


Computação para dispositivos móveis

Interfaces de usuário e Usabilidade



“A interface do usuário de uma WebApp é a sua “primeira impressão”. Independentemente do valor de seu conteúdo, da sofisticação de suas capacidades de processamento e serviços, e do benefício geral da própria WebApp, uma interface mal projetada desapontará o usuário em potencial e pode, de fato, fazer com que o usuário vá para outro lugar. Devido ao grande volume de WebApps concorrentes em praticamente qualquer área, a interface precisa atrair um usuário em potencial imediatamente.”

S., PRESSMAN, R., LOWE, Brian. *Engenharia Web*. LTC, 08/2009.

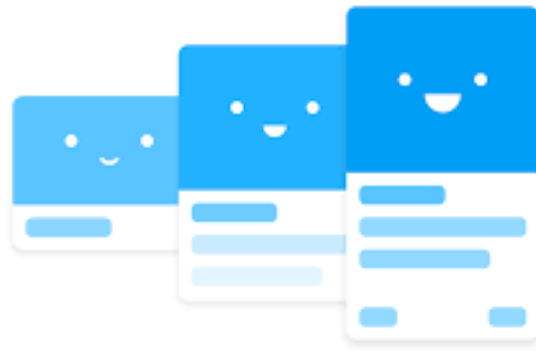
Características Básicas uma Interface de Usuário

- Interfaces eficazes são visualmente evidentes e tolerantes a erros, inculcando em seus usuários um sentido de controle. Os usuários rapidamente veem a profundidade de suas opções, descobrem como alcançar seus objetivos e realizar seu trabalho.
- Interfaces eficazes não incomodam o usuário com o funcionamento interno do sistema. O trabalho é cuidadosa e continuamente salvo, com a opção total para o usuário desfazer qualquer atividade a qualquer momento.
- Aplicações e serviços eficazes realizam um máximo de trabalho, enquanto exigem um mínimo de informação dos usuários.

Tognozzi, Bruno

Inculcar: Passar dados ou informações acerca de; informar

Interface



- vários significados
- mobile User Interface (UI)
- exibição gráfica e geralmente sensível ao toque em um dispositivo móvel
- permite ao usuário interagir com os aplicativos, recursos, conteúdos e funções do dispositivo
- tamanho reduzido + controles de toque



Princípios a Aplicar no Projeto de Interfaces

Antecipação

Uma aplicação deve ser projetada de modo que antecipe o próximo movimento do usuário.

Por exemplo, considere uma aplicação de suporte ao cliente, desenvolvida por um fabricante de impressoras de computador. Um usuário solicitou um objeto de conteúdo que apresenta informações sobre um driver de impressora para um sistema operacional recém lançado. O projetista da aplicação deve antecipar que o usuário poderia solicitar um download do driver e deve oferecer facilidades de navegação que permitam que isso aconteça sem exigir que o usuário procure essa capacidade.

Aplicativos de banco que ao entrar verificam o clipboard e se encontrar um código digitável de um boleto já direcionam para o pagamento.

Comunicação

A interface deverá comunicar o status de qualquer atividade iniciada pelo usuário.

A comunicação pode ser óbvia (por exemplo, uma mensagem de texto) ou sutil (por exemplo, uma folha de papel movendo-se por uma impressora para indicar que a impressão está a caminho).

A interface também deverá comunicar o status do usuário (por exemplo, a identificação do usuário) e o local dentro da hierarquia de conteúdo da aplicação.

Consistência

O uso de controles de navegação, menus, ícones e estética (por exemplo, cor, forma, leiaute) deverá ser consistente em toda a aplicação.

Por exemplo, se o texto azul sublinhado implica um link de navegação, o conteúdo nunca deverá incorporar texto azul sublinhado que não implique um link. Além disso, um objeto, digamos, um triângulo amarelo, usado para indicar uma mensagem de advertência antes que o usuário chame uma função ou ação em particular, não deverá ser usado para outras finalidades em outra parte da aplicação. Finalmente, cada recurso da interface deverá responder de uma maneira que seja consistente com as expectativas do usuário.

Autonomia Controlada

A interface deverá facilitar o movimento do usuário pela aplicação, mas deve fazer isso de uma maneira que garanta as convenções de navegação que foram estabelecidas para a aplicação.

Por exemplo, a navegação por partes protegidas da aplicação deve ser controlada por um ID de usuário e uma senha, e não deve haver mecanismo de navegação que permita um usuário contornar esses controles.

Eficiência

O projeto de uma WebApp ou App e sua interface deverão otimizar a eficiência do trabalho do usuário, e não a eficiência do engenheiro Web que a projetou e construiu, ou do ambiente cliente-servidor que a executa.

