



# Interação Humano - Computador

Aline Chagas Rodrigues Marques

Pedro Henrique Barboza da Silva



**Curso Técnico em Informática**

Educação a Distância

2020



# Interação Humano – Computador

Aline Chagas Rodrigues Marques

Pedro Henrique Barboza da Silva

## Curso Técnico em Informática

Escola Técnica Estadual Professor Antônio Carlos Gomes da Costa

Educação a Distância

Recife

1.ed. | Fev. 2020



Licença Pública Creative Commons  
Atribuição-NãoComercial-Compartilhável 4.0 Internacional

#### **Professor(es) Autor(es)**

Aline Chagas Rodrigues Marques  
Pedro Henrique Barboza da Silva

#### **Revisão**

Pedro Henrique Barboza da Silva  
Aline Chagas Rodrigues Marques

#### **Coordenação de Curso**

José Américo Teixeira de Barros

#### **Coordenação Design Educacional**

Deisiane Gomes Bazante

#### **Design Educacional**

Ana Cristina do Amaral e Silva Jaeger  
Helisangela Maria Andrade Ferreira  
Izabela Pereira Cavalcanti  
Jailson Miranda  
Roberto de Freitas Morais Sobrinho

#### **Descrição de imagens**

Sunnye Rose Carlos Gomes

#### **Catálogo e Normalização**

Hugo Cavalcanti (Crb-4 2129)

#### **Diagramação**

Jailson Miranda

#### **Coordenação Executiva**

George Bento Catunda  
Renata Marques de Otero  
Manoel Vanderley dos Santos Neto

#### **Coordenação Geral**

Maria de Araújo Medeiros Souza  
Maria de Lourdes Cordeiro Marques

#### **Secretaria Executiva de Educação Integral e Profissional**

**Escola Técnica Estadual**  
**Professor Antônio Carlos Gomes da Costa**

#### **Gerência de Educação a distância**

fevereiro, 2020

#### **Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISDB**

S719i

Marques Aline Chagas Rodrigues.

Interação Humano Computador: Curso Técnico em Desenvolvimento de Sistemas:  
Educação a distância / Aline Chagas Rodrigues Marques, Pedro Henrique Barboza da Silva. –  
Recife: Escola Técnica Estadual Professor Antônio Carlos Gomes da Costa, 2020.  
60 p.: il.

Inclui referências bibliográficas.

Caderno eletrônico produzido em fevereiro de 2020 pela Escola Técnica Estadual Professor  
Antônio Carlos Gomes da Costa.

1. Ciência da Computação. 2. Interação homem-computador. 3. Design de Interação. I.  
Título.

CDU – 331:316.776



## Sumário

Introdução .....	5
1.Competência 01   Introdução à interação humano – computador.....	6
1.1 Interação humano – computador .....	6
1.1.1 Componentes da interação humano computador .....	9
1.2 Interface .....	11
1.2.1 Gerações das interfaces homem – computador.....	14
1.2.2 – Tipos de interfaces homem – computador .....	17
1.3 Interação .....	19
1.4 Usabilidade.....	22
2.Competência 02   Processo de desenvolvimento de aplicativos .....	24
2.1 Processo de software .....	24
2.2 Modelagem .....	26
2.2.1 Processo do projeto de interface.....	27
2.3 Protótipos.....	32
3.Competência 03   Aplicação de conceitos de usabilidade na construção de uma interface móvel	37
3.1. Recursos para construção de interfaces IOs e Android .....	37
3.2 Orientações sobre usabilidade.....	42
3.2.1 Acessibilidade digital.....	50
3.3 Modelagem de protótipo .....	53
Conclusão .....	58
Referências .....	59
Minicurrículo do Professor .....	60



## Introdução

Eis que você acorda pela manhã, através da sua assistente pessoal, que já lhe informa quais tarefas você precisa realizar naquele dia, a previsão do tempo, o trânsito da cidade e ela deseja a você um excelente dia. Você também deseja o mesmo para ela, e a mesma agradece.

Em seguida, você acessa a sua rede social através de um aplicativo e após visualizar várias postagens e curtir algumas, tira uma foto do dia ensolarado que está fazendo e realiza uma postagem em sua rede com a mensagem: “Bom dia, Comunidade!”.

Posteriormente, ao se dirigir para o trabalho, lembra que precisa retirar dinheiro em um caixa eletrônico. Após esperar as pessoas pagarem contas no caixa eletrônico e tirarem os extratos e ficarem olhando expressivamente para o papel impresso, você acessa sua conta, e através de alguns cliques, reconhecimento de digital, digitar a senha do cartão e enfim, consegue sacar o dinheiro. Ufa!

E é claro, que depois de tudo isso, você já está atrasado para o trabalho. Então, chega ao escritório e já está afobado para ligar o computador. Acessa a sua máquina, digitando seu login e senha do sistema. Acessa seu e-mail e envia a mensagem que o coordenador estava esperando, já faziam uns 5 minutos e responde imediatamente. Começa a trabalhar acessando vários sistemas e chega a hora mais esperada: Pausa para o café.

Na situação hipotética descrita acima, percebemos que o nosso personagem (você) tem várias situações que envolvem IHC (interação humano-computador) como: a assistente pessoal, o aplicativo da rede social, o caixa eletrônico, sistema, e-mail. Tudo isso, são exemplos de interfaces que permitem que seja realizada a comunicação entre o homem e máquina. Quando a comunicação flui entre essas duas partes, dizemos que está ocorrendo um processo de interação humano – computador.

Neste e-book nos aprofundaremos mais sobre esse assunto: na competência 01 veremos os principais conceitos da área de IHC; na competência 02 como funciona o processo para o desenvolvimento destas interfaces e na última competência como podemos aplicar a usabilidade para construir interfaces mobiles.

Agora que você já está contextualizado com o assunto deste caderno de estudos, vamos prosseguir com os conceitos pertinentes a esta área, pois você já está usando uma interface: o computador! Bons estudos.



## 1.Competência 01 | Introdução à interação humano – computador

Nesta competência serão descritos os fundamentos sobre a área de interação humano-computador, assim como, serão vistos os conceitos sobre interface, interação e usabilidade, pois é necessário que você tenha o entendimento disso para que o entendimento seja mais fluido para as próximas competências. Vamos começar?

