológica de eventos.

Ex.: Livros de História, Grade de programação, banco de notícias.

Localização

Compara informações de diferentes locais.

Ex.: Previsão do tempo, pesquisa boca de urna, atlas.

Sequência

Organiza itens por ordem de grandeza e valor.

Ex.: Lista de preços, classificação do campeonato.

O que é e para que serve a A.I

Para Batley (2007), a “arquitetura não aborda somente criações robustas e estruturas atraentes, aborda também o que está contido no seu interior: a sua funcionalidade”. A arquitetura fornece a estrutura, a informação vai determinar a funcionalidade. Segundo Rosenfeld e Morville (1998), a AI é fruto da combinação de quatro elementos:

- a) **Sistema de organização:** refere-se a uma maneira lógica de classificação informacional, definindo os tipos de relacionamento entre itens de conteúdos e grupos.
- b) **Sistema de rotulagem:** representa a nomenclatura dada a cada grupo ou categoria de informação.
- c) **Sistema de navegação:** apresenta a trajetória que o usuário terá disponível no website, sistema ou aplicativo para acessar cada informação com a distribuição de links, botões e menus.
- d) **Sistema de busca:** é a estratégia para auxiliar o usuário na localização e no acesso rápido a informações armazenadas no website.

Padrões de interação. Por que em mobile é diferente?

A primeira coisa que temos que entender é que criar interfaces para dispositivos móveis **é diferente**... e não é “**só**” por causa do tamanho.

O principal efeito da portabilidade é que as pessoas usam esses dispositivos nos mais variados lugares e contextos.

1- Por causa do tamanho e da estreita largura de banda, os usuários esperam uma interface simples e limpa além do rápido carregamento do conteúdo

2- Por causa da portabilidade, podemos levá-lo a todo lugar e ainda operá-lo em paralelo com outras atividades como vendo TV ou almoçando.

3- Uma agência chamada Foolproof realizou uma pesquisa revelando que 63% das pessoas têm uma relação quase afetiva com seu aparelho e se sentem perdidas quando estão fora de alcance

Esquemas típicos de navegação mobile

É possível dizer que os dispositivos móveis já têm seu próprio padrão de A.I.

Enquanto sites responsivos (sites comuns que re-adaptam seu layout ao dispositivo) tendem a seguir os padrões, digamos, mais “clássicos”, os app nativos (feitos na linguagem do dispositivo) seguem uma estrutura de navegação já consolidada.

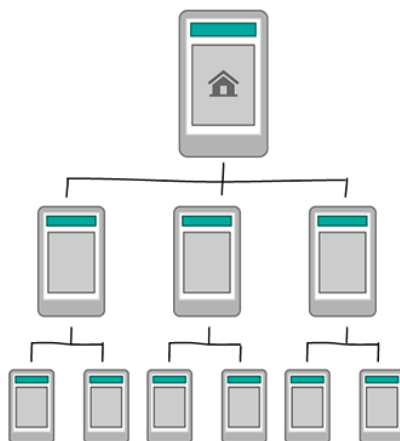
Atenção.

Os exemplos a seguir são **convenções** já consolidadas no mercado e não **regras** rígidas. As convenções são soluções que alguém desenvolveu e que tiveram boa aceitação.

1- Hierarquia

Se assemelha bastante com a estrutura hierárquica de um website. Se você estiver em um projeto de site responsivo, é mais adequado seguir este padrão.

Figura 47 - Exemplo de esquema hierárquico.



Entretanto, nada o impede de melhorar e incrementar a navegação com outros esquemas.

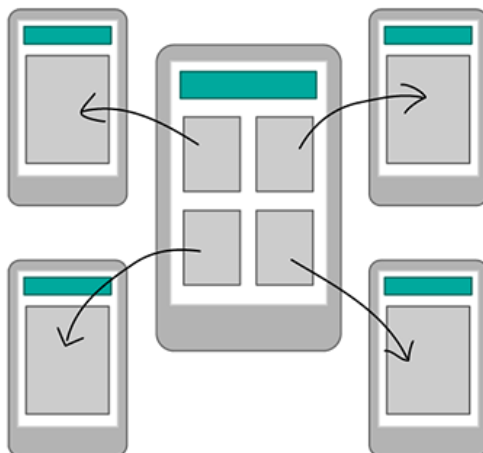
Vantagem: Ajuda a estruturar informações complexas.

Cuidado: Navegação longa e multi-facetada pode ser um problema em telas pequenas.

2- Hub & spoke

É um padrão que típico no iPhone no qual o usuário parte de uma tela inicial (*hub*) e navega para as de fora (*spokes*). Para navegar para outro *spoke*, é preciso voltar ao *hub*. Não há como navegar entre *spokes*.

Figura 48 - Exemplo do esquema Hub & spoke.



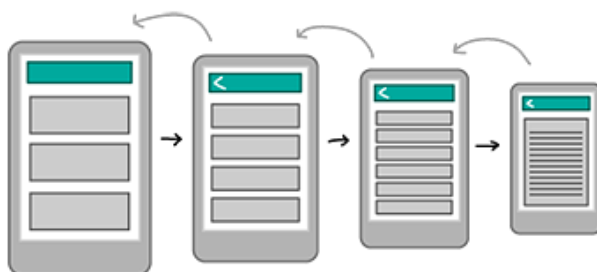
Vantagem: Útil para ferramentas com multi-funções ou que possuem restrições de navegação em que o usuário precisa interagir com uma tela por vez, como numa tela de configuração, por exemplo.

Cuidado: Não use em casos em que o usuário demanda tarefas múltiplas.

3- Nested doll

É um padrão faz com que o usuário vá aprofundando em detalhe no conteúdo. Em condições difíceis, promove uma navegação mais segura e faz com que o usuário tenha uma noção mais precisa de onde está e se deve ir para frente ou para trás.

Figura 49 - Exemplo de esquema Nested doll.



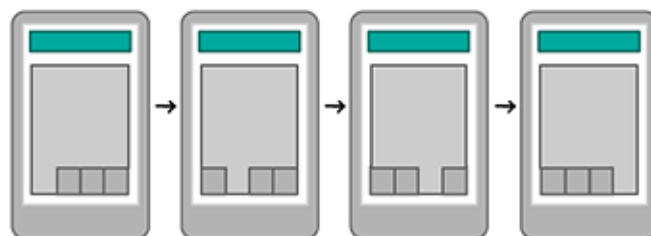
Vantagem: Útil para navegar entre tópicos fortemente relacionados entre si.

Cuidado: Os usuários não serão capazes de alternar rapidamente entre tópicos. Considerar se este padrão ajuda ou atrapalha o usuário a explorar o conteúdo.

4- Tabbed view

Também chamado de painel de abas, é um padrão já bem difundido e faz com que os usuários tenham uma noção rápida do conteúdo todo.

Figura 50 - Esquema de navegação Tabbed view.



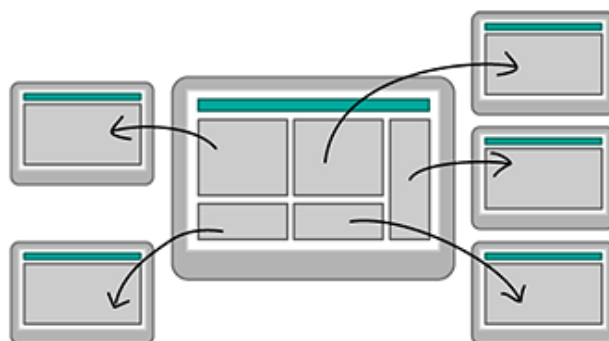
Vantagem: Útil para aplicativos de ferramentas (não de conteúdo) e multitarefas.

Cuidado: Esse padrão é para estruturas simples, não para as complexas.

5- Dashboard

Permite uma interface mais detalhada do conteúdo já na tela de abertura. Serve para mostrar conteúdos e ferramentas ao mesmo tempo e é mais indicado para tablets.

Figura 51 - Esquema Dashboard.



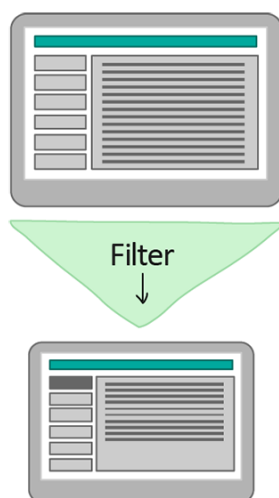
Vantagem: Útil para aplicativos com ferramentas e conteúdos em tablets.

Cuidado: Demanda mais espaço e maior habilidade com design de interfaces.

6- Filtered view

Permite navegar em um conjunto de dados por meio de filtros que, quando acionados, exibem o conteúdo relacionado e ocultam os demais. Pode ser útil em navegações multifacetadas.

Figura 52 - Esquema Filterd view.



Vantagem: Útil para aplicativos e sites com grande quantidade de conteúdo como vídeos, artigos e imagens. Bom para revistas e sites de conteúdo.

Cuidado: Como no dashboard, demanda mais espaço e maior habilidade com design de interfaces.

Algumas considerações

1. Uma boa Arquitetura de Informação leva em conta o **modelo mental dos usuários**.
2. Um **erro bastante comum** em organizar conteúdos para websites é pensar uma navegação espelhando a **estrutura organizacional da empresa**.
3. Lembre-se: para o usuário lá fora, sua estrutura interna e suas siglas podem não fazer sentido algum.
4. Na medida do possível **tente sempre validar a arquitetura** com usuários potenciais. Se não for possível, ao menos com pessoas próximas.
5. Como bem disse Jacob Nielsen, “**qualquer teste é melhor que nenhum teste**”.

Capítulo 8. Prototipagem

Protótipos são modelos.

No contexto do software, são representações visuais que servem para melhorar a visibilidade e compreensão de telas, fluxos e interações, sempre do ponto de vista do usuário.

Podem se de baixa ou alta fidelidade (ou resolução).

Protótipos de baixa fidelidade, geralmente são feitos em papel.

Os de alta fidelidade, também são chamados de Wireframes, ou protótipos funcionais

Sempre acontecem depois da Arquitetura de Informação.

Prototipação em papel = Prototipação rápida

- É um método que permite delinear de forma ágil e barata, a interface de um sistema (web, app, game, etc.)
- Devem elucidar a lógica e as regras de negócio do produto
- É bem barato! (Gasta-se, basicamente, papel, lápis e tempo)
- Possui o fator “engajamento” (por envolver mais pessoas e ter um caráter quase lúdico)
- Não precisa ser bonito! (Aliás, a proposta é ser ágil. O foco é na experiência de uso)
- Não deve refletir a aparência estética (design)
- Podem ser usados em testes de usabilidade.

Figura 53 - Exemplo de um protótipo em papel.



Pode servir de documentação, em lugar às documentações mais extensas.

Deve ser um reflexo de:

- Modelagem de tarefas.
- Requisitos.
- Arquitetura de informação.
- Conteúdo.

Benefícios

- As alterações de design, decorrentes de testes com usuário, podem ser feitas em tempo real.
- As pessoas que participam do teste ficam mais confortáveis em criticar o produto, pois têm plena noção que se trata de um protótipo.

- Propor alterações conceituais nessa etapa, economiza tempo e dinheiro, já que alterar um software pronto é sempre mais custoso.

“Estima-se que seja 100 vezes mais barato fazer mudanças antes de escrever qualquer código, do que aplicá-las após a implementação.” (Jakob Nielsen)

Construindo o protótipo rápido

Antes de começar, veja abaixo quais aspectos podem ser abordados num protótipo rápido.

O conceito. O protótipo deve ter a capacidade de mostrar à equipe e os usuários quais são os conceitos principais do produto.

A terminologia. Valida a terminologia ou rotulagem usada na etapa de A.I.

A navegação. Confirma se os usuários entendem o fluxo de telas proposto na etapa de A.I.

O conteúdo. O protótipo exibe os conteúdos no nível de profundidade adequado?

O layout. O conteúdo e os elementos interativos estão dispostos da maneira correta?

Novas funcionalidades. É possível identificar quais são as novas funcionalidades que estão presentes no produto (útil para casos de atualizações de produtos já existentes.)

Em 2009, a *Interactions Magazine* publicou um artigo de três pesquisadores da Universidade de Indiana, que propunha uma nova abordagem para a prototipagem em papel. A proposta seria inserir os desenhos no próprio dispositivo para que os testes fossem mais fidedignos.

Segundo o artigo, a prototipagem deveria conter as etapas a seguir:

1. Desenhar os protótipos
2. Digitalizar os desenhos e ajustar os tamanhos das telas
3. Salvar as imagens na galeria de fotos do dispositivo
4. Ordenar as imagens na ordem correta do fluxo de operação
5. Ao testar, explicar ao usuário que a interação se dará ao fazer o “slide” de uma foto para outra

Entretanto, já existem ferramentas próprias destinadas à prototipagem via “Paper in Screen”, como o *POP – Prototyping On Paper*.

Passo 1

Desenhe os protótipos com papel e caneta/lápis

Figura 54 - Desenhando protótipos em papel.



Passo 2

Fotografe seus protótipos com o aplicativo POP.

Figura 55 - Fotografando os protótipos com o aplicativo POP.



Passo 3

Ajuste o tamanho do desenho (*move and crop*) na tela no aplicativo.

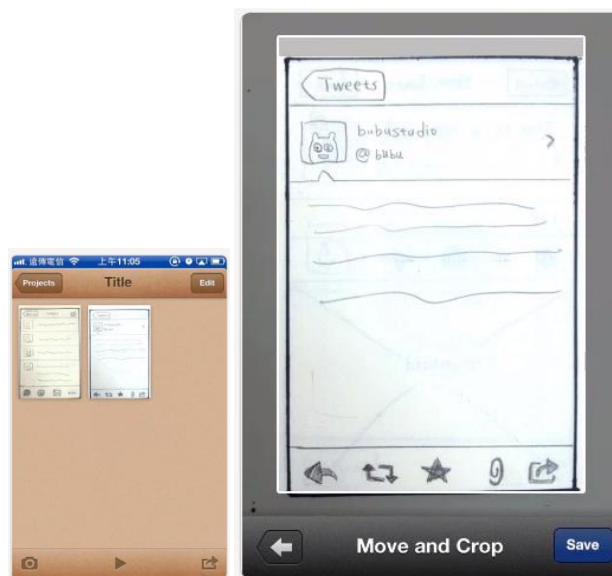
Figura 56 - Ajustando as dimensões no aplicativo.



Passo 4

Ajuste o tamanho do desenho (*move and crop*) na tela do aplicativo e clique em “Save”.

Figura 57 - Repare que seu desenho será salvo juntamente com as outras telas desenhadas.



Passo 5

Faça os links entre as telas desenhadas, marcando as “áreas sensíveis” em cada desenho.

Figura 58 - Ao clicar em “Link to”, basta escolher outra tela desenhada.



Passo 6

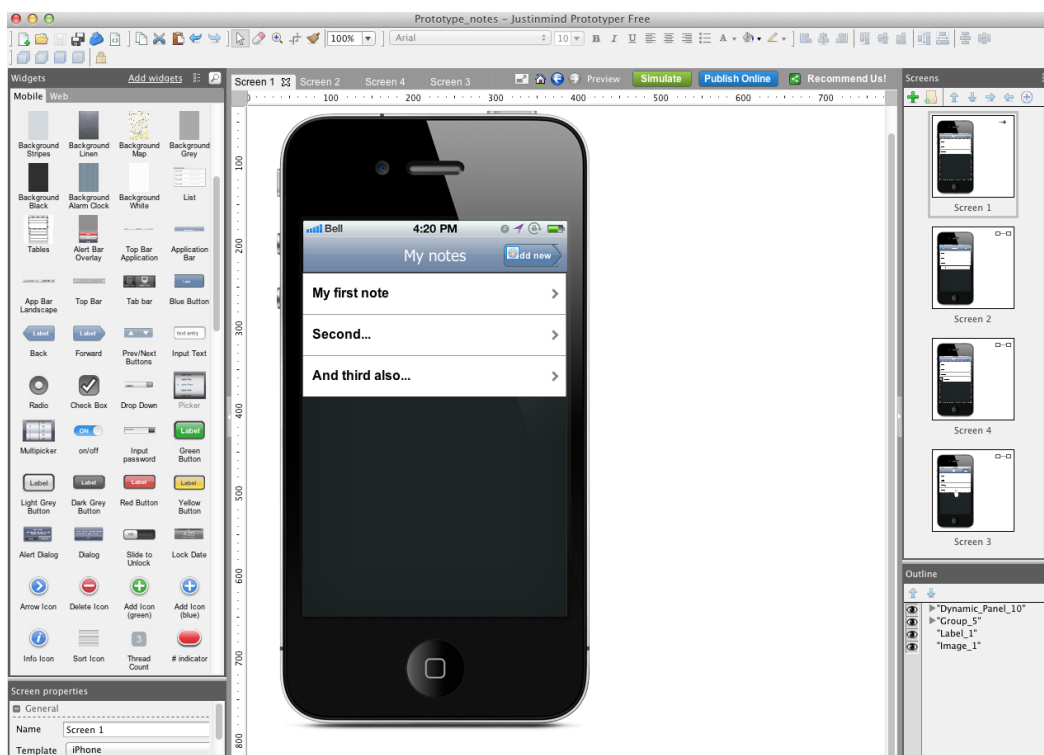
Publique seu protótipo na plataforma on-line e convide usuários para testá-lo. Seu protótipo pode ser **público** (qualquer pessoa com a URL) ou **privado** (apenas para as pessoas convidadas por você).

Protótipos funcionais (wireframes)

Também conhecidos como **Wireframes**.

São protótipos que simulam o comportamento, funcionalidade e a navegação de forma mais semelhante à versão final.

Figura 59 - Snapshot do JustInMind.



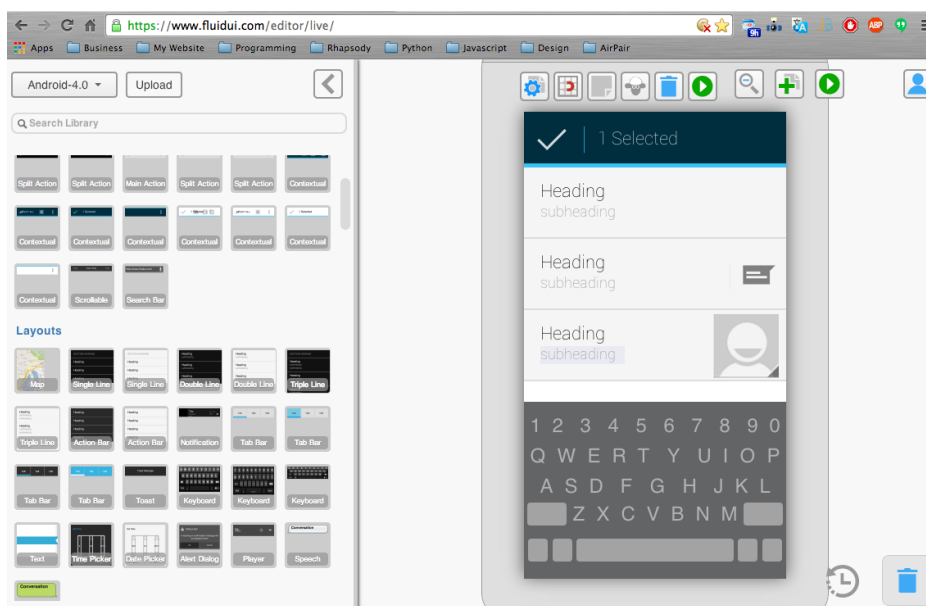
São feitos a partir de ferramentas mais elaboradas como:

- JustInMind.
- Excode (Apple).

- Axure.

Existem ainda ferramentas online, como o excelente Fluid, que além de possuir muitos componentes prontos, é multiplataforma e ainda permite compartilhar a URL do protótipo no final.

Figura 60 - Snapshot do Fluid.



Considerações

1. Muitos acreditam que prototipação em papel ou funcional resolvem todas as questões de usabilidade relacionadas à interação mobile – o que não é verdade.
2. Um **erro bastante comum** é não considerar aspectos como **ambiente**, **contexto** e até o **hardware**.
3. Considerar as questões ligadas ao hardware podem ser, ao mesmo tempo, **restrições** e **oportunidades**.
4. Prototipar é mais útil para **novos produtos** ou **novos designs** de parte de um produto maior. Para produtos existentes, é melhor usar a versão atual.

5. Útil para sites responsivos cujo conteúdo é mais **rico e cheio de atualizações**. Para sites mais estáticos ou mais simples, pode ser um fator que aumente os custos e os prazos.

Capítulo 9. Elementos de design

Diversas plataformas adotam suas diretrizes de **design e interação**.

O objetivo é dar aos desenvolvedores um guia que irá **nortear os aspectos visuais e interativos** de cada plataforma. No link abaixo você encontra uma lista completa de diretrizes de interação de várias plataformas: <http://goo.gl/4nFA68>.

Como dito anteriormente, no mercado plataformas mobile, o que vemos é uma corrida de dois carros: iOS e Android. Por questão de escopo e limitação de tempo.

Mas eu vou passar um link com uma lista completa de style guides.

Cada plataforma possui seus próprios princípios (ou manifestos) em relação ao conceito adotado em seus designs:

iOS 7 Manifesto:

Respeito. A interface do usuário ajuda os usuários a entenderem e a interagirem com o conteúdo, mas nunca compete com ele.

Clareza. Texto é legível em todos os tamanhos, ícones são precisos bem desenhados, adornos são sutis e adequados ao contexto, e o design é orientado a dar foco na funcionalidade.

Profundidade. Estética visual e movimentos suaves transmitem “vitalidade” e aumentam a satisfação e compreensão dos usuários.

- <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html>

Android Manifesto:

Encante-me. Beleza e estética em vários níveis com desenhos e tipografias nítidas. [...] Seu app deve combinar beleza e simplicidade e minimizar o esforço do usuário sempre.

Simplifique minha vida. Seu app deve ser fácil de entender e ajudar em tarefas simples. [...] Pessoas de diversas culturas e idades devem sentir-se no controle das ações.

Faça-me ser incrível. Fazer um app fácil não é suficiente. As pessoas devem ser capazes de personalizar funções e ter uma experiência de uso nas mais inventivas maneiras.

➤ <http://developer.android.com/design/get-started/creative-vision.html>

Na maioria das plataformas, as diretrizes dizem respeito a:

Estilo visual - Temas diversos / Tipografia / Iconografia / Métricas e grid / Estrutura de tela / Branding

Padrões de interação - Gesture / Navegação / Feedback / Animações / Seleção / Notificações

Figura 61 - Acesse o documento completo em: <http://goo.gl/sUysJt>.



Figura 62 - iOS Developer Library. Acesse: <http://goo.gl/01xkAk>.

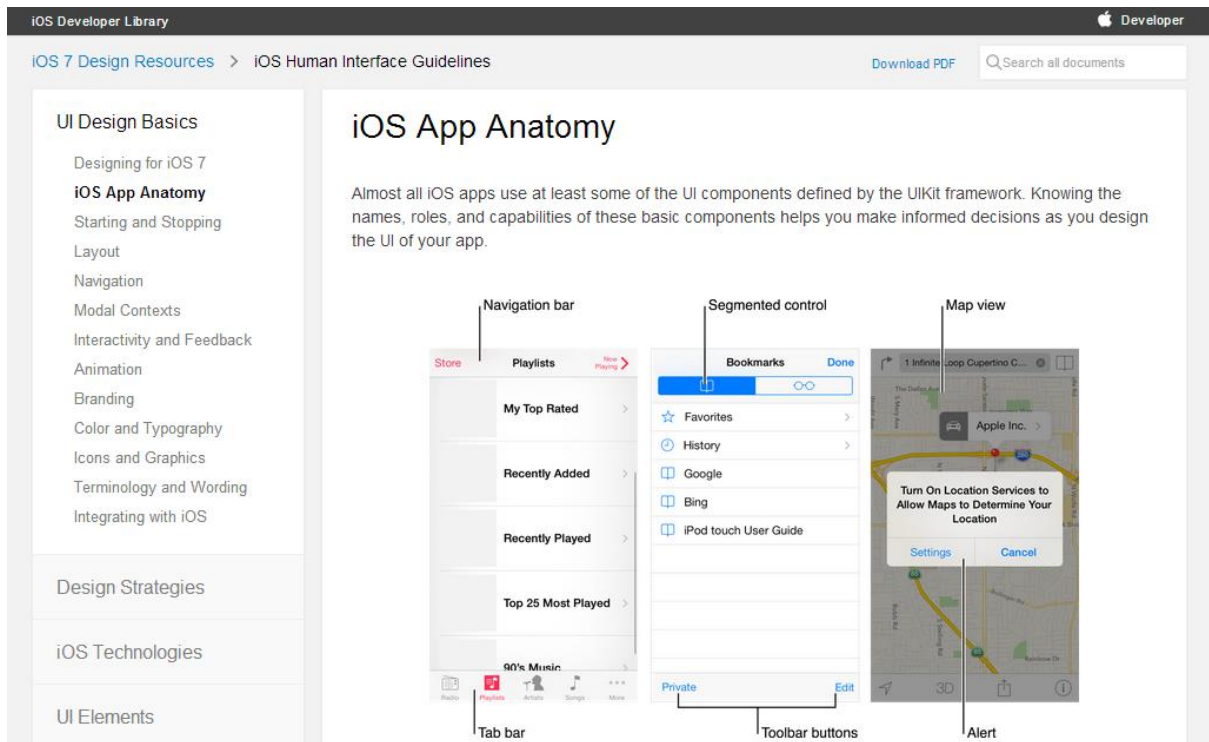
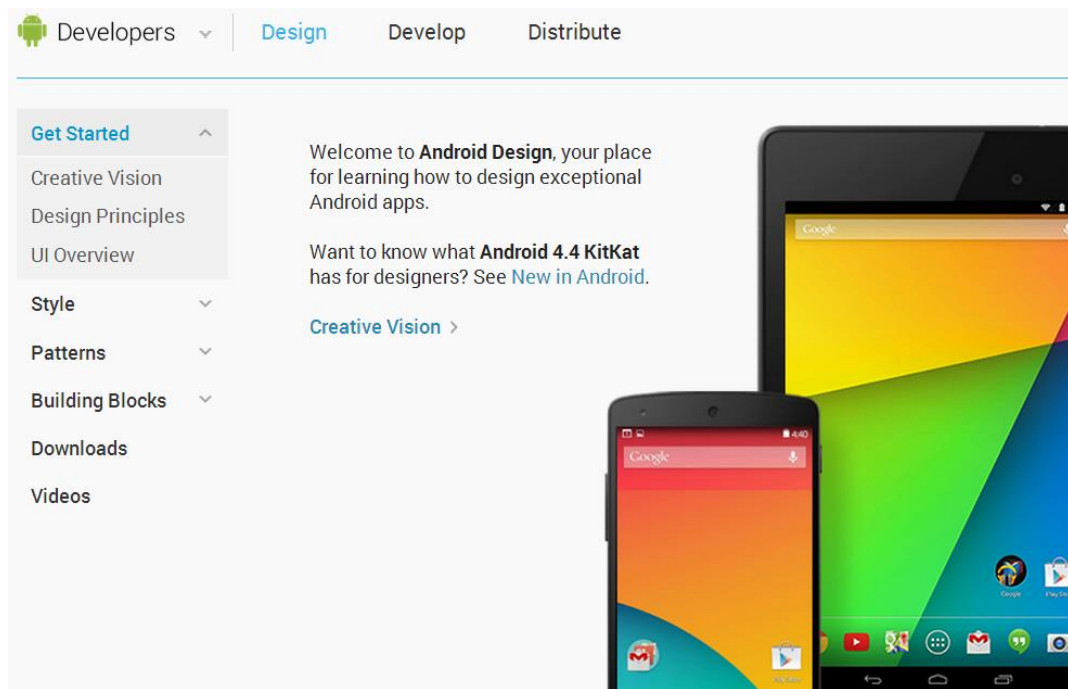


Figura 63 - Android Developer. Acesse: <http://developer.android.com/design/>



Design Stencils

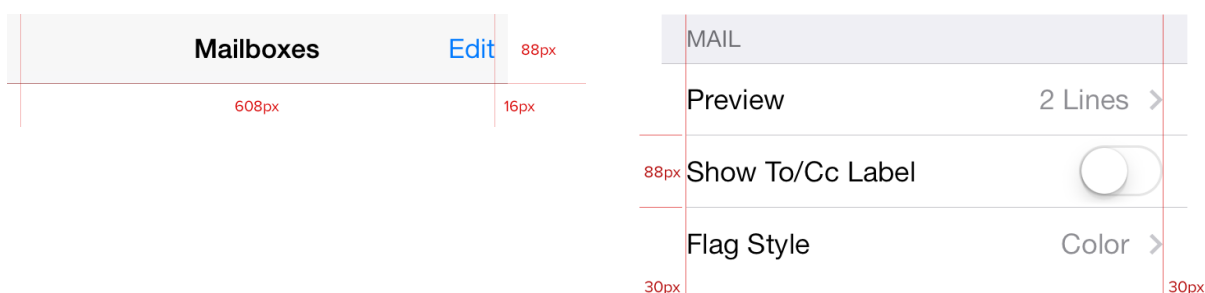
Existem hoje diversos sites que oferecem download imagens como ícones, botões, caixas, barras e demais elementos de design para cada plataforma.

Esses elementos são chamados de *stencils*. Geralmente são oferecidos em:

- Um pacote de arquivos de imagem (png) com todos os elementos visuais reunidos em um .zip, ou
- Um arquivo editável com as imagens em vetor para serem estilizadas individualmente. Pode ser em formato Photoshop (psd), Illustrator (ai).

Além da documentação oficial, existem outras paralelas que ajudam a compreender melhor como funcionam outros padrões até de versões antigas e outros dispositivos, como é o caso do iOS design cheat sheet⁹:

Figura 64 - iOS design cheat sheet.

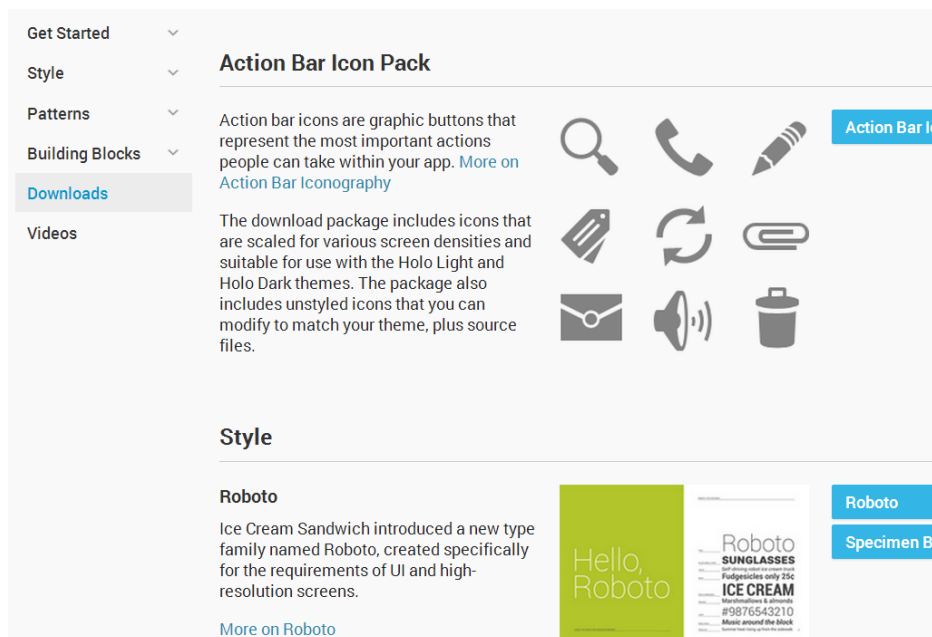


Já o Android oferece uma seção no site totalmente dedicada ao download¹⁰ de stencils em diversos formatos, inclusive a tipografia oficial da plataforma (Roboto):

⁹ <http://ivomynttinen.com/blog/the-ios-7-design-cheat-sheet/>

¹⁰ <http://developer.android.com/design/downloads/>

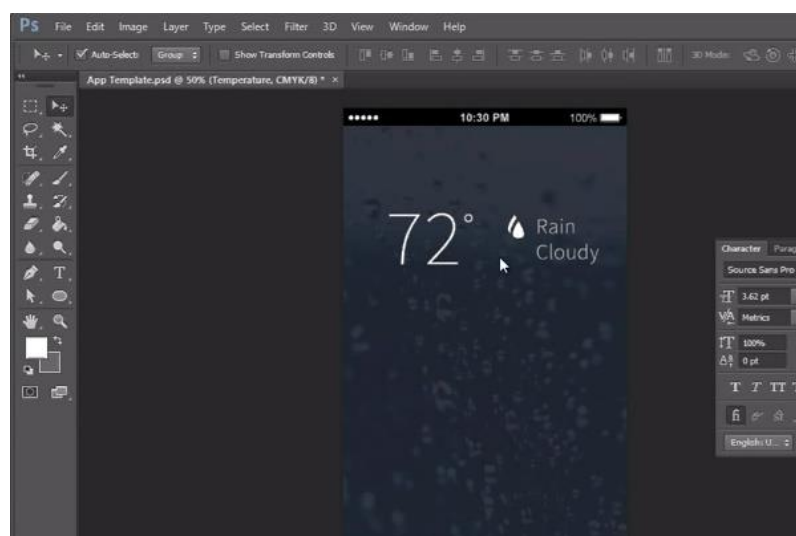
Figura 65 - Seção de downloads no Android Developer.



Ferramentas

Provavelmente o Photoshop é a ferramenta mais utilizada no desenho de interfaces, tanto para Web quanto para mobile.

Figura 66 - Snapshot do Photoshop.



O Omnigraffle (apenas Apple) também é bastante utilizado para aplicações web e mobile. É ainda muito usado para protótipos.

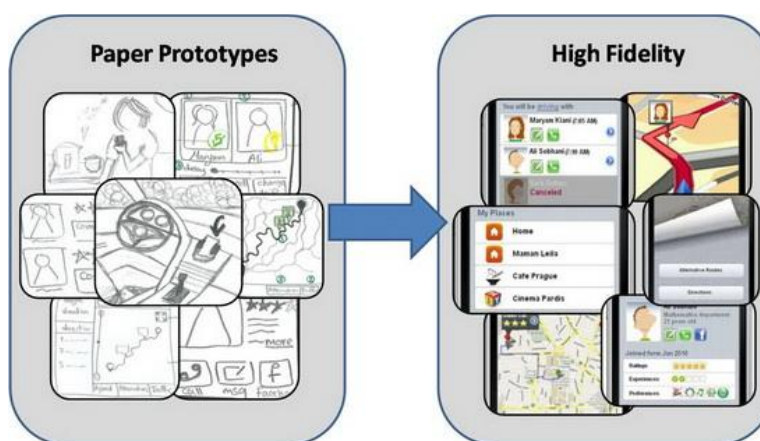
Figura 67 - Snapshot do Omnigraffle.



Considerações

A fase do design da interface, também considerada como um protótipo de **alta fidelidade**, acontece depois da etapa de prototipagem.

Figura 68 - Protótipos de baixa e alta fidelidade.



1. Geralmente é feita por designers de interface, mas nada impede que você mesmo faça os desenhos, uma vez que já conhece os padrões de interação e estilo de cada plataforma.
2. Para demandar esse tipo de serviço, é necessário que se conheça bem as diretrizes de design aqui apresentadas.
3. Você pode ainda criar um guia de estilo para o seu aplicativo, assim que o layout estiver concluído, a fim de orientar os desenvolvedores na hora de implementar as telas e componentes.

Considerações finais sobre interfaces para dispositivos móveis

Esperamos que, com este material, você tenha uma noção do que é necessário para construir uma interface para dispositivos móveis, bem como suas etapas, métodos, técnicas e processos.

Como dito, o propósito dessa disciplina não é formar designers de interface, mas oferecer uma referência para demandar ou mesmo desenvolver uma interface por conta própria.

Obrigado!

Referências

GOTHELF, Jeff. *Lean UX: Applying Lean Principles to Improve User Experience*. O'Reilly Media. 2013.

HOOBER, Steven. *Designing Mobile Interfaces*. O'Reilly.

NIELSEN, Jakob, BUDIU, Raluca. *Mobile Usability*. 2012. 1ª edição.

NEIL, Theresa. *Mobile Design Pattern Gallery*. O'Reilly. 1ª edição.