



Interação Humano Computador

Bruno Felipe de França Souza

Curso Técnico em Informática

Educação a Distância

2016



EXPEDIENTE

Professor Autor

Bruno Felipe de França Souza

Design Instrucional

Deyvid Souza Nascimento
Maria de Fátima Duarte Angeiras
Renata Marques de Otero
Terezinha Mônica Sinício Beltrão

Revisão de Língua Portuguesa

Letícia Garcia

Diagramação

Izabela Cavalcanti

Coordenação

Anderson Elias

Coordenação Executiva

George Bento Catunda

Coordenação Geral

Paulo Fernando de Vasconcelos Dutra

Conteúdo produzido para os Cursos Técnicos da Secretaria Executiva de Educação
Profissional de Pernambuco, em convênio com o Ministério da Educação
(Rede e-Tec Brasil).

Setembro, 2016

Catálogo na fonte

Bibliotecário Hugo Carlos Cavalcanti, CRB4-2129

S719i

Souza, Bruno Felipe de França.

Interação Humano Computador: Curso Técnico em Informática: Educação a distância / Bruno Felipe de França Souza. – Recife: Secretaria Executiva de Educação Profissional de Pernambuco, 2016.

36 p.: il.

Inclui referências bibliográficas.

1. Educação a distância. 2. Ciência da Computação. 3. Interação homem-computador. 3. Design de Interação. I. Souza, Bruno Felipe de França. II. Título. III. Secretaria Executiva de Educação Profissional de Pernambuco. IV. Rede e-Tec Brasil.

CDD – 004.5



Sumário

Introdução	6
1.Competência 01 Introdução à Interação Humano-Computador	7
1.1 Interação Humano-Computador (IHC).....	7
1.2 Interface.....	10
1.3 Interface e interação	11
1.4 Desenvolvimento de uma Interface	14
2.Competência 02 Processo de Desenvolvimento de Aplicativos	16
2.1 Requisitos.....	16
2.2 Tipos de requisitos.....	17
2.3 Documentando os requisitos	19
2.4 Projeto de software	20
2.5 Etapas envolvidas no projeto de software.....	22
2.6 Resolução de tela	23
2.7 Cores	23
2.8 Disposição dos elementos na tela	24
3.Competência 03 Aplicação de Conceitos de Usabilidade na Construção de uma Interface Móvel	27
3.1 Os 7 mandamentos para construção de uma interface	27
3.2 Construção de uma interface para dispositivos móveis	28
Conclusão.....	35
Referências	36
Minicurriculo do Professor	37



Introdução

Olá, tudo bem com você?

Nesta disciplina vamos aprender conceitos que irão nos ajudar a desenvolver *software* que seja mais fácil para a interação dos usuários. A área de Interação Humano-Computador (IHC) é a responsável por esses conhecimentos. Iremos tratar mais especificamente da usabilidade em dispositivos móveis, pois esses vêm sendo utilizados em maior escala atualmente e a tendência é aumentarem cada vez mais. Só no Brasil temos mais celulares do tipo *smarthphones* do que pessoas, sem falar dos *tablets*.

Ao desenvolver um *software*, sempre é importante sabermos o público-alvo que irá usá-lo, para assim o produto atender melhor a necessidade desse grupo. Além disso, é importante saber aspectos como cores a serem usadas no projeto gráfico, disposição dos elementos na tela, o que colocar na tela para o usuário, entre outros.

Como vamos tratar de dispositivos móveis, temos que levar em consideração que o espaço da tela que o usuário irá interagir é menor que o computador de mesa ou notebook. Também o usuário de um dispositivo móvel como o celular tem a atenção perdida rapidamente pelo motivo de estar usando o dispositivo em lugares diferentes a cada momento: no transporte, no shopping, na praça, na praia. Temos que saber bem como atrair e manter a atenção do usuário ao usar o software em um dispositivo móvel.

Bom, na disciplina iremos começar conhecendo um pouco sobre a área de IHC e depois nos concentraremos na usabilidade de dispositivos móveis, finalizando com a prática envolvendo os conceitos estudados.

Então? Vamos começar?



1.Competência 01 | Introdução à Interação Humano-Computador

Nesta seção iremos falar um pouco sobre a área de IHC e como ela é importante na concepção de interfaces gráficas que irão ser manipuladas pelo usuário do software. Depois, vamos ver como chegamos à era dos dispositivos móveis e a devida atenção que devemos dar para a concepção de interfaces para os usuários.

Percebeu as palavras **software** e **interface**? Não se preocupe porque mais a frente você irá aprender cada uma delas.

Então, pronto para começar?

1.1 Interação Humano-Computador (IHC)

Primeiramente, gostaria de fazer uma pergunta a você, convidá-lo a refletir um pouco:

O que é IHC?

A interação Humano-Computador começou a ser desvendada há um bom tempo, em meados dos anos 80, com o intuito de apresentar esse novo campo de estudo. O termo surgiu da necessidade de mostrar que o foco de interesse é mais amplo do que apenas construir **interfaces** e abrange todos os aspectos relacionados com a interação entre usuários e computadores.

Interface é algo que fica entre o usuário e o dispositivo móvel, algo feito para o usuário manipular o elemento que está sendo apresentado. Por exemplo, a página do Facebook é a interface e a tela do dispositivo é o meio por onde interagimos



Como já sabemos o que é **interface**, a interação é feita por meio do diálogo entre os usuários e o sistema. Esse diálogo se dá por meios da interface. Conforme mostra a Figura 1, os usuários enviam as informações para o sistema e esperam a resposta do sistema, ou seja, em outras palavras, o usuário executa uma ação e o sistema responde.

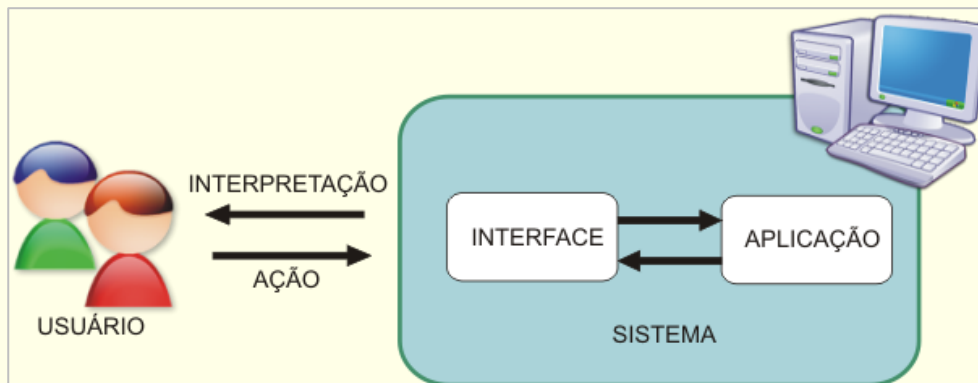


Figura 1 - Processo de interação humano-computador.

Fonte: <http://tiqx.blogspot.com.br/2012/02/compreenda-importancia-da-interacao.html>

Descrição: A interação começa com o usuário que requisita uma ação ao sistema por meio de uma interface. O sistema irá processar a requisição e devolver uma resposta ao usuário que será entregue pela interface

A IHC refere-se a uma área de pesquisa que tem por objetivo estudar a comunicação existente entre as pessoas e os sistemas de computadores. Desde o primeiro computador, o ENIAC, a relação entre o homem e o computador vem se estreitando. O surgimento dos microcomputadores e dos novos sistemas operacionais possibilitou o uso de interfaces que simplificaram a interação entre eles. Sendo assim, podemos dizer que a IHC é o estudo da interação entre pessoas e computadores e de tudo que está envolvido nesta relação.



Para saber mais sobre o primeiro computador, o ENIAC, acesse a página abaixo:
<https://pt.wikipedia.org/wiki/ENIAC>



Para entender de uma forma mais fácil: a área de IHC se preocupa em estudar **como** nós interagimos com os sistemas computacionais apresentados nas mais diversas facetas, procurando fazer com que a interação seja simples e natural.

A IHC é uma área totalmente multidisciplinar (conforme mostra a Figura 2) que envolve áreas como: Ciência da Computação, Psicologia, Ergonomia, Fatores Humanos, Linguística, Inteligência artificial, Engenharia, dentre outras. A IHC está voltada para a aplicação do conhecimento destas disciplinas para produzir interfaces amigáveis. Não é uma disciplina essencialmente voltada para o estudo de computação ou do ser humano, mas para a comunicação entre estas duas entidades. Conhecimento sobre as limitações da capacidade humana e restrições da tecnologia existente devem ser ponderados para oferecer ao usuário um meio adequado através do qual eles podem interagir com os computadores.



Figura 2 - Disciplina que formam IHC.

Fonte: Preece, 1994. <http://oldwww.acm.org/perlman/preece.html>

Descrição: IHC é multidisciplinar e envolve áreas como: Ciência da Computação, Psicologia, Ergonomia, Fatores Humanos, Linguística, Inteligência artificial, Engenharia, dentre outras.

Na próxima seção iremos conhecer mais a fundo o que é uma interface, como alguns autores a



definem e alguns exemplos.

1.2 Interface

Quando vamos interagir com um dispositivo móvel, por exemplo, precisamos ter um meio para enviar os comandos a serem executados. Esse "meio" que usamos é chamado comumente de *interface*.

Alguns autores definem interface como:

- Parte do sistema com o qual o usuário entra em contato físico, perceptível e cognitivo.
- Imagem do sistema.
- Ambiente virtual para interatividade.
- Mensagem do *designer* (profissional que produz algo) para o usuário.

Um dos princípios básicos quando tratamos da construção de uma interface é o conceito de **interface amigável**. Uma interface amigável é algo onde o usuário interage e parece ser invisível. Algo que deve ser fácil de usar e de aprender, reduzindo custos com treinamento e uso diário dos usuários. Alguns pontos importantes sobre interfaces amigáveis são:

- Prevenir ou minimizar taxa de erros do usuário;
- Recordação rápida e ser atrativa (os usuários decidem não usar um software por causa da interface);
- O desenvolvimento da interface corresponde a um alto custo no desenvolvimento do sistema como um todo.



Figura 3 - Interface do Facebook para dispositivos móveis.

Fonte: <http://tecnologia.umcomo.com.br/articulo/como-desativar-o-facebook-movel-13221.html>

Descrição: interface do site Facebook para dispositivos móveis, onde o usuário insere seu login e senha.

Na próxima seção vamos descobrir como **interagimos** por meio das **interfaces**.

1.3 Interface e interação

Sempre que precisamos nos comunicar com o sistema para dizer o que queremos fazer, é iniciado um processo de **interação**. Essa interação é a capacidade do sistema em dar respostas a nós usuários e, ao mesmo tempo, envolve a capacidade do usuário de se comunicar com o sistema.

Por exemplo, atualmente, a forma de interação com os dispositivos móveis é baseada no conceito de *touchscreen*, ou seja, toque na tela. A partir do toque enviamos comandos ao sistema e o mesmo nos devolve uma resposta. Então, fazendo um paralelo entre interface e interação:

Interface:

- Aquilo que interliga dois sistemas;
- Software e hardware para comunicação entre usuário e sistema.



Interação:

- Processo de comunicação usuário-sistema;
- Processo que engloba as ações dos usuários no sistema e interpretações dos resultados.

A interação possui o sentido em que ela trafega, podendo ser:

- Do usuário para o sistema: a forma mais comum onde o usuário faz uma ação e espera um resultado.
- Do sistema para o usuário: quando o usuário entra com uma ação e o sistema fornece o resultado (a saída).
- Do sistema para o sistema: quando o sistema se comunica com outro sistema para realizar uma operação. Por exemplo: quando você publica uma foto no Instagram e pede para ser publicada também no Facebook. Há uma interação entre dois sistemas.



Um aplicativo que usa realidade aumentada e que está sendo uma febre entre os usuários é o Pokémon Go:

https://pt.wikipedia.org/wiki/Pokémon_GO

A nossa interação com o sistema pode ser feita de várias maneiras, algumas delas são: menus, teclas rápidas (atalhos), preenchimento de formulários, linguagem de comando, questão/resposta, linguagem natural, manipulação direta, realidade virtual.

Quando estamos interagindo com o sistema por meio de **menus** temos uma lista de opções organizadas que facilitam o acesso às funcionalidades do sistema para usuários com pouco treinamento. Podemos ver um exemplo de menu na Figura 4.



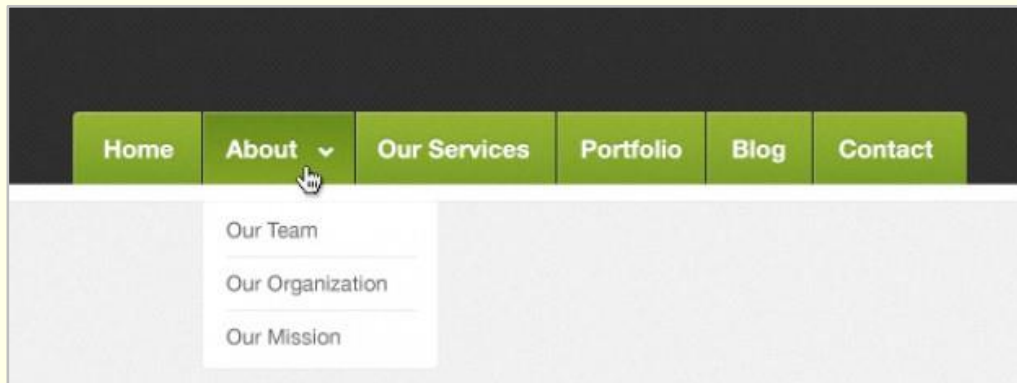


Figura 4 - Exemplo de um menu.

Fonte: http://br.freepik.com/psd-gratis/menu-de-navegacao-por-abas_595642.htm

Descrição: tipo de navegação por menu horizontal.

Os **atalhos** são uma forma de interação usados por usuários mais experientes com o sistema. Combinações de teclas para acionar diretamente uma função do sistema. Traz mais **agilidade**. Exemplos: CTRL + C para copiar , CTRL + V para colar.

Os **formulários** são bem utilizados para preencher informações e enviar para o sistema. Baseados na digitação e preenchimento de campos, formulários eletrônicos são similares aos de papel e geralmente complementados por menus.

Outra forma de interação que está sendo muito usada na atualidade é a **realidade aumentada**. Nesse tipo de interação os usuários interagem com elementos que são sobrepostos no mundo real. É um tipo de interação mais avançada e que exige dispositivos como sensores para aumentar o grau de interação com os usuários.

Na Figura 5 podemos ver um exemplo de um elemento (urso) se sobrepondo no mundo real (mão do usuário). Esse exemplo ilustra muito bem o conceito de realidade virtual aumentada.





Figura 5 - Exemplo de realidade aumentada.

Fonte:

https://en.wikipedia.org/wiki/Augmented_reality#/media/File:AR_EdiBear0001390_1.jpg

Descrição: Sobreposição de um elemento (urso) no mundo real (mão do usuário)

Agora vamos dar uma lida em como é feito o processo de desenvolvimento de uma interface.

1.4 Desenvolvimento de uma Interface

Bom, agora que já vimos um pouco de IHC e conversamos sobre interfaces e interação, vamos refletir um pouco:

Como é feito o desenvolvimento de uma interface de usuário para um dispositivo móvel?

Para começarmos a desenvolver um projeto de interface para o usuário, devemos levar em consideração alguns fatores:

Perfil do usuário	Para quem?
Dispositivos de Interação	Como?
Tarefas	O quê? Quando?

Em relação ao **perfil do usuário**, temos que levar em consideração quem será nosso **público alvo**, ou seja, quais grupos de pessoas irão usar nosso aplicativo? Isso é muito importante para sabermos como escolher as cores, tamanho de fonte, disposição dos elementos na tela e até mesmo onde inserir o aplicativo. Os usuários possuem diferentes características, são elas: físicas, cognitivas, culturais e organizacionais.



O **dispositivo de interação** é uma figura importante também, no nosso caso, um **dispositivo móvel**. Dessa forma, temos que levar em consideração o tamanho da tela que será reduzida, conexão com a internet que pode ser limitada ou mais lenta, o meio onde o usuário está inserido, pois ele pode estar usando o dispositivo móvel em qualquer lugar.

Outro ponto importante são as **tarefas** que o aplicativo irá executar. Exibição de texto, de imagens, de vídeos, é um jogo? Todas essas tarefas precisam de implementações e interações diferentes. Geralmente para um jogo usamos as duas mãos para interagir enquanto que para visualização de imagens basta o usuário usar uma das mãos e clicar com um dedo na imagem. Essas tarefas podem ser executadas manualmente, automaticamente ou de forma interativa. Nas tarefas interativas o usuário faz manualmente uma ação que pode desencadear todo um processo automático no sistema.

Trazendo esses conceitos para a nossa realidade, dispositivos móveis, como deveríamos levar em consideração o perfil do usuário e as tarefas tendo em mente o desenvolvimento de um aplicativo? Hoje em dia temos pessoas de diferentes classes sociais e culturais usando dispositivos móveis. Além disso, a forma de interação com esses dispositivos deve ser bastante dinâmica e prender a atenção do usuário. É, realmente, um desafio e tanto!

Como já estudamos a parte de fundamentação teórica de IHC, na próxima semana iremos focar nos passos que devemos tomar para desenvolver aplicativos voltados para os dispositivos móveis, levando em consideração alguns conceitos que aprendemos nesta semana.

Gostaria de sugerir para vocês analisarem a interface de algum aplicativo. Pode ser de algum que vocês usam no celular: Facebook, Instagram ou mesmo algum web site. O que acharam? É fácil de usar e encontrar as funcionalidades? E as cores usadas? É agradável de ler o texto? Vamos discutir um pouco sobre isso durante a semana.

Vamos em frente, pessoal!



2.Competência 02 | Processo de Desenvolvimento de Aplicativos

Esta semana é muito importante. Ela é a base para o próximo assunto que abordaremos na semana seguinte. Seu comprometimento no estudo deve ser total, pois é fundamental para o processo de desenvolvimento de aplicativos para dispositivos móveis.

Aqui, aprenderemos os conceitos da engenharia de software aplicados ao desenvolvimento de aplicativos para a área móvel. Assim, os passos tomados nos levarão a desenvolver um bom aplicativo para um determinado público- alvo.

2.1 Requisitos

Como dito anteriormente, é muito importante sabermos o público-alvo que irá usar o nosso aplicativo. Isso ajuda bastante na escolha da melhor abordagem em relação ao desenvolvimento de sua interface. Após essa fase, entramos na captação dos **requisitos**, área muito importante da engenharia de software.

"Um requisito é uma característica do sistema ou a descrição de algo que o sistema é capaz de realizar para atingir os seus objetivos "

Os requisitos são muito importantes para:

- Estabelecer uma base de concordância entre o cliente e o fornecedor sobre o qual o aplicativo será feito;
- Fornecer uma referência para a validação do produto final;
- Reduzir o custo de desenvolvimento;
- Estabelecer o escopo (abrangência) de um sistema;
- Base para a gestão de projetos de software.



Os requisitos são uma forma essencial de documentarmos o que o cliente precisa que o aplicativo execute e, assim, desenvolvermos apenas o que foi pedido. Também é uma forma do desenvolvedor (ou empresa) e o cliente se resguardar sobre o que foi pedido ou não para ser concebido.

2.2 Tipos de requisitos

Para um melhor entendimento, é preciso classificarmos os requisitos em três tipos: requisitos de usuário, requisitos de sistema e requisitos de software.

Requisitos de Usuário	Representam declarações em linguagem natural e também em diagramas sobre as funções que o sistema deve fornecer e as restrições sob as quais deve operar.
Requisitos de Sistema	Constituem um documento estruturado que estabelece detalhadamente as funções e as restrições de sistema.
Requisitos de Software	Compreendem uma descrição detalhada do software que serve como base para o projeto e a implementação.

Esses requisitos estão intimamente ligados ao público escolhido. Para cada grupo temos interesses específicos em se tratando de requisitos. Os requisitos para desenvolvedores de sistemas **são** diferentes de um requisito para um usuário final, por exemplo, um desenvolvedor precisa saber em qual linguagem de programação o aplicativo deve ser desenvolvido, já para o usuário final isso fica transparente, ele não quer saber dessa informação.



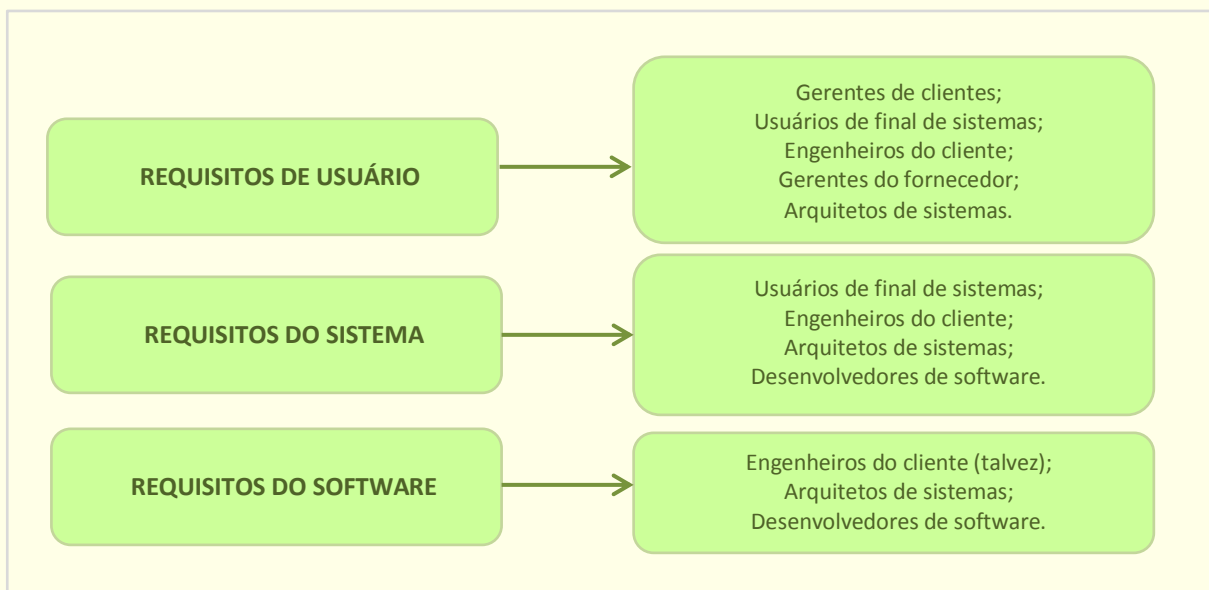


Figura 6 -Público-alvo de cada requisito.

Fonte: próprio autor (2016).

Descrição: público-alvo dos requisitos de usuário: gerentes de clientes, usuários finais de sistemas, engenheiros do cliente, gerentes do fornecedor e arquitetos de sistemas. Público alvo dos requisitos do sistema: usuários finais de sistemas, engenheiros do cliente, arquitetos de sistemas e desenvolvedores de software. Público alvo dos requisitos do software: engenheiros do cliente (talvez), arquitetos de sistemas e desenvolvedores de software.

Os requisitos que acabamos de abordar são classificados de duas formas: **requisitos funcionais (RF)** e **requisitos não funcionais (RNF)**. Basicamente, RF são aqueles que o sistema deve executar, são as funcionalidades que o sistema deve ter de forma obrigatória. Já os RNF, estão ligados ao desempenho do sistema e **podem não estar presentes** em um determinado momento. Para deixar mais claro, vamos ver algumas definições.

- **Requisitos Funcionais:**

- São requisitos diretamente ligados a funcionalidade do software, descrevem as funções que o software deve executar;
- Na definição dos requisitos funcionais o foco é “o que” e não “como”.

- **Requisitos Não Funcionais:**

- São requisitos que expressam condições que o software deve atender ou qualidades específicas que o software deve ter.
- Os requisitos não funcionais determinam restrições no software, como: usabilidade, desempenho, custos, entre outros.



Agora que já conhecemos os tipos e categorias dos requisitos, vamos praticar um pouco. Imagine um cenário onde você irá desenvolver um jogo para um *tablet*. Qual o público-alvo envolvido? Cite alguns RF e RNF que você vislumbra que serão necessários para o jogo. Ao final, diga se esses RF e RNF são requisitos de usuário, sistema e/ou software.

2.3 Documentando os requisitos

Vamos agora olhar o processo geral do gerenciamento de requisitos. Na Figura 7, após capturarmos os requisitos precisamos documentá-los. Essa documentação é feita por meio de um **documento de requisitos ou documento de especificação de software**

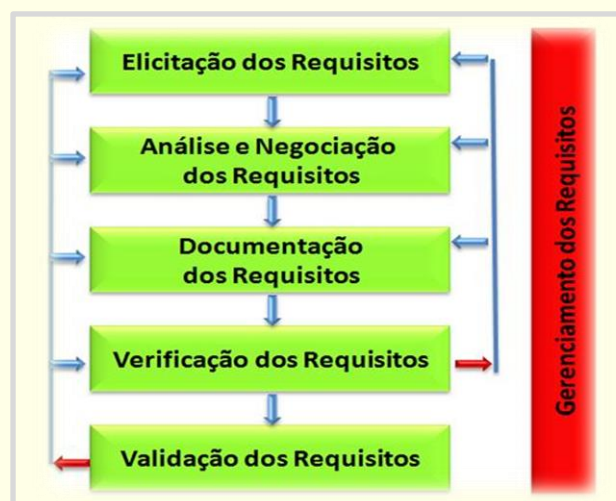


Figura 7.

Fonte: próprio autor (2015).

Descrição: fluxo do gerenciamento de requisitos onde ressaltamos a documentação dos requisitos.

Dessa forma, um documento de requisitos deve descrever detalhadamente os requisitos do sistema. sejam eles requisitos funcionais e não funcionais. Segundo Sommerville, um documento de requisitos deve conter:

- Introdução, com um resumo geral do sistema e principais características;
- Os serviços e as funções que o sistema deve prover;
- As limitações sobre as quais o sistema deve operar e as limitações nos processos usados



para desenvolver o sistema;

- As propriedades gerais do sistema;
- As definições de outros sistemas com o qual o sistema deve se integrar;
- As informações sobre o domínio da aplicação do sistema;
- As descrições sobre o *hardware* no qual o sistema irá executar.

O documento de requisitos é onde iremos descrever um conjunto de requisitos do usuário e do sistema que formaliza o entendimento em várias perspectivas. Isso quer dizer que tanto o cliente como a empresa (ou profissional) estão de acordo com os requisitos contidos no documento.

Os principais interessados no documento de requisitos são:

- Clientes, para confirmar a completude dos requisitos e propor alterações;
- Gestores, para orçar o sistema e planejar o processo de desenvolvimento;
- Engenheiros, para compreender o sistema a ser desenvolvido;
- Engenheiros (testes), para desenvolver testes no intuito de validar o cumprimento dos requisitos;
- Engenheiros (manutenção), para compreender o sistema e a ligação entre as suas partes.

2.4 Projeto de software

Nesse tópico vamos aprender em que momento será realizado o projeto de software, quem é o responsável pelo projeto, a importância das atividades do projeto e os conceitos fundamentais envolvidos.

Vamos entender a importância do projeto de software por meio de uma metáfora. Suponha que João, um respeitado Analista de Sistemas, consegue juntar uma boa grana para realizar seu grande sonho: construir uma casa de campo de dois quartos, bem iluminada e ventilada... João julga que tem conhecimento suficiente e já pensou em todos os detalhes de sua casa dos sonhos, rabisca uma coisinha aqui e outra acolá... e... contrata diretamente os pedreiros, e ele mesmo diz (de boca)

como é a estrutura da casa, começando imediatamente a construí-la...

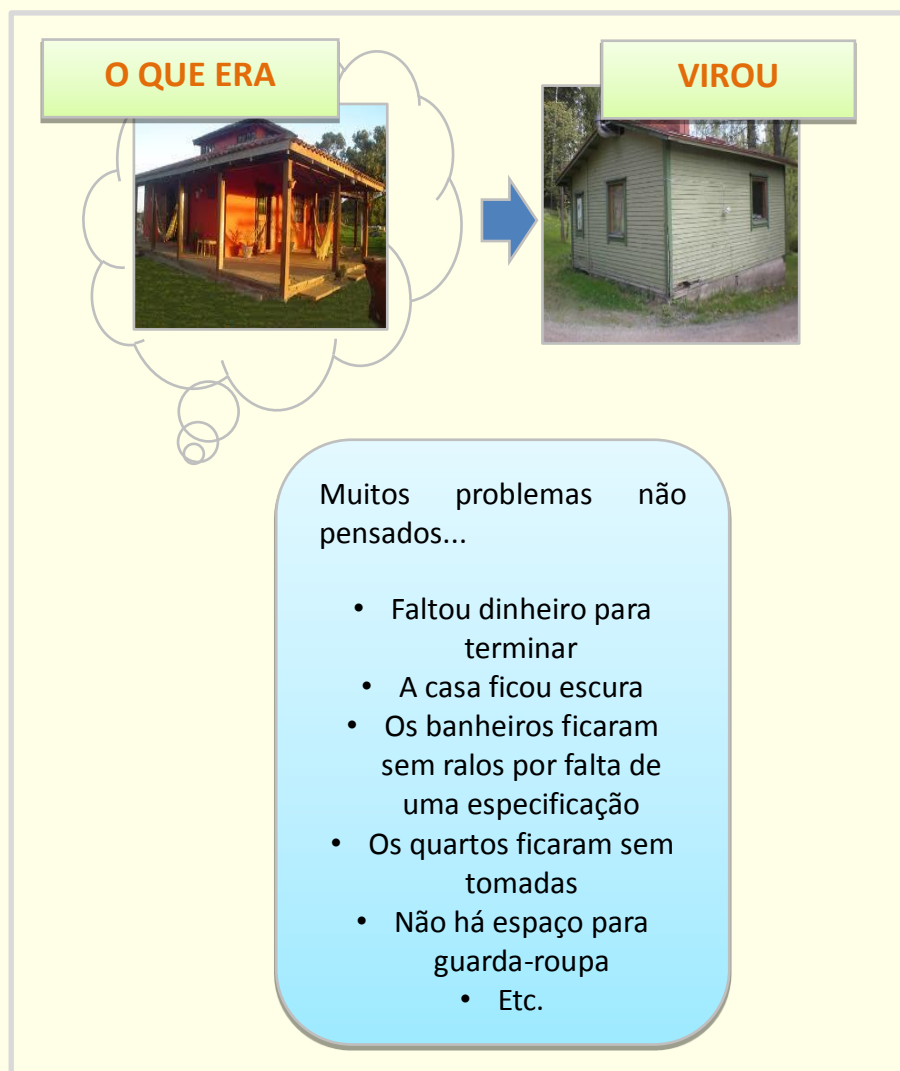


Figura 8.

Fonte: próprio autor (2015).

Descrição: os problemas envolvidos no projeto da casa dos sonhos de João. Faltou dinheiro para terminar, a casa ficou escura, os banheiros ficaram sem ralos por falta de uma especificação, os quartos ficaram sem tomadas e não há espaço para o guarda-roupa.

Uma forma mais apropriada seria João contratar um serviço especializado para fazer o projeto da casa de seus sonhos para só depois passar a construí-la!

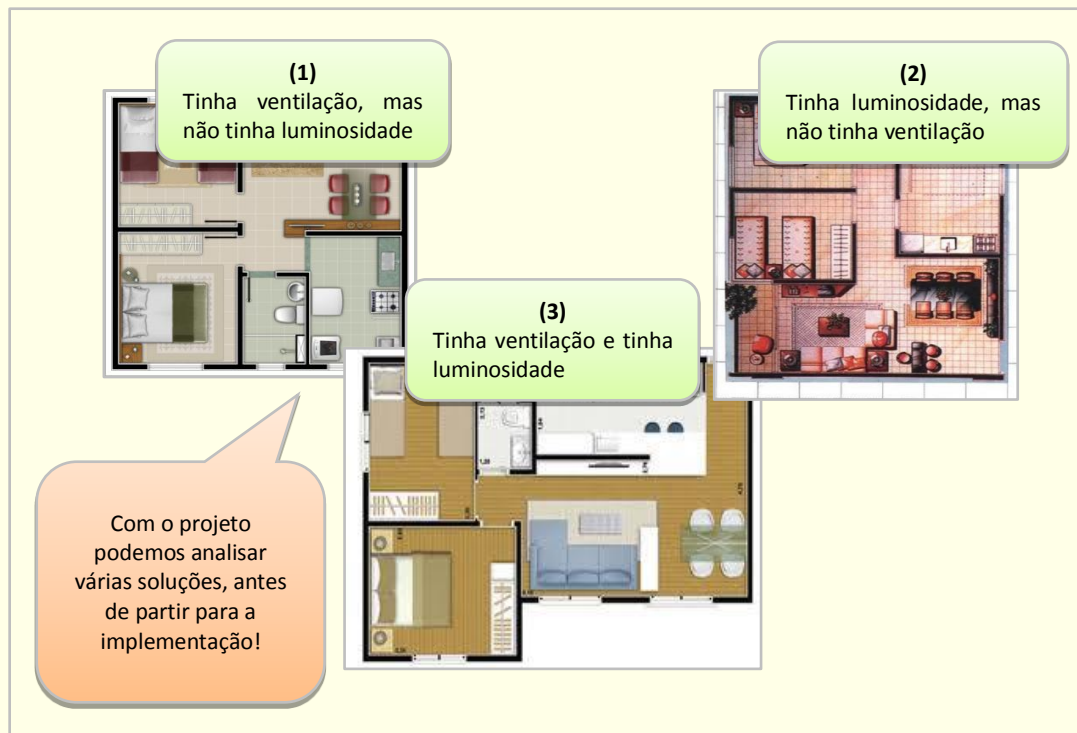


Figura 9.

Fonte: próprio autor (2015).

Descrição: Projeto da casa dos sonhos de João feito de forma correta com o auxílio de especialistas. Com o projeto correto podemos analisar várias soluções, antes de partir para a implementação.

Por meio da metáfora queremos chegar à seguinte conclusão: o projeto ajudou a validar os requisitos da casa dos sonhos de João, ajudou a identificar potenciais defeitos antes de sua implementação, diminuiu o risco do projeto falhar, auxiliou no planejamento e na determinação dos custos do projeto. Ele serviu de base para a comunicação entre todos os envolvidos. **Temos que planejar antes de colocarmos em prática!**

2.5 Etapas envolvidas no projeto de software

Bom, antes de darmos início ao aplicativo, precisamos planejar como ele será desenvolvido. Essa etapa consiste em projetar a melhor solução antes de colocar em prática. As etapas principais são: escolha da resolução da tela, cores, dispositivos alvo, disposição dos elementos na tela. Vamos dar uma olhada breve em cada um desses elementos.

2.6 Resolução de tela

Algo muito importante em um projeto de *software* para dispositivos móveis é o tamanho da resolução da tela em que o usuário irá utilizar o aplicativo. Temos que ter em mente que diversos dispositivos possuem uma resolução diferente e que nosso aplicativo **deve adaptar-se** ao tamanho da tela do dispositivo do usuário. Uma técnica muito difundida atualmente é o **layout responsivo** onde o conteúdo se adapta ao tamanho da tela do usuário.

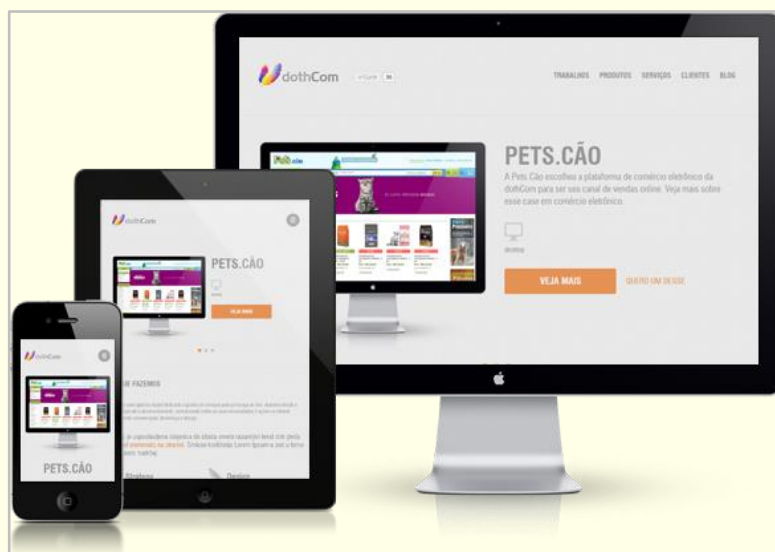


Figura 10 - Exemplo de layout responsivo.

Fonte: www.dothcom.net/wp-content/uploads/2012/11/responsive-web-design.png

Descrição: Exemplo de um layout responsivo. Independentemente do tamanho da tela, o conteúdo se adapta.

Dessa maneira, utilizar a técnica de layout responsivo diminui bastante o problema de adaptar o conteúdo ao tamanho da tela do dispositivo. Então, se formos desenvolver para apenas um tamanho específico de tela, temos que saber bem as dimensões da tela. Um monitor de uso comum, por exemplo, tem as dimensões 1280x720, isso medido em *pixels* (px) que é a unidade padrão para medição de tamanho de tela de um dispositivo.

2.7 Cores

A escolha das cores a serem usadas na criação dos elementos de um aplicativo está muito ligada a



quem vai usar o dispositivo. Comumente, associamos a cor com a imagem que queremos passar do nosso projeto. Em projetos para a *web* geralmente são utilizadas as **cores verde e azul**, para mostrar mensagens de sucesso ao usuário após uma determinada operação como, por exemplo, um cadastro do usuário.



Para conhecer uma lista completa de cores em hexadecimal e RGB acesso o site:
<http://erikasarti.net/html/tabela-cores/>

A cor pode “Influenciar nosso estado de espírito, criar diferentes atmosferas, alterar visualmente as proporções de um ambiente, aquecer ou esfriar um ambiente, valorizar e criar centros de interesse.” (Gurgel, 2007).

No desenvolvimento de aplicativos, como estamos trabalhando com tela de dispositivos móveis, usamos bastante a seleção de cores pelo código **hexadecimal** de representação. A cor preta é representada por #000000, também é usado a representação em RGB (do inglês Red, Green, Blue) {0,0,0}.

Todas as áreas de um aplicativo devem seguir a cor predominante que foi usada no projeto e cada área deve manter a identidade visual do sistema. Se a cor predominante é o verde, cada área diferente deve trazer essa cor para fazer referência ao projeto do sistema.

2.8 Disposição dos elementos na tela

A principal forma de interação do usuário com os aplicativos em dispositivos móveis é por meio do **toque**. A resposta ao toque funciona por meio de sensores na tela do dispositivo. Ao serem tocados, emitem um comando para o dispositivo que, dessa forma, processa-o e mostra um resultado. Essa tecnologia ficou muito popular por conta da disseminação dos *smartphones*.



O lugar em que cada elemento do aplicativo vai ficar na tela é muito importante para seu uso. Segundo uma pesquisa realizada por profissionais de design de experiência do usuário, 49% das pessoas seguram e usam o *smartphone* com apenas uma mão. Isso mostra a preocupação que devemos ter em colocar os principais elementos agrupados de forma que o usuário consiga utilizar apenas usando uma mão. Na imagem abaixo, podemos ver como os ícones são posicionados abaixo da tela para facilitar o acesso.



Link para a pesquisa sobre usuários que seguram o *smartphone* com apenas uma das mãos:
<http://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php>

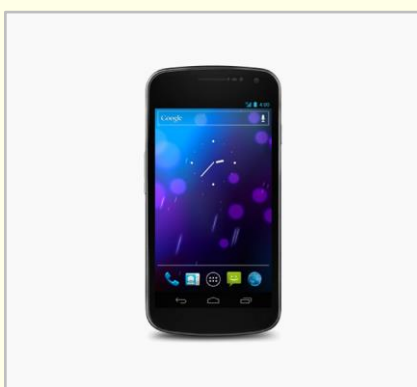


Figura 11 - Disposição dos elementos na tela.

Fonte: <http://tableless.com.br/design-para-telas-sensíveis-ao-toque/>

Descrição: a maioria dos elementos são colocados na parte inferior da tela para facilitar o toque usando apenas uma das mãos.

Quanto à direção dos elementos, direita ou esquerda, não é algo que possa ser previsto com exatidão. Existe uma grande variação na pegada, inclusive entre pessoas canhotas e destros. Na dúvida, coloque a navegação em uma posição que favoreça as duas mãos. Deixar o botão principal na posição central ou utilizar campos de entrada com 100% de largura para formulário de "*login*" são boas práticas.

Quando estamos tratando de sites para serem exibidos no dispositivo móvel temos que levar em consideração que o site é um aplicativo que executa dentro de outro aplicativo, o navegador (*browser*). O site *tableless* (www.tableless.com.br) tem um comentário muito pontual sobre o uso

de menus em sites para dispositivos móveis:

"Deixar a lista de links aberta no topo também pode ser uma estratégia perigosa. Dependendo da quantidade de opções da sua página, isso pode ofuscar completamente o conteúdo. A melhor prática de navegação, neste caso, é colapsar o menu em apenas uma âncora. Este tipo de abordagem pode ser facilmente criada utilizando JavaScript ou CSS3. Se você precisa de uma ajudinha, o plug-in ResponsiveNav dá conta do recado." (tableless).

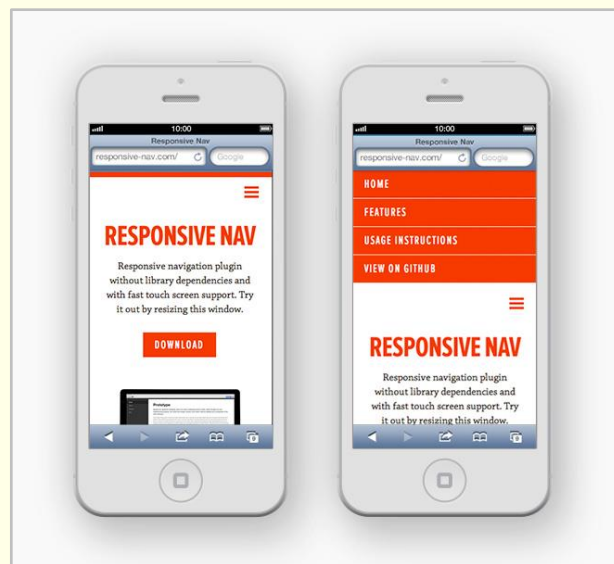


Figura 12 - Exemplo de menu que expande o conteúdo.

Fonte: www.tableless.com.br

Descrição: menu que expande o conteúdo quando clicado.



Agora que já vimos um pouco do passo a passo para o projeto de um aplicativo, vamos imaginar que iremos desenvolver um! Discuta com algum colega sobre um determinado aplicativo e comece a capturar os requisitos e a fazer o projeto em relação às cores, resolução de tela, público e disposição dos elementos. Algo que vale a pena ver é a questão de prototipagem, deem uma olhada no link:

<https://nuvemandroid.wordpress.com/2013/10/15/ferramentas-de-prototipagem-de-tela-para-dispositivos-moveis/>

Vamos praticar, pessoal! Na terceira semana iremos abordar alguns *softwares* para nos ajudar durante o desenvolvimento dos aplicativos para dispositivos móveis.



3.Competência 03 | Aplicação de Conceitos de Usabilidade na Construção de uma Interface Móvel

Bom, chegamos à última semana dos nossos estudos sobre usabilidade para dispositivos móveis. O que estão achando até o momento? Essa disciplina é muito importante para a área de desenvolvimento de *software*!

Nesta semana iremos analisar algumas dicas para construção de uma interface móvel e, de fato, construir uma usando um *software* online para geração de *layouts* (esboço de tela). Vamos em frente!

3.1 Os 7 mandamentos para construção de uma interface

Quando estamos falando em ajudar a vida do usuário no uso da interface do dispositivo, algumas dicas são bem úteis. Essa compilação de dicas foi feita a partir da observação contínua do uso de dispositivos móveis pelas pessoas.

1. **Tenha poucas funções no mesmo ambiente. Mantenha o foco!** Em cada tela do nosso aplicativo devemos colocar poucas funcionalidades, apenas as que realmente interessam para o objetivo da tarefa a fazer. Poluir a tela com muitas funcionalidades só vai confundir o usuário;
2. **"Tudo é essencial" não é uma opção.** Tudo que está contido naquela tela é essencial para desempenhar a tarefa. Não devemos colocar algo que não tenha utilidade no momento;
3. **Use botões grandes. Lembre-se do toque para um dedo e não para o cursor do *mouse*.** Como tudo nos dispositivos móveis é feito quase em sua totalidade com a funcionalidade *touchscreen*, vale salientar a importância de ter botões grandes;
4. **Pouca digitação! Formulários pequenos e atalhos.** É incômodo para os usuários ficarem digitando muita informação, dessa forma reduzir o tamanho do formulário é algo importante a se fazer. Atalhos também ajudam na hora de inserir um anexo ao formulário, por exemplo;
5. **A forma deve refletir a função, que é mais importante que a beleza aparente.** Um bom exemplo para a forma refletir a função é com o ícone de "salvar" de alguns aplicativos. Na sua



maioria o ícone usado é de um disquete que representa bem a "função" que é salvar o documento. Às vezes, queremos dar outra forma para aquela função e isso não reflete bem o resultado;

6. Existem padrões de interfaces para cada plataforma (iOS, Android, Windows Phone, etc.). Se notarmos, a maioria dos aplicativos de Android, por exemplo, tem o mesmo padrão de interface. Vale a pena conhecer os padrões existentes de cada sistema operacional;

7. **Nunca use o padrão de interface de uma plataforma em outra.** Simplesmente não use o padrão de interface do iOS no Android. Isso pode confundir, e muito, o usuário!

3.2 Construção de uma interface para dispositivos móveis

Bom pessoal, agora vamos realmente colocar a mão na massa e desenvolver uma interface para dispositivos móveis. Vamos pôr em prática tudo que vimos sobre os conceitos de usabilidade (foco: facilitar a vida do usuário) em relação aos aspectos de cores, tamanhos de telas, disposição de elementos e, claro, levando em consideração os requisitos fornecidos pelo cliente ou usuários.

Vamos usar a ferramenta *online* chamada *fluidui* que pode ser acessada no endereço www.fluidui.com. Essa ferramenta é paga, porém para nossa necessidade não precisaremos pagar para usá-la. Além disso, ela é em inglês, mas irei ajudá-los na tradução de alguns termos.

É muito importante se familiarizar com as ferramentas em inglês, é essencial no mundo da Tecnologia da Informação (TI).

Primeiramente, vocês precisam criar uma conta clicando no botão "CreateAccount" e serão levados para a tela com o formulário para preencher. A Figura 13 ilustra esse passo. No campo "I'm designing for" (estou desenvolvendo para...) escolham a opção Android - Phone (5:3) (um sistema operacional usado em celulares smathphone). O "5:3" indica o tamanho de tela padrão em que iremos trabalhar, porém esse tamanho pode se adaptar a telas maiores. Ao final clicar em "Start designing".



Figura 13 - Formulário para cadastro no Fluidui.com
Fonte: www.fluidui.com
Descrição: Formulário para criação de conta no Fluidui.com.

Após criar sua conta, dê dois cliques no texto mostrado na Figura 14. Isso irá dar início ao seu projeto.

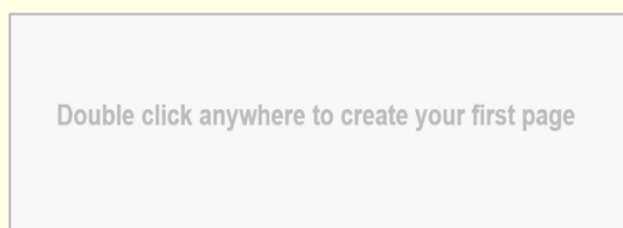


Figura 14 - Criar a primeira página e iniciar o projeto.
Fonte: www.fluidui.com
Descrição: criar a primeira página e começar o projeto.

Dado início ao projeto, você irá entrar em sua "dashboard" ou área de trabalho. Nela irá conter uma página em branco, conforme a Figura 15 e uma janela ao lado esquerdo com vários ícones organizados em seções, Figura 16. Podemos perceber que cada seção irá editar uma parte



específica da interface que estamos criando.

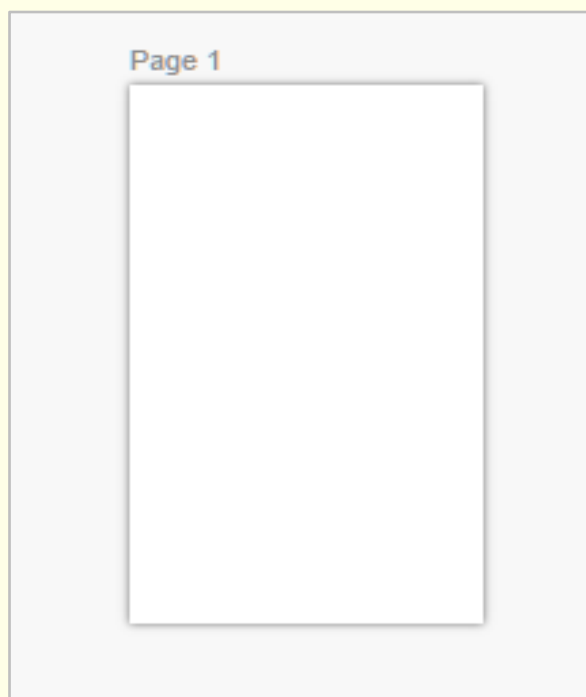


Figura 15 - Dashboard ou área de trabalho.

Fonte: www.fluidui.com

Descrição: dashboard ou área de trabalho para desenvolver a interface. Inicialmente a área de trabalho está em branco



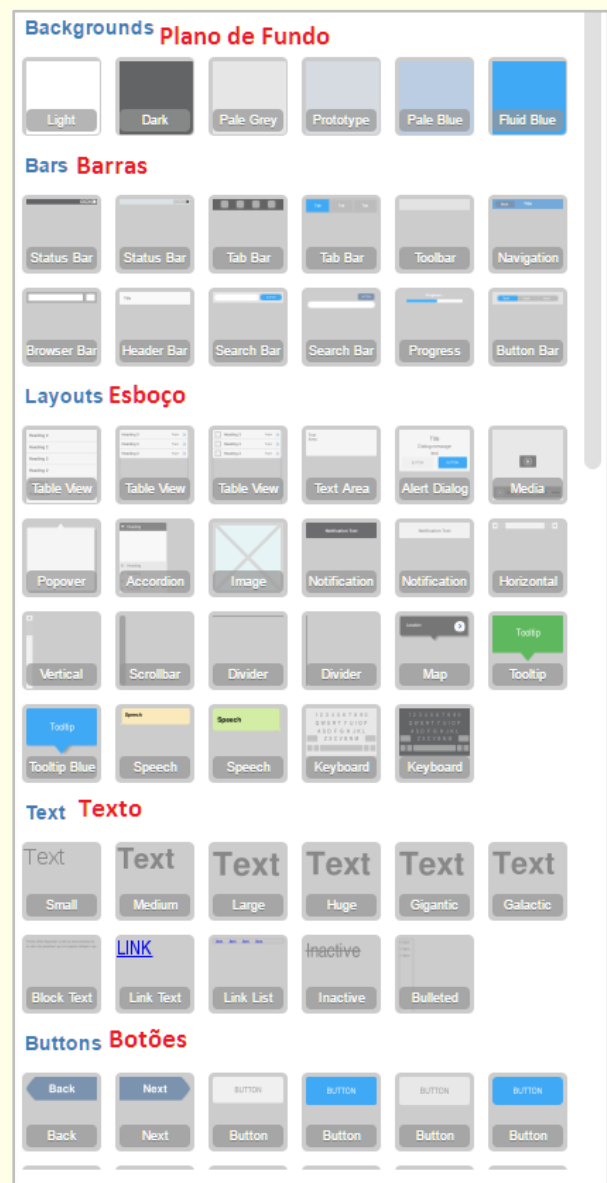


Figura 16 - Janela com as diversas funcionalidades para serem inseridas na interface.

Fonte: www.fluidui.com

Descrição: conjunto de funcionalidades para serem aplicadas na interface.



- **Background (plano de fundo):** irá editar a cor usada como plano de fundo da interface. Esse ponto é muito importante devido aos elementos que serão sobrepostos. Normalmente, é escolhida a cor branca, preta ou cinza claro;
- **Bars (barras):** é a barra superior que mostra as notificações no dispositivo móvel;
- **Layouts (esboços):** diversos tipos de elementos que podemos colocar na interface. Balões de mensagens, teclados, notificações, imagens, vídeos. Vale a pena ir clicando em cada um e conferindo;
- **Text (textos):** vários tipos e tamanhos de textos inclusive *links* para serem colocados na interface;
- **Buttons (botões):** são os botões que irão executar alguma função. Vale lembrar da dica de usar botões grandes e que expressem sua real funcionalidade;
- **Controls (controles):** controles que podemos habilitar ou não no aplicativo, diretamente ligados às funcionalidades;
- **Icons (ícones):** ícones que podem ser usados para executar alguma funcionalidade;
- **Invisible Link Areas (áreas de links invisíveis):** áreas invisíveis para auxiliar no desenvolvimento, ou seja, uma espécie de marcação para depois ser adicionada alguma funcionalidade.

Cada um desses itens que selecionamos podemos editar suas propriedades, por exemplo, se selecionarmos um texto aparecerá uma caixa de ferramentas acima mostrando algumas opções para formatar aquele texto. Para cada item que selecionarem, vejam as opções que são disponibilizadas! É uma ótima maneira de aprender o que podemos fazer. A Figura 17 ilustra essa caixa de ferramentas realçada em vermelho.



Figura 17 - Caixa de ferramenta para personalização.

Fonte: www.fluidui.com.

Descrição: caixa de ferramenta para edição.

Outra funcionalidade muito importante é a adição de outra página. Para adicionar, basta clicar no ícone "+" no canto superior direito. Com isso, podemos criar N páginas e, assim, definir o fluxo que o aplicativo terá ao clicar em determinado botão que leva para outra página. A Figura 18 mostra onde adicionar uma nova página (realçado em vermelho). A Figura 19 mostra um exemplo com três páginas.

É muito importante definirmos bem o fluxo de páginas para não deixar o usuário confuso na navegação. Um fluxo que contenha muitas páginas não é algo desejável, então é bom dosar bem a quantidade de fluxos, no máximo três fluxos, o que é uma quantidade aceitável.

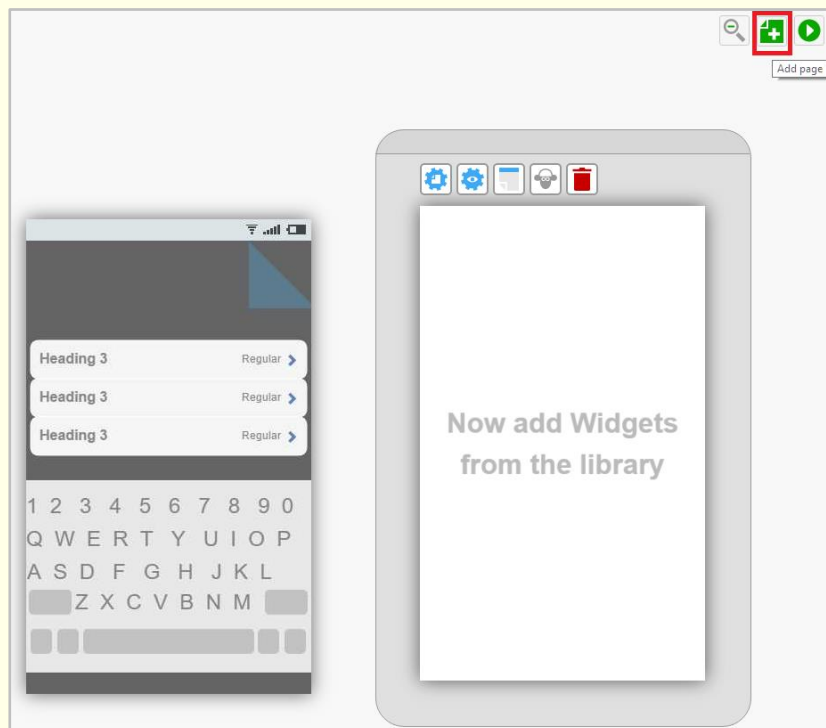


Figura 18 - Inserir uma nova página.

Fonte: www.fluidui.com

Descrição: Como inserir uma nova página ao projeto.

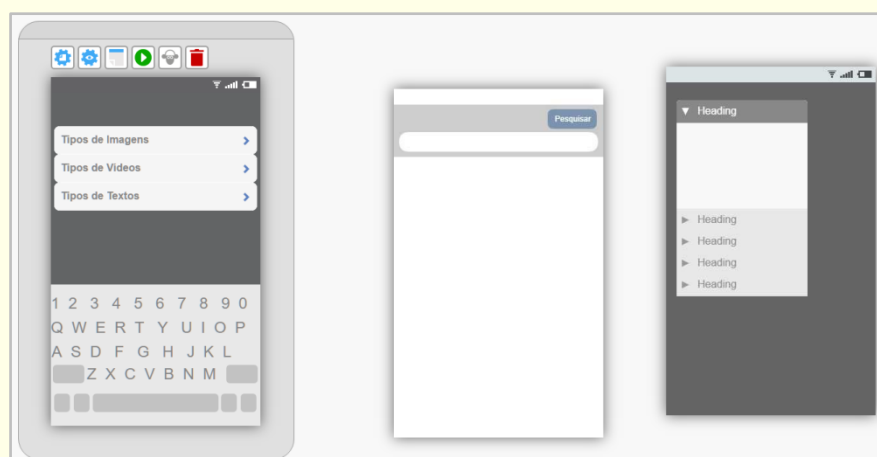


Figura 19 - Exemplo com mais de uma página.

Fonte: www.fluidui.com

Descrição: exemplo de projeto com mais de uma página.

Bom, utilizando mais o aplicativo você irá conhecer as diversas funcionalidades e incrementar o seu projeto de interface. Claro, sempre levando em consideração as boas práticas estudadas ao desenvolver o projeto. Espero que vocês tenham assimilado o conteúdo que é essencial para a área de desenvolvimento de software.



Conclusão

Com isto terminamos esta disciplina, mas entenda que você viu a ponta do iceberg, ou seja, apenas o começo. Existe muito mais conhecimento que deve ser procurado e aprendido para se aperfeiçoar. Mas apenas ler ou assistir uma aula não é suficiente. Para aprender você deve praticar, mais e mais.

Então, pesquise mais sobre os conceitos de usabilidade, metodologias de desenvolvimento de software e criação de interfaces para dispositivos móveis. Na internet você encontrará muito material interessante. E quanto mais cedo começar a praticar, mais rápido se tornará um profissional competente.

Um grande abraço e até a próxima.



Referências

BARBOSA, Simone Diniz Junqueira; SILVA, Bruno Santana da Campus. **Interação Humano-computador**. São Paulo: Campus, 2010.

www.tableless.com.br

www.fluidui.com

www.nngroup.com/articles/mobile-usability-update/



Minicurrículo do Professor



BRUNO FELIPE é formado em Sistemas pela Faculdade Marista, especialista em Administração de Banco de Dados pela Estácio de Sá, mestre em Ciência da Computação pela UFPE e doutorando em Ciência da Computação pela mesma universidade. Possui experiência na área de Análise e Desenvolvimento de Sistemas, Banco de Dados e Segurança Computacional.