Flexibilidade

A interface deverá ser flexível o suficiente para permitir que alguns usuários realizem tarefas diferentemente e que outros explorem a aplicação de forma mais ou menos aleatória.

Em cada caso, ela deve permitir que os usuários entendam onde estão e fornecer aos usuários a funcionalidade de desfazer erros e retornar de caminhos de navegação mal escolhidos.

Foco

A interface da aplicação (e o conteúdo que ela apresenta) deverá permanecer focada na(s) tarefa(s) que o usuário tem mãos.

Em toda hipermídia (WebApp) existe uma tendência de direcionar o uso a um conteúdo fracamente relacionado. Por quê? Porque isso é muito fácil de se fazer!

O problema é que os usuários podem rapidamente ficar perdidos em muitas camadas de informação de apoio e perder de vista o conteúdo original que eles queriam em primeiro lugar.

A interface da App/WebApp e seu conteúdo deverão permanecer focados em qualquer que seja o objetivo que os usuários precisem atingir.

Lei de Fitt

“O tempo para se chegar a um alvo é uma função da distância até o alvo e do seu tamanho”

Com base em um estudo realizado na década de 1950 [Fit54], a lei de Fitt “é um método eficaz de modelar movimentos rápidos, direcionados, onde um apêndice (como uma mão) começa em repouso, em uma posição inicial específica, e se move para terminar dentro de uma área de destino.” Se uma sequência de seleções ou entradas padronizadas (com muitas opções diferentes dentro da sequência) for definida por uma tarefa do usuário, a primeira seleção (por exemplo, pegar o mouse) deverá estar fisicamente próxima da seleção seguinte.

Objetos de interface de usuário

Uma vasta biblioteca de objetos (e padrões) reutilizáveis de interface humana foi desenvolvida para Apps/WebApps. Use-os.

Qualquer objeto de interface que pode ser “visto, ouvido, tocado ou percebido de alguma outra maneira” por um usuário final pode ser adquirido de qualquer uma de uma série de bibliotecas de objetos.

Redução da Latência

Em vez de fazer o usuário esperar até que alguma operação interna termine (por exemplo, o downloading de uma imagem gráfica complexa), a WebApp deverá usar a capacidade multitarefa da aplicação de modo a permitir que o usuário prossiga com o trabalho como se a operação tivesse terminado.

Além de reduzir a latência, os atrasos precisam ser reconhecidos, de modo que o usuário entenda o que está acontecendo. Isso inclui:

- **fornecer retroalimentação de áudio** (por exemplo, um clique ou som de sino) quando uma seleção não resulta em uma ação imediata pela WebApp,
- **exibir um relógio animado ou barra de progresso** para indicar que o processamento está sendo realizado e
- **oferecer algum entretenimento** (por exemplo, uma animação ou apresentação de texto) enquanto ocorre um processamento demorado.

Capacidade de Aprendizado

Uma interface de aplicação deverá ser projetada para minimizar o tempo de aprendizado e, uma vez aprendida, minimizar o reaprendizado exigido quando a aplicação for visitada novamente.

Em geral, a interface deverá enfatizar um projeto simples, intuitivo, que organiza o conteúdo e a funcionalidade em categorias que são óbvias ao usuário.

Metáforas

Uma interface que usa uma metáfora de interação é mais fácil de se aprender e mais fácil de se usar, desde que a metáfora seja apropriada para a aplicação e o usuário.

Uma metáfora deverá evocar imagens e conceitos da experiência do usuário, mas não precisa ser uma reprodução exata de uma experiência do mundo real.

Por exemplo, um site de comércio eletrônico que implementa pagamento de contas automatizado para uma instituição financeira usa uma metáfora de talão de cheques ou boleto/DDA (não é surpresa) para auxiliar o usuário a especificar e programar os pagamentos de conta.

Mantenha a integridade do produto de trabalho

Um produto de trabalho (por exemplo, um formulário preenchido pelo usuário, uma lista especificada pelo usuário) deve ser salvo automaticamente, de modo que não se perca se houver um erro. Cada um de nós já experimentou a frustração associada ao preencher um formulário da App/WebApp extenso e perder todo o conteúdo só por causa de um erro (feito por nós, pela WebApp ou na transmissão do cliente ao servidor).

Para evitar isso, uma App/WebApp deve ser projetada para salvar automaticamente todos os dados especificados pelo usuário.

A interface deverá apoiar essa função e oferecer ao usuário um mecanismo fácil para recuperar informações “perdidas”.

Legibilidade

Toda a informação apresentada pela interface deverá ser legível por jovens e idosos (ou portadores de necessidades especiais).

O projetista da interface deverá enfatizar escolhas de estilo de texto, tamanho de fonte e cor de segundo plano legíveis, que melhoram o contraste.

Acompanhe o estado

Quando apropriado, o estado das interações do usuário deve ser acompanhado e armazenado de modo que os usuários possam fazer o logoff e retornar mais tarde para continuar de onde pararam. Em geral (para webApps), os cookies podem ser utilizados para armazenar informações de estado. Porém, os cookies são uma tecnologia controvertida, e outras soluções de projeto podem ser mais aceitas por alguns usuários.

Navegação visível

Uma interface de WebApp bem projetada oferece “a ilusão de que os usuários estão no mesmo lugar, com o trabalho trazido a eles” [Tognozzi].

Quando essa técnica é usada, a navegação não é uma preocupação para o usuário. Ao invés disso, o usuário recupera objetos de conteúdo e seleciona funções que são exibidas e executadas pela interface.

Acessibilidade

Nossas Apps são Acessíveis ?

- Diretrizes W3C

<http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html>

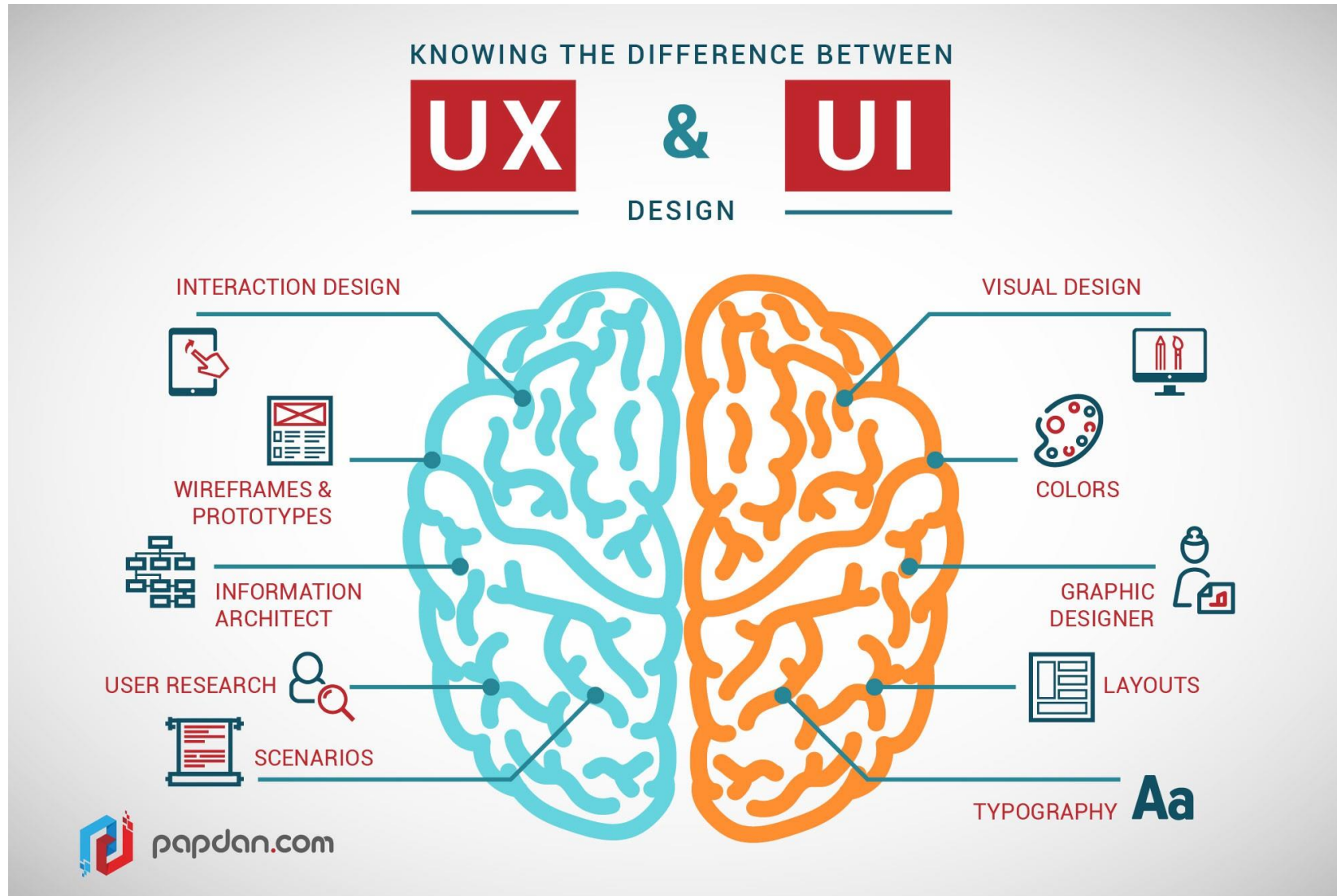
- Acessibilidade Legal

<http://www.acessibilidadelegal.com/>

UX - User Experience

audaciosamente indo onde
nenhuma
Interface jamais esteve. ...

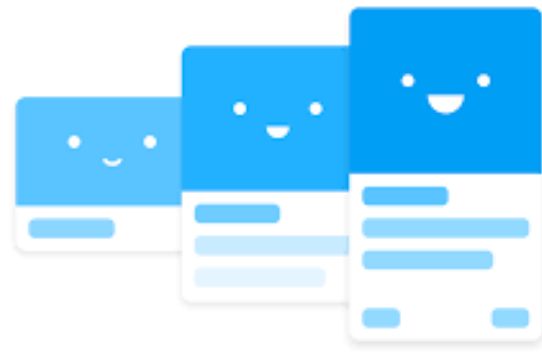
UX - User Experience



A large, irregular blue ink splatter or blotch serves as the background for the text. The splatter has a textured, painterly appearance with various shades of blue and some white highlights, giving it a dynamic and artistic feel. It is centered on a plain white background.

**Projeto de
Aplicativos**

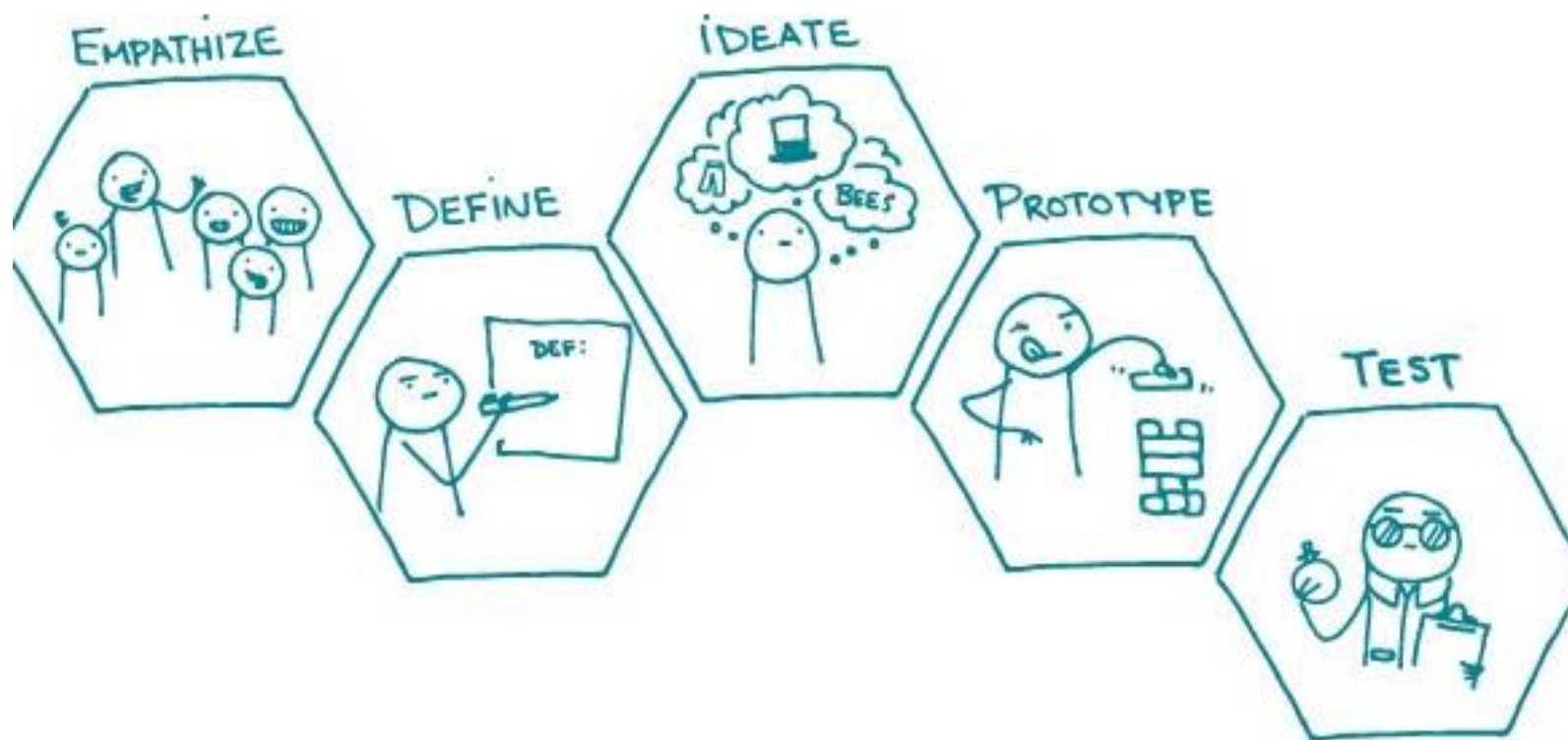
**Definição de
Interface**



Interface

- vários significados
- mobile User Interface (UI)
- exibição gráfica e geralmente sensível ao toque em um dispositivo móvel
- permite ao usuário interagir com os aplicativos, recursos, conteúdos e funções do dispositivo
- tamanho reduzido + controles de toque

DESIGN THINKING!



Design Thinking

Conjunto de métodos e processos para abordar problemas, relacionados a futuras aquisições de informações, análise de conhecimento e propostas de soluções

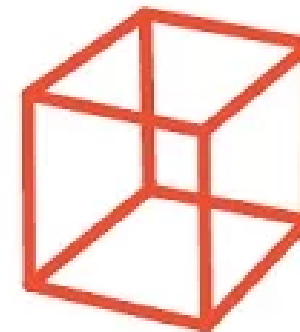
As a <user role>

I want <goal>

so that <benefit>.

User Stories

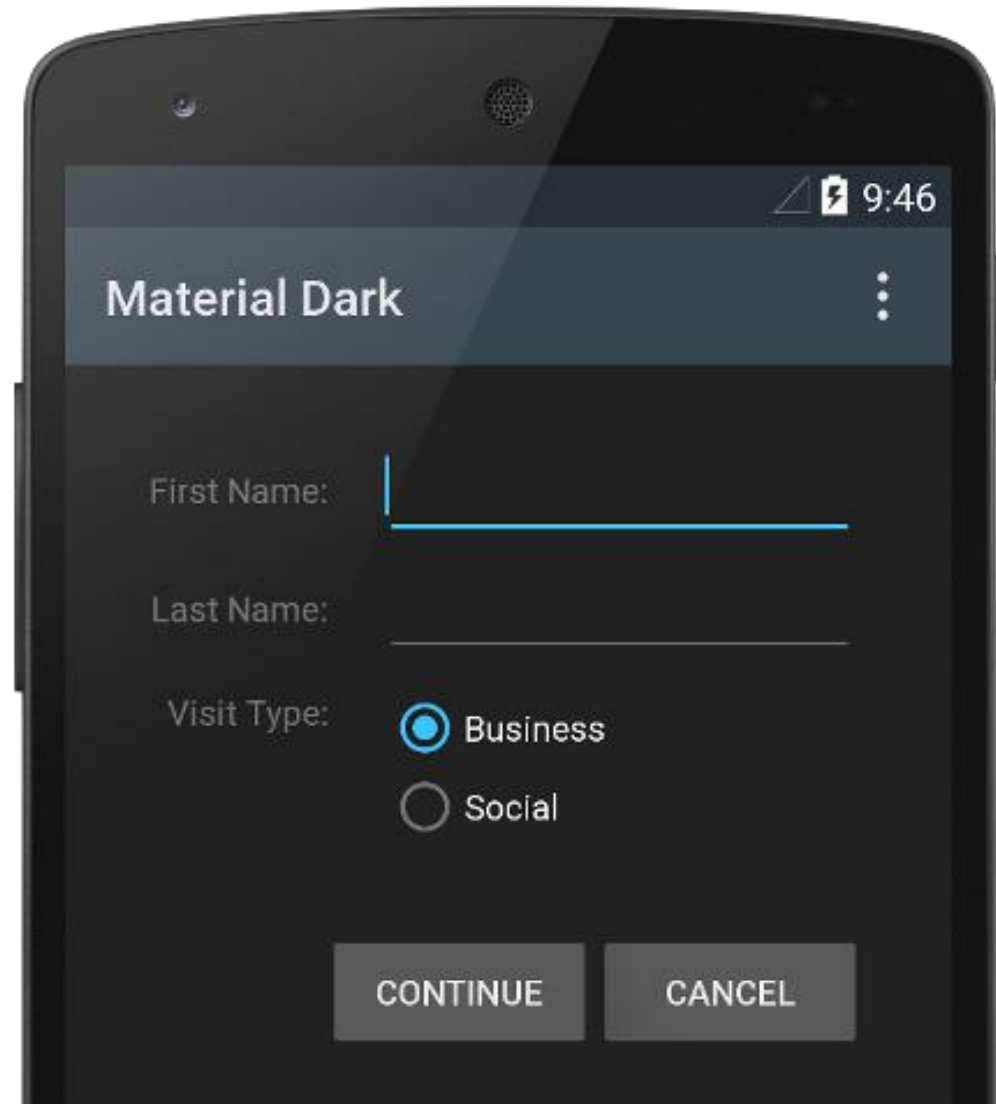
Descrição concisa de uma necessidade do usuário do produto (ou seja, de um “requisito”) sob o ponto de vista desse usuário



Prototipação

Processo fundamental onde as soluções recém-criadas são tangibilizadas, testadas e verificadas na prática

Principais Componentes de Interfaces Mobile



Android e o Material Design

- Material Design
 - Há alguns anos já é o padrão para interfaces Android (<https://developer.android.com/design/material/?hl=pt-br>)
- Linguagem visual que sintetiza os princípios clássicos do bom design com a inovação da tecnologia e da ciência

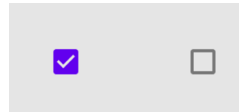
Saiba mais em :

https://www.youtube.com/watch?time_continue=65&v=YaG_ljfzeUw

240 componentes ! <https://material.io/design/guidelines-overview/#addition>

Controles para entradas

checkboxes



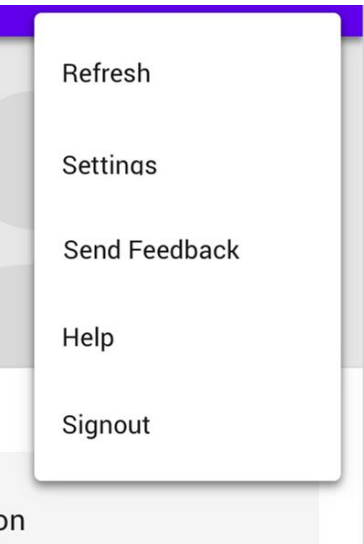
radio buttons



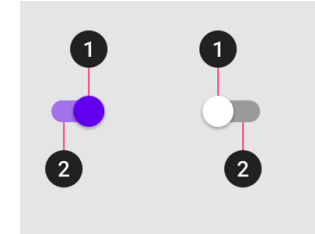
buttons



dropdown lists



switches / toggles



text fields





Controles de navegação

breadcrumbs

sliders

search fields

pagination

tags

icons



Controles de informação

tooltips

icons

progress bar

notifications

message boxes

modal windows



Containers

accordions

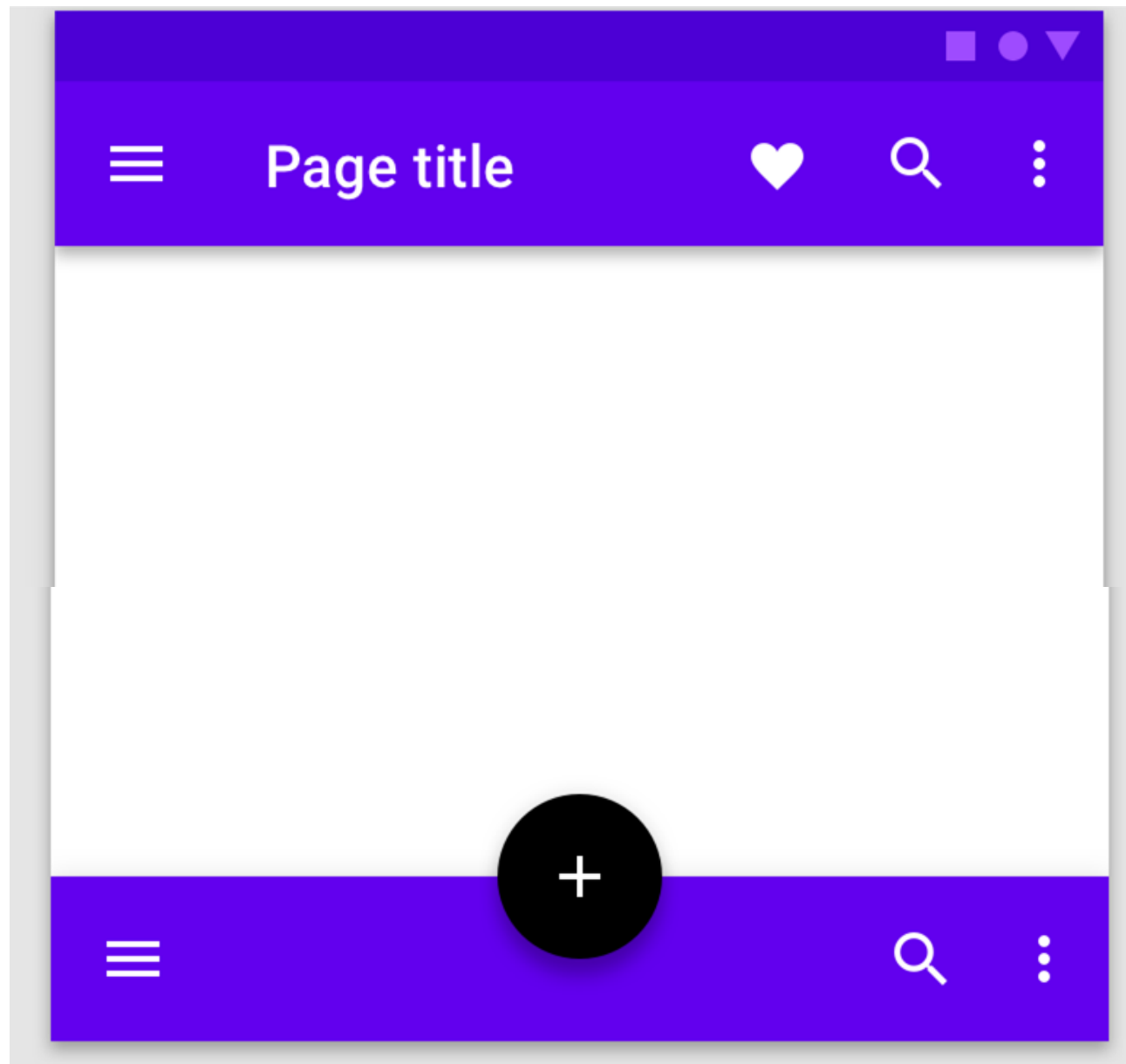
cards



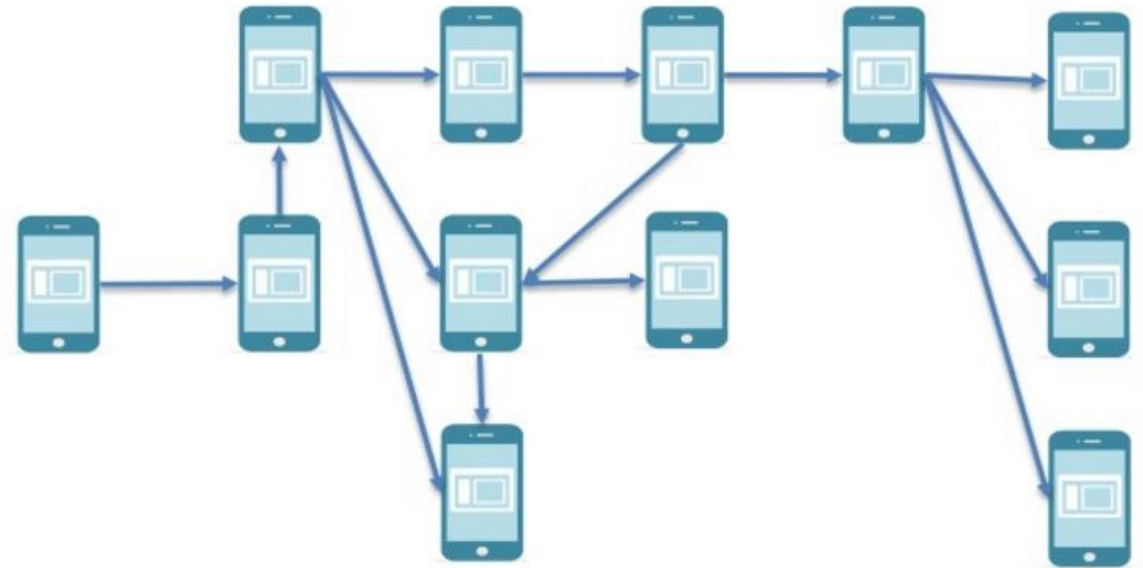
Bars

AppBar de baixo

AppBar de cima



Arquitetura de um Aplicativo

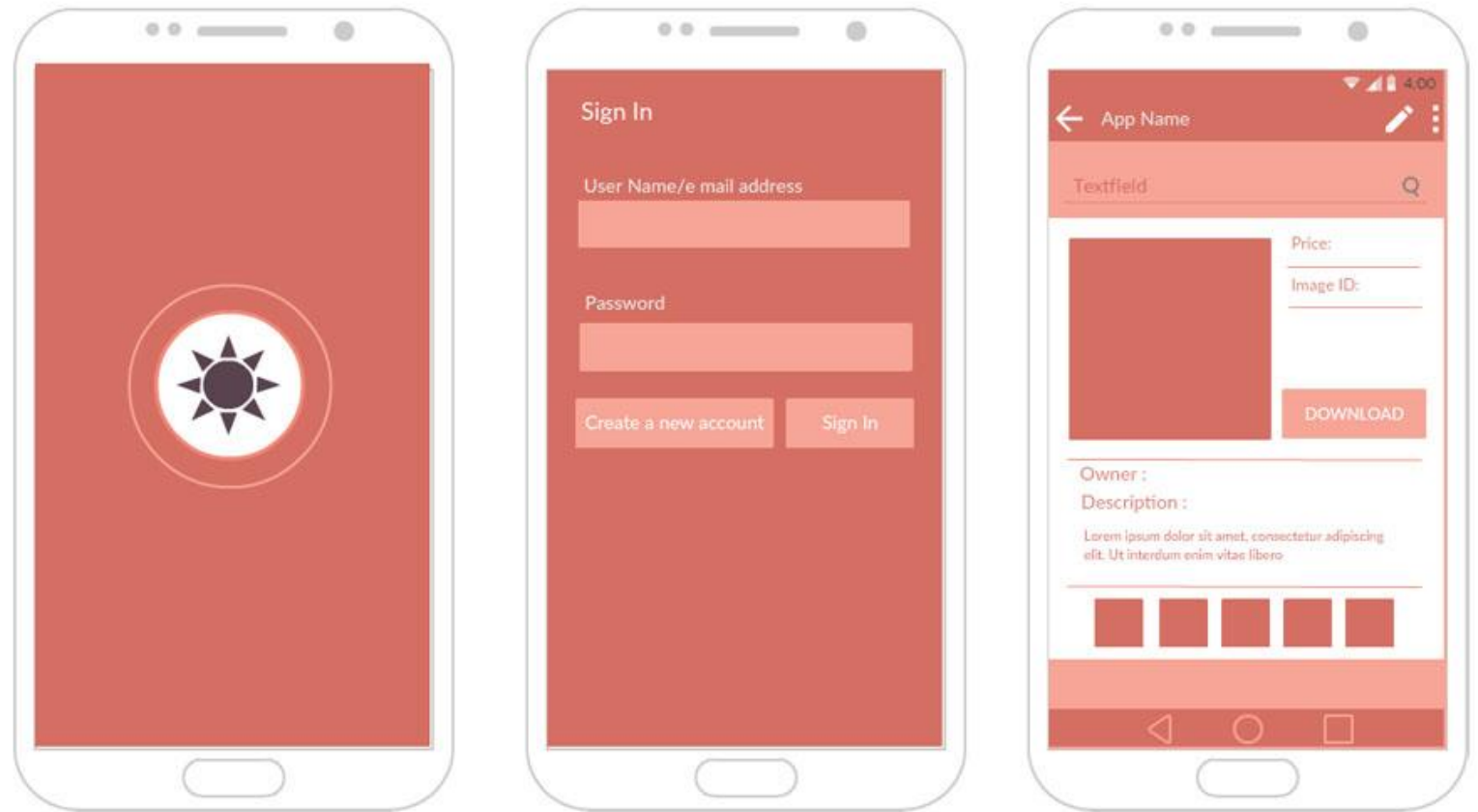


Arquitetura

- Splash screen - tela introdutória
- Primeiro acesso
- Tela principal
- Menus / Bars
- Funcionalidades
- Acesso a outros serviços / apps

- **FOCO NA EXPERIÊNCIA DO USUÁRIO (UX)**

Três telas padrão



Ferramentas de prototipação

- <https://www.fluidui.com/>
- <https://marvelapp.com/>
- <https://www.figma.com/>
- Ninjamock (<https://ninjamock.com/>)
- Mockflow (<https://www.mockflow.com/>)
- Mockingbot (<https://mockingbot.com/>)
- Framebox (<http://framebox.org/>)
- Moqups (<https://moqups.com/>)
- Balsamiq (<https://balsamiq.cloud/>)
- Microsoft Visio
- Lucidchart
- Microsoft Powerpoint
- Papel e caneta

Atividade 01

Em no máximo **quartetos**, construir o protótipo de um aplicativo

Mínimo de 4 telas usando componentes diferentes, que façam sentido, claro

Usar uma ferramenta de prototipação

Entregar o protótipo no BlackBoard