### 1.1 Interação humano – computador

A área de interação humano computador (IHC) surgiu em meados dos anos 80 apresentando uma nova área de estudo dentro da informática, a qual procurava relacionar todos os parâmetros importantes para a construção de interfaces e a interação entre usuários e máquinas.

Conforme as diretrizes curriculares do MEC (Ministério da Educação) definem a área de IHC, para os cursos da área de computação, como uma disciplina que está relacionada ao projeto, implementação e avaliação de sistemas computacionais que promovem a interação e a facilidade de uso das interfaces.

Adicionalmente, a área de IHC está relacionada à especificação de processos, diálogos e ações, em que os usuários realizam interações com o computador de forma mais eficiente e intuitiva.

Como a área de IHC se preocupa tanto com o usuário quanto com a máquina, é imprescindível que quem estuda essa área entenda conceitos inerentes às máquinas e ao humano como:

- **Máquinas:** é necessário o conhecimento básico de sistemas operacionais, ambientes de desenvolvimento, linguagens de programação e técnicas de computação gráfica.
- **Humano:** abrange conhecimentos do tipo de informação para um projeto gráfico e industrial, ciências sociais, teoria da comunicação, linguística, desempenho humano e psicologia cognitiva.

Além disso, é importante ter o domínio da área de engenharia e projeto, assim como, as técnicas e métodos que englobam estas áreas e que são importantes para a área de IHC.

Como pode ser visto, a área de IHC abrange várias áreas como: **ciência da computação**, no que se refere ao projeto e implementação de interfaces; **psicologia**, referente à teoria dos processos cognitivos e análise de forma empírica quanto ao comportamento do usuário no uso de



computadores e **sociologia e antropologia**, a qual aborda o processo interativo entre tecnologia e trabalho.

Dentro deste contexto, a IHC é classificada como uma área interdisciplinar por abordar estas áreas e outros fatores como produtividade, aspectos organizacionais, sociais e comportamentais, segurança, eficiência, entre outros.

Para ilustrar melhor as disciplinas que corroboram para a área de IHC, veja a Figura 1:

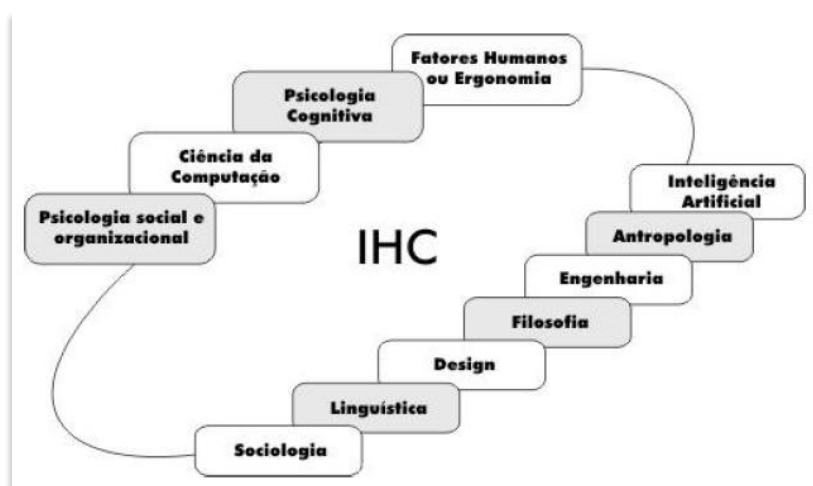


Figura 1 – Disciplinas que compõem IHC.

Fonte: (PREECE et al., 2005)

**Descrição da Imagem:** vários retângulos (nas cores branca e cinza), com as disciplinas que compõem a área de interação humano computador (IHC).

Vejamos como estas disciplinas atuam na área de IHC:

- **Psicologia social:** procura investigar a origem e causas do comportamento humano em um contexto social. Como exemplo podemos citar a forte influência do comportamento de um indivíduo nas tarefas de diferentes grupos.
- **Psicologia organizacional:** aborda as estruturas organizacionais sobre como o surgimento dos computadores influenciaram as práticas de trabalho e entendimento do funcionamento das organizações quanto à alguns aspectos como ambiente de trabalho, eficiência, fluxo de informação, tecnologia, contexto social, entre outros.
- **Ciência da computação:** relaciona todo o potencial que a tecnologia pode oferecer para a área de IHC, oferecendo suporte à utilização de softwares que auxiliem o processo de *design*, implantação e manutenção de sistemas com o uso de alguns



recursos como: ferramentas de prototipação, linguagens de programação, entre outros.

- **Psicologia cognitiva:** compreende o comportamento humano e seus processos da mente no que se refere ao processo de interação e uso de interfaces. Destaca-se também a memória, aprendizagem e como os usuários solucionam os problemas ao se deparar com as limitações da interface.
- **Fatores humanos ou ergonomia:** conceber o *design* de ferramentas para atender melhor às necessidades dos usuários, fazendo com que as tarefas sejam mais confortáveis e satisfatórias. Algumas contribuições podem ser destacadas como: *design* de *hardwares* ergonômicos.
- **Inteligência artificial:** contribui para o desenvolvimento da **abstração** do conhecimento humano na solução de problemas utilizando agentes inteligentes. Esses agentes auxiliam os usuários a lidar com o excesso de informação, na navegação e na busca e organização da informação.



A ciência da computação é baseada em abstração. Abstração é a ideia de isolar apenas as características importantes de um conceito ou fenômeno, em detrimento dos detalhes.

- **Antropologia, filosofia e sociologia:** compreender o que ocorre com os indivíduos ao interagirem entre si e com as máquinas, se preocupando com todas as fases do processo comunicativo, seja durante ou depois que ocorre a interação.
- **Engenharia:** construir modelos e testes empíricos sobre validação de interfaces, além de ser um ponto de apoio para a área de engenharia de software.
- **Design:** contribui na área de *design* gráfico para a construção de sites e outros artefatos relacionados ao processo de produção de *websites*.
- **Linguística:** explora a linguagem natural para o processo de construção de interfaces e possibilita inserir aspectos culturais em um produto.





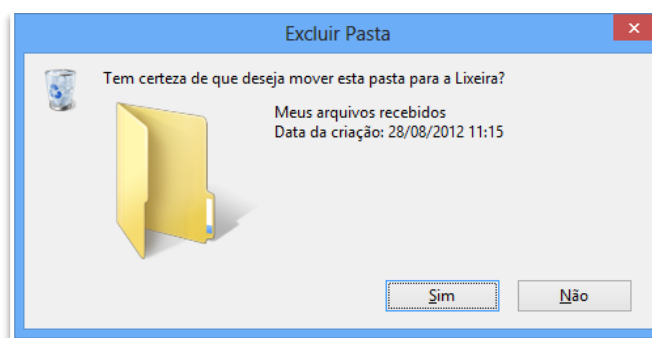
Como você pode perceber, muitas são as áreas envolvidas dentro de IHC, promovendo assim, uma maior sedimentação da área do ponto de vista multidisciplinar. Vale ressaltar que, a partir do conhecimento dessas áreas, podem ser construídos desde sistemas complexos com o uso de inteligência artificial até sistemas de baixa complexidade como *websites*.

Por fim, a área de IHC possui alguns desafios como: não ser excludente mediante o avanço da tecnologia. À medida que a tecnologia avança oferecendo novidades e infinitas possibilidades, há a preocupação de criar interfaces com maior objetividade e clareza na estrutura destas novas funcionalidades.

## 1.1.1 Componentes da interação humano computador

Para Nielsen, um grande autor da área, os componentes em IHC foram agrupados com a finalidade de estruturar e organizar as interfaces e os conhecimentos dos usuários. A classificação proposta pelo autor ocorreu da seguinte forma:

- **Caixas de diálogo:** são sequências comunicativas e interativas que auxiliam os usuários em seus processos interativos com a máquina. Os componentes básicos de uma caixa de diálogo são: **ações**, aquelas que fornecem uma entrada significativa acompanhada de uma resposta do sistema e **tarefas**, as quais são uma sequência de ações para um determinado objetivo, como por exemplo, a exclusão de um arquivo conforme podemos visualizar na Figura 2:



**Figura 2 – Exemplo de caixa de diálogo**

**Fonte:** autora

**Descrição da Imagem:** uma tela aberta (forma retangular na cor azul), com pasta (na cor amarela) para ser excluída.  
**Descrição do Texto:** texto descritivo para exclusão de arquivos que serão confirmados mediante o click do mouse sobre um dos botões: Sim, se deseja excluir o arquivo e Não, se não deseja realizar a exclusão.



- **Objetos de interação:** são objetos que constroem elementos apresentados em interfaces em que os usuários podem interagir. Para isso, utilizam **metáforas**, janelas, botões e apresentam elementos básicos como plano de fundo, primeiro plano e bordas, facilmente identificados.



Saiba mais sobre metáforas através do link:

<http://catarinasdesign.com.br/as-metaphoras-e-as-interfaces-de-usuario/>

- **Ícones:** são representações do mundo real para o virtual e devem ser considerados os seus significados, pois o processo de assimilação com a função em muitos deles, já estão bastante difundidos. Na Figura 3 temos exemplos de ícones de cartas para associar ao envio de e-mails:



Figura 3 – Exemplo de ícones

Fonte: (FREEPIK, 2019)

**Descrição da imagem:** apresenta o ícone da carta com fundo azul e a palavra e-mail.

- **Denominações e abreviaturas:** utilizam da linguagem natural para expor a informação em interfaces que sejam consideradas habituais para os usuários. Tanto as denominações quanto as abreviaturas devem fazer parte do contexto do usuário que está utilizando a interface: Vejamos um exemplo:

## O MELHOR EM LICITAÇÕES

### A SALA DE DISPUTAS MAIS TRANSPARENTE DO MERCADO

A LICITANET prima pela total transparência, pois basta qualquer cidadão acessar ao site para consultar detalhes sobre editais, atas e demais dados sobre os processos licitatórios, oportunizando assim, a qualquer cidadão ou órgão fiscalizador, o acompanhamento dos certames.



**Figura 4 – Tela do site licitanet**

**Fonte:** (LICITANET, 2019)

**Descrição da imagem:** apresenta textos que informam a tela de informações do site de licitações

- No site Licitanet encontramos denominações conhecidas dos órgãos públicos como: licitações, disputas. O usuário não tem dificuldade em entender a linguagem do *website*, pois fazem parte do seu vocabulário.
- **Primitivas:** são formas de expressar um objeto de interação com componentes relacionados ao cognitivo humano como o uso de fontes, cores, bordas, *layout*, *background* e sons. Vale ressaltar que o uso dessas primitivas deve ser feito com cautela, pois cada um deles representa uma finalidade específica na interface.

Portanto, os componentes em IHC tem um objetivo muito importante que é de organizar a informação na interface, utilizando o componente mais adequado para facilitar as tarefas dos usuários. Além de organização, fornece um melhor aproveitamento da interface utilizando o recurso adequado para repassar a informação de forma precisa.

## 1.2 Interface

Ao utilizar seu dispositivo móvel e através de alguns meios (aplicativos, sistema operacional, entre outros) você consegue estabelecer uma comunicação entre os comandos que você



envia e recebe os resultados do seu comando. Tudo isso é realizado através de *interfaces*! Vamos esclarecer melhor este conceito.

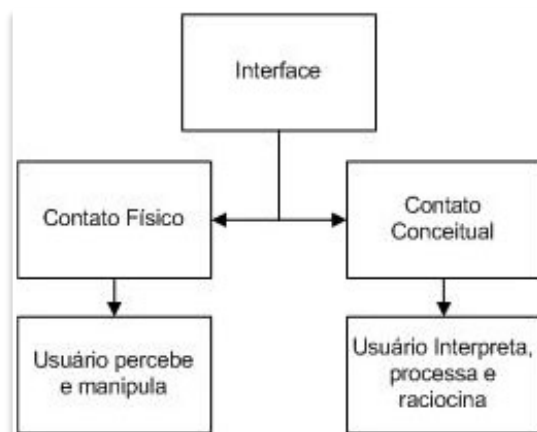
O termo *interface* pode ser definido como todo recurso que utilizamos para interagir com outra parte. Esse é o conceito mais simples encontrado sobre interface, ou seja, é o modo como ocorre a “comunicação” entre duas partes diferentes, como por exemplo, entre o homem e a máquina, que não podem se conectar diretamente.

A interface é o “meio do caminho” entre uma pessoa e um computador, por exemplo, e serve como um “serviço de tradução” entre esses dois lados, que não falam a mesma linguagem. Sem ela, teríamos que, por exemplo, ler os códigos de programação que fazem nossos equipamentos funcionarem.

Mas será que somente utilizamos interfaces em computação? Não. Veja um exemplo que pode ser utilizado em seu dia-a-dia: Você quer furar uma parede, mas precisa de uma ferramenta para que assim você possa realizar esta tarefa. Sendo assim, você utiliza uma furadeira. Pronto! A furadeira seria a nossa *interface*, pois é ela que vai te ajudar a cumprir determinado objetivo e você vai precisar interagir com a mesma. Ficou claro? Espero que sim!

Diante disso, percebemos que é necessário um contato do usuário com a interface, que, na área de computação, pode ser realizado de duas formas:

- **Física:** o usuário entra em contato com os elementos físicos do computador (*mouse* e teclado) acessando a interface para interagir e após isso, compreender as mensagens de retorno do sistema, que pode ser através de sons ou visualmente pelo monitor.
- **Conceitual:** após o usuário ter entrado em contato e durante a utilização do sistema, ele interpreta as respostas emitidas pelo sistema e avalia se os seus objetivos e/ou metas foram atingidas.
- Com base nisso, a Figura 5 ilustra essas formas de contato com a interface:



**Figura 5 – Formas de contato com a interface**

**Fonte:** (NIELSEN, 1990)

**Descrição da imagem:** apresenta 5 quadrados com setas que indicam as duas formas de contato com a interface e como isso ocorre, seja pela manipulação dos elementos ou pela interpretação e raciocínio realizado pelo usuário.

Perceba que a diferença entre os dois tipos de contato se estabelece quando no físico o usuário só interage com a interface e manipula a informação recebida, já no conceitual o usuário realiza atividades de processamento da informação e raciocina para saber se a sua interação alcançou os objetivos com o uso da interface.

Por conseguinte, as interfaces podem ser divididas em:

- **Interface homem-computador:** cujo objetivo é de servir de elo de comunicação entre duas partes para troca de informações: o computador e o homem, no sentido global da palavra.
- **Interface de usuário:** É a parte "visível" do projeto ou sistema, no qual o usuário interage. Ainda podemos dizer que é o meio pelo qual uma pessoa interage com um dispositivo, software ou aplicativo. Essa interação pode ser feita através de elementos que forneçam ações entre o dispositivo e o usuário, como por exemplo, botões, *links*, menus, entre outros.

Outro ponto que não podemos deixar de conceituar é o termo interface amigável, que é tão utilizado em IHC. É um recurso onde o usuário interage e parece ser invisível e 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 (em muitos casos, quando não se tem essa característica, 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 que deve ser levado em consideração em detrimento de custos com treinamento.

Um exemplo que podemos citar que tem uma interface amigável é o *Login* do *Facebook* através do site:



**Figura 6 – Login do Facebook**

**Fonte:** (FREEPIK, 2019)

**Descrição da imagem:** a imagem apresenta uma pessoa manipulando um telefone para acessar a interface de uma rede social chamada Facebook.

Para realizar o *Login* basta inserir seu *e-mail* ou telefone e sua senha e clicar no botão entrar. O sistema possibilita a interface amigável quando o usuário ao visualizar a tela já sabe como utilizar e quais informações são solicitadas a ele, promovendo assim, o fácil uso da interface. Aposto que você não precisou de treinamento para utilizar o *Facebook*, não é?

## 1.2.1 Gerações das interfaces homem – computador

Conforme os computadores foram evoluindo, as interfaces acompanharam esse processo evolutivo e puderam fornecer ao usuário uma melhor facilidade de uso e melhoria no acesso das funcionalidades com a atualização de vários componentes utilizados na interface. Temos 4 (quatro) gerações desse tipo de interface que são:

### Primeira geração



Os computadores resolviam problemas específicos e eram operados por especialistas que tinham que manipular comandos complexos para acessar recursos de *hardware*. Além disso, a relação do homem computador era bem restrita e o computador era manipulado por chaves e botões.

Nesse contexto, o especialista criava comandos de forma textual para acessar todos os recursos da máquina utilizando uma interface muito simples. Alguns exemplos de interface dessa geração são: *unix*, *ms-dos* e sistema operacional dos *mainframes*.

**Nos links abaixo você sabe mais sobre cada um desses sistemas operacionais:**

Unix: <https://www.tecmundo.com.br/macros/10556-unix-o-pai-de-todos-os-sistemas-operacionais.htm>

MS-DOS: <https://brasilescola.uol.com.br/informatica/ms-dos.htm>

Sistema dos mainframes: <http://coopersystem.com.br/entenda-o-que-e-mainframe>

Na Figura 7 temos um especialista manipulando um computador da empresa IBM (*International Business Machines Corporation*) através de um cartão perfurado para interagir com o sistema operacional da IBM:



**Figura 7 – Computador IBM**

**Fonte:** (BARBOSA, 2010)

**Descrição da imagem:** a imagem apresenta duas pessoas em uma sala, em que uma pessoa está sentada e outra em pé, manipulando um computador da empresa IBM.

## Segunda geração

Nessa geração as interfaces começam a utilizar um padrão de opções em um menu numerado. Através do número digitado na interface, a decisão selecionada é repassada para o computador executar. Veja uma ilustração na Figura 8:

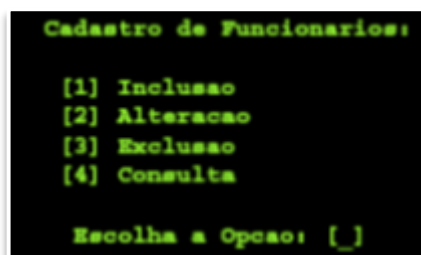


Figura 8 – Interface para cadastro de usuários

Fonte: (BARBOSA, 2010)

**Descrição da imagem:** a imagem apresenta um fundo preto com letras verdes pedindo para o usuário escolher uma opção que pode ser inclusão, alteração, exclusão e consulta.

Na tela para realizar opções para manipular um funcionário é necessário digitar um comando para executar a ação. Por exemplo, para incluir um funcionário, o especialista insere no campo “Escolha a opção: ” o número 1 entre colchetes.

## Terceira geração

Nesta geração temos uma grande evolução dos sistemas operacionais e suas interfaces, que são orientadas a janelas e possuem o recurso de clicar utilizando o *mouse*. Aqui surgem as interfaces conhecidas como WIMP, um acrônimo em inglês para Janelas (**W**indows), Ícones (**I**cons), **M**enus e Apontadores (**P**ointers), as quais permitem a interação através de componentes de interação virtuais denominados *widgets*. Este componente é implementado com o auxílio das tecnologias de interfaces gráficas, que proporcionam o desenho de janelas e o controle de entrada através do teclado e do mouse em cada uma destas janelas (vide Figura 9):

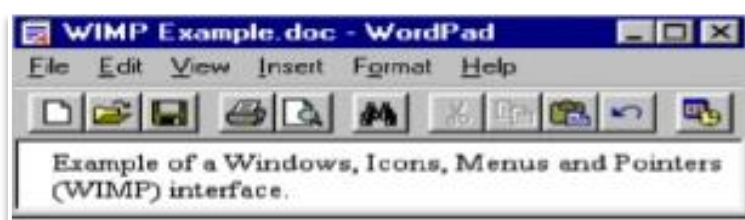


Figura 9 – Exemplo de Wimp.

Fonte: (BARBOSA, 2010)

**Descrição da imagem:** a imagem apresenta uma representação da interface WIMP com texto em inglês disponibilizando um menu com opções File, Edit, View, Insert, Format e Help. Além de ícones que representam ações do sistema como abrir arquivo no ícone de pasta. E possui o texto de exemplo no fundo branco.





Os softwares de interfaces que implementam estes estilos permitem a construção de ícones que facilitam a interação através do mouse, comportando-se como dispositivos virtuais de interação.

## Quarta geração

Nesta geração somados os atributos da terceira geração tem-se os recursos de: *hipertexto*, em que apenas uma palavra tem atrelada a ela um conjunto de informações e multitarefa, em que os computadores passaram a executar várias tarefas de forma simultânea.

O [sistema](#) de hipertexto mais conhecido atualmente é a [World Wide Web](#), no entanto a Internet não é o único suporte onde este modelo de organização da informação e produção textual se manifesta.

**Figura 10 – Exemplo de hipertexto**

**Fonte:** (BARBOSA, 2010)

**Descrição da imagem:** a imagem apresenta um fundo branco com um texto descritivo nas cores em azul e preto.

Neste exemplo temos alguns *hipertextos* como a palavra *sistema*, em que ao clicar nessa palavra, o usuário encontra outros conceitos relacionados a sistema. E a palavra *World Wide Web* em que faz um ligação com os conceitos relacionados a WWW.

## 1.2.2 – Tipos de interfaces homem – computador

Vimos que as interfaces foram evoluindo conforme as gerações dos computadores e muitas das funcionalidades acabaram agregando muitas funções em poucos botões. Vamos, agora, aos tipos de interfaces homem-computador que existem:

- **CUI (*character - based user interface*):** como o próprio nome diz, são as interfaces que possuem apenas caracteres alfanuméricos e textos e são bastante utilizadas na primeira geração de computadores. Veja a figura abaixo que expõe esse tipo de interface:



```
caelj@mac.local ~ $ mkdir repo
caelj@mac.local ~ $ cd repo
caelj@mac.local repo $ git init .
Initialized empty Git repository in /Users/caelj/repo/.git/
caelj@mac.local repo $ touch foo
caelj@mac.local repo $ git stage .
caelj@mac.local repo $ git commit -m "Epoch."
[master (root-commit) 9ed3c75] Epoch.
 0 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 foo
caelj@mac.local repo (master) $ git checkout -b some_feature_branch
Switched to a new branch 'some_feature_branch'
caelj@mac.local repo (some_feature_branch) $ git branch
  master
* some_feature_branch
caelj@mac.local repo (some_feature_branch) $ git checkout master
Switched to branch 'master'
caelj@mac.local repo (master) $
```

Figura 11 – Comandos no software GIT.

Fonte: (GIT, 2019)

Descrição da imagem: a imagem apresenta um fundo preto com trechos de códigos utilizados na linguagem de comando.

- **GUI (graphical user interface):** são consideradas as interfaces que fazem uso de metáforas utilizando ícones abstraídos do mundo real para o virtual como: pastas, botões, janelas e fazem parte da terceira geração de computadores. A figura abaixo exemplifica esse tipo de interface:



Figura 12 – Sistema operacional Windows 95

Fonte: (BARBOSA, 2010)

Descrição da imagem: a imagem apresenta uma tela com fundo verde e três janelas sobrepostas sobre o menu do Windows chamado Programas.



- **PUI (pen-based user interface):** são os tipos de interfaces que apresentam um monitor de vídeo dispostos em bloco de anotações ou planilha que recebe dados através de uma caneta. A metáfora aqui utilizada é a do ato de escrever. A Figura abaixo reflete esse tipo de interface:



**Figura 13 – Interface PUI**

**Fonte:** (BARBOSA, 2010)

**Descrição da imagem:** na imagem tem-se um monitor com várias quadrados e uma pessoa utilizando uma caneta para manipular estes elementos da tela do computador.

Por fim, entendemos os variados tipos de interfaces que acompanharam a evolução dos computadores e puderam facilitar a vida do usuário cada vez mais, com a disponibilização de novos componentes e recursos.

## 1.3 Interação

Define-se interação como a forma de comunicação entre os usuários e a máquina através do uso da interface. Nesse sentido, os usuários enviam informações para o sistema, o qual envia resposta adequada para as ações dos usuários.

Nessa interação são requisitados dispositivos **de entrada** (como exemplo, *mouse*) que fornece dados de entrada para a máquina e **de saída**, como mensagens de *feedback* da máquina para o usuário.

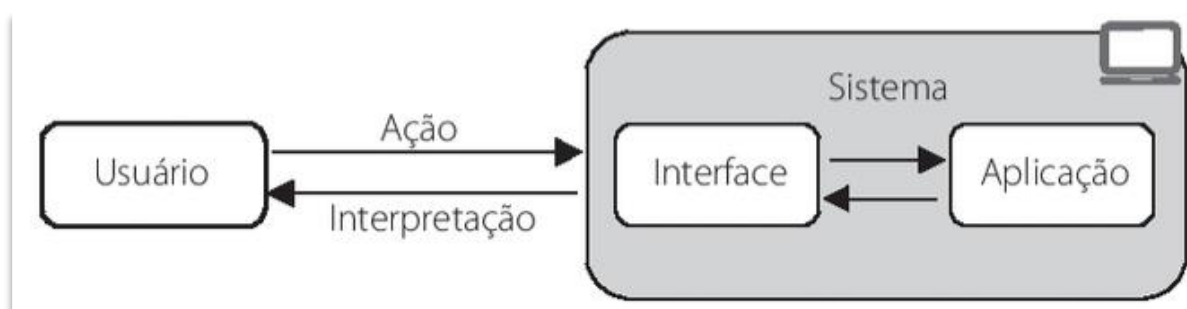
A seleção da tecnologia utilizada para os dispositivos de entrada e saída é importante para o processo da construção de uma interface. Como se tem uma enorme variedade neste tipo de



dispositivos, um grande desafio para o desenvolvedor da interface é achar a combinação perfeita entre tecnologia, aplicação e dispositivos de entrada e saída (E/S), já que isso afeta diretamente a usabilidade da interface.

Vale ressaltar que a interface, em computação, configura como um sistema de comunicação entre o usuário e a máquina, o que faz dela tanto um meio para essa interação como uma ferramenta que executa diversas funções.

Essa interação é o conjunto formado por sistema e usuário, o qual executa ações sobre o sistema, que os interpreta e os comunica em linguagem de máquina às aplicações disponíveis (PREECE, ROGERS e SHARP, 2002). A figura abaixo expõe essa interação:



**Figura 14 – Processo de interação humano – computador.**

**Fonte:** (PREECE, ROGERS e SHARP, 2002)

**Descrição da imagem:** a imagem apresenta um retângulo com bordas arredondadas com fundo cinza e um ícone de monitor. Tem-se também mais três retângulos com bordas arredondadas contendo cada um o seguinte texto: usuário, interface e aplicação, além de setas que informam que a ação é do sentido da esquerda para a direita e a interpretação é da direita para a esquerda.

Na figura acima percebemos que 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.

Por conseguinte, todas as vezes que precisamos nos comunicar com o sistema para informar o que precisamos fazer, é iniciado este processo de interação, a qual é a capacidade do sistema em fornecer respostas para os usuários e também, está atrelado ao processo de comunicação do usuário com o sistema.

Um exemplo deste processo de interação é o uso do *touch screen* (toque na tela), em que a forma de interação é baseada no toque da tela que funciona como entrada de dados e o sistema recebe isso, e devolve uma resposta para o usuário.



Dentro desse contexto, é importante sabermos diferenciar interface de interação para que os conceitos não sejam confundidos. Vamos visualizar isso na Tabela abaixo:

Interface	Interação
É o meio que interliga dois ou mais sistemas.	Processo de comunicação usuário-sistema.
Aquilo que facilita a comunicação do usuário com a máquina através de software e hardware.	Processo que engloba as ações dos usuários no sistema e interpretações dos resultados.

Tabela 1 – Diferenças entre interface e interação.

Fonte: Autora

Agora que você entendeu a diferença entre interface e interação, vejamos como a interação é classificada em 3 (três) sentidos dependendo da forma em que segue o fluxo de comunicação, podendo ser:

- **Do usuário para o sistema:** é a forma mais habitual em que o usuário realiza uma ação e aguarda uma resposta.
- **Do sistema para o usuário:** está relacionada a entrada de dados do usuário através de uma ação e o sistema emite uma saída de dados, podendo ser um *feedback*.
- **Do sistema para o sistema:** quando há a comunicação de um sistema com o outro para poder realizar uma ação. Por exemplo, quando o usuário posta uma foto no *Instagram* e solicita que seja publicada no *Facebook*, ocorrendo, assim, a interação entre esses dois sistemas (*Instagram* e *Facebook*).
- Podemos ainda dizer que a interação com o sistema pode ocorrer através de vários componentes como:
  - **Menus:** fornece uma lista de opções expondo funcionalidades do sistema de fácil acesso para os usuários.
  - **Atalhos:** são utilizadas por usuários mais experientes e são combinações de teclas que fornecem uma ação trazendo mais agilidade.
  - **Formulários:** baseiam-se na digitação de informações e preenchimento de campos, utilizando a metáfora (comparação de uma ação virtual com uma do mundo real) de formulários em papel.
  - **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.

Por fim, entendemos que a interação reflete uma comunicação realizada entre o usuário e o computador utilizando a interface, a qual é somente o meio que interliga estes dois atores (usuário e máquina). A comunicação pode partir de vários sentidos e utilizam-se de vários componentes para realizar essa interação com o sistema.

## 1.4 Usabilidade

Outro conceito importante é Usabilidade, a qual prediz que se deve verificar se a funcionalidade do sistema faz o que deve ser feito, por exemplo, se um jogo efetivamente diverte ou um *software* educacional auxilia o aprendizado dos alunos. Resumindo, é a questão relacionada a quão bem os usuários podem usar a funcionalidade definida.

Mas como podemos medir a usabilidade da *interface*? A norma ISO 9241 descreve a usabilidade de forma objetiva e clara. De acordo com a norma, para verificar o nível de usabilidade de uma *interface*, deve-se avaliar os seguintes elementos: eficácia, eficiência e satisfação.

O primeiro deles — a **eficácia** — implica que a tarefa que se pretende executar deve ser realizada de forma correta e completa. Em outras palavras, se o aplicativo ou site desempenha a função a que se propõe quando o usuário a solicita.

A **eficiência**, por sua vez, diz respeito aos recursos utilizados para que a tarefa seja desempenhada, ou seja, desde recursos financeiros e tempo utilizados para desenvolver a funcionalidade, até a quantidade de memória que ela gasta para ser executada. Assim, para um *design* eficiente, determinadas funcionalidades devem reduzir ao mínimo os recursos empregados.

Por último, a **satisfação** diz respeito ao conforto e facilidade que o usuário tem ao usar a ferramenta. Algumas perguntas podem ajudar você a saber se a usabilidade da interface atinge os requisitos de satisfação como: A navegação é intuitiva? As funcionalidades estão bem exibidas? Cada comando é bem explicado? A satisfação é uma diretriz para você pensar essas questões.

Além disso, a norma ISO 9241 define um conjunto de características que devem ser empregadas em um sistema para que ele obtenha um nível alto de usabilidade como:



- **Inteligibilidade:** compreende-se como a facilidade que o usuário entende as funcionalidades dispostas no sistema e faz uma avaliação para saber se as mesmas atendem suas necessidades específicas.
- **Apreensibilidade:** está relacionada com a facilidade de aprendizado do sistema para os seus possíveis usuários.
- **Operacionalidade:** facilidade de operação do sistema pelos usuários e como o sistema tolera erros de execução no sistema por parte de seus usuários.
- **Atratividade:** o quão atrativo o sistema está sendo representado para o usuário, incluindo os recursos visuais e gráficos das interfaces.

Todas essas características, segundo a norma, representam como a usabilidade pode ser melhorada a nível de sistemas e interfaces. Além disso, a norma também esclarece um conjunto de sete (7) princípios ergonômicos para projetos e avaliação de IHC:

- **Adaptação à tarefa:** é a capacidade que o sistema tem de se adaptar às preferências dos usuários.
- **Autodescrição:** respostas imediatas do sistema para os seus usuários, é mais comumente nomeada como *feedback* que o usuário recebe mediante uma ação.
- **Controle ao usuário:** refere-se ao controle que o usuário tem sobre o sistema e o processamento realizado.
- **Conformidade às expectativas dos usuários:** está relacionada às funcionalidades que devem ser de fácil acesso para os usuários e que atenda aos objetivos dos usuários.
- **Tolerância aos erros:** como o sistema detecta e previne erros na entrada de dados.
- **Facilidade de individualização:** é a capacidade da aplicação de adaptar ao contexto e às preferências do usuário.
- **Facilidade de aprendizagem:** refere-se a como os usuários aprendem de forma fácil a utilizar a interface.

Portanto, é de extrema importância que estes princípios sejam utilizados para promover uma boa usabilidade para seus usuários, no sentido, de que se sintam satisfeitos com relação ao uso da interface.



## 2.Competência 02 | Processo de desenvolvimento de aplicativos

Agora que você já entendeu todos os conceitos da área de IHC, vamos nos aprofundar no processo de desenvolvimento de aplicativos que envolvem a modelagem da interface. Perceba que dentro das várias etapas de desenvolvimento de software, estaremos preocupados em aprender sobre o projeto da interface. Vamos começar!

### 2.1 Processo de software

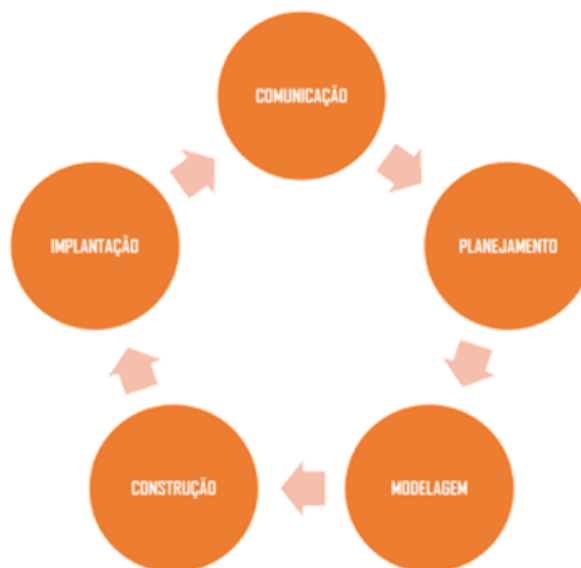
Processo é um conjunto de atividades, ações e tarefas realizadas na criação de artefatos. No contexto da engenharia de software, um processo não é uma prescrição rígida de como desenvolver um software.

Ao contrário, é uma abordagem adaptável que possibilita às pessoas (a equipe de software) realizar o trabalho de selecionar e escolher o conjunto apropriado de ações e tarefas. A intenção é a de sempre entregar software dentro do prazo e com qualidade suficiente para satisfazer aqueles que patrocinaram sua criação e aqueles que vão utilizá-lo.

Além disso, em um processo temos uma metodologia que estabelece o alicerce para um processo de engenharia de software completo por meio da identificação de um pequeno número de atividades metodológicas aplicáveis a todos os projetos de software, independentemente de tamanho ou complexidade.

Vale ressaltar que uma metodologia de processo genérica para engenharia de software compreende cinco atividades (vide Figura abaixo):





**Figura 15 – Metodologia de processo genérica.**

Fonte: a autora

**Descrição da imagem:** cinco círculos ( na cor laranja) com setas entre eles no sentido da direita para esquerda, dando ideia de movimento; cada um contendo os seguintes textos: comunicação, planejamento, modelagem, construção e implantação, com setas indicativas formando um círculo sequencial.

Agora, vamos descrever resumidamente estas atividades:

- **Comunicação:** Antes que qualquer trabalho técnico possa começar, é de importância fundamental se comunicar e colaborar com o cliente (e outros envolvidos). A intenção é entender os objetivos dos envolvidos para o projeto e reunir requisitos que ajudem a definir os recursos e as funções do software.
- **Planejamento:** Um projeto de software é uma jornada complicada, e a atividade de planejamento cria um “mapa” que ajuda a guiar a equipe na sua jornada. O mapa – denominado plano de projeto de software – define o trabalho de engenharia de software, descrevendo as tarefas técnicas a serem conduzidos, os riscos prováveis, os recursos que serão necessários, os produtos resultantes a serem produzidos e um cronograma de trabalho.
- **Modelagem:** Independentemente de ser um paisagista, um construtor de pontes, um engenheiro aeronáutico, um carpinteiro ou um arquiteto, trabalha-se com modelos todos os dias. Esta etapa está envolvida com o uso de modelos e tem como objetivo criar um “esboço” para que se possa ter uma ideia do todo – qual será o aspecto do projeto em termos de arquitetura, como as partes constituintes se encaixarão e



várias outras características. Se necessário, refina-se o esboço com mais detalhes, numa tentativa de compreender melhor o problema e como resolvê-lo. Um engenheiro de software faz a mesma coisa, criando modelos para entender melhor as necessidades do software e o projeto que vai atender à essas necessidades.

- **Construção:** Tudo que for projetado nas etapas anteriores deve ser construído nesta fase. Essa atividade combina geração de código (manual ou automatizada) e testes necessários para revelar erros na codificação.
- **Implantação:** O software (como uma entidade completa ou como um incremento parcialmente concluído) é entregue ao cliente, que avalia o produto entregue e fornece *feedback*, baseado na avaliação.

Essas cinco atividades metodológicas genéricas podem ser utilizadas para o desenvolvimento de programas pequenos e simples, para a criação de aplicações para a Internet e para a engenharia de grandes e complexos sistemas baseados em computador. Os detalhes do processo de software serão bem diferentes em cada caso, mas as atividades metodológicas permanecerão as mesmas. Para nosso estudo vamos focar na atividade de modelagem, certo?

## 2.2 Modelagem

A modelagem de um aplicativo/sistema é o processo que permite realizar a conversão da especificação do sistema em um produto executável, utilizando sempre, processos de projeto e programação de software.

Vale ressaltar que um projeto de software é uma descrição da estrutura de software a ser implementado, dos modelos, estruturas de dados, das interfaces, dos componentes e dos algoritmos utilizados pelo sistema. Os projetistas não chegam a um produto final, mas conseguem refinar mais como será a projeção do software a ser produzido.

Existem 4 (quatro) atividades básicas envolvidas no processo de projeto, em computação como:

- **Projeto de arquitetura:** visa identificar a estrutura geral do sistema, os componentes e os relacionamentos entre estas partes no sistema.



- **Projeto de interface:** objetiva definir a interface utilizada por esses componentes. Vamos detalhar melhor esta atividade na seção a seguir. Aguarde!
- **Projeto de componente:** tem como finalidade especificar cada componente do sistema e seu funcionamento. Um componente pode ser uma funcionalidade do software.
- **Projeto de banco de dados:** com a finalidade de projetar a estrutura de dados do sistema e como deve ser a representação destes em um banco de dados.

Como o foco do nosso caderno de estudos é o desenvolvimento e projeto de interfaces, vamos nos ater a atividade de projeto de interface, com a utilização de um processo. Vejamos a seguir.

## 2.2.1 Processo do projeto de interface

A maioria dos sistemas utilizados atualmente requerem basicamente uma fluida interação humana e uma boa usabilidade para assim, o projeto ser considerado bem-sucedido. Independente do contexto de uso e da interface, a usabilidade é um fator chave para promover a interação que o usuário precisa para realizar suas tarefas de forma satisfatória. Entretanto, um mal projeto de interface, causa frustração e baixo rendimento no trabalho do usuário.

Mas o que é um projeto de interface? É a elaboração de etapas que criam uma forma de comunicação e um mecanismo de interação fluido entre o computador e o usuário. Baseia-se em um conjunto de princípios para identificar objetos e ações da interface e assim, criar um *layout* de tela que forma a base de um protótipo de interface do usuário.

Tais princípios foram divididos primeiramente no que foram nomeadas **Regras de Ouro** que formam a base para toda a construção de um projeto de interface. Vejamos quais são:

### 1. Deixar o usuário no comando

Refere-se a um conjunto de diretrizes que proporcionam o usuário manter o controle sobre as suas ações durante a interação com a interface. Para isso, são fornecidas as seguintes diretrizes:



- Defina modos de interações (estado atual da interface) para que o usuário consiga se locomover por esses modos sem nenhum esforço, fazendo com que ele não realize ações desnecessárias;
- Promova uma interação flexível para englobar diversas preferências de interação seja por teclado, *mouse*, *touch screen* entre outros;
- Possibilite o ato de interromper e desfazer qualquer ação do usuário durante a interação com a interface;
- Permita que o usuário personalize seus níveis de interação à medida que ele constata que realiza uma sequência de interações de forma repetida;
- Oculte todos os detalhes técnicos da aplicação para o usuário comum.

## **2. Reduzir a carga de memória do usuário**

Este princípio prediz que quanto mais um usuário precisa se recordar das suas interações com a interface, maior é a probabilidade de erros. Para isso, a interface deve sugerir alguns caminhos possíveis para que o usuário consiga realizar sua tarefa. Para tanto algumas dicas favorecem esse princípio como:

- Reduzir a demanda da memória recente através do fornecimento de atalhos recentes feitos pelo usuário, assim ele não precisa se recordar do que foi feito para se chegar até aquele modo de interação na interface;
- Inserir metáforas do mundo real no *layout* visual da interface para que o usuário apoie suas ações em indicações visuais compreensíveis para ele;
- Revelar as informações progressivamente conforme o usuário demonstra interesse por maiores detalhamentos sobre as funcionalidades disponibilizadas em um nível de maior abstração.

## **3. Tornar a interface consistente**

Aqui as informações e o *layout* visual devem se manter consistentes, como forma de não causar certa confusão no processo interativo do usuário com a aplicação. Para obter isso, algumas orientações são recomendadas:



- Permitir que o usuário saiba em qual contexto do sistema ele está e para onde poderá seguir;
- Manter a mesma regra de projeto ao longo de uma família de aplicações parecidas;
- Manter a sequência interativa padrão, após ter se tornado um hábito para o usuário. Mudar isso pode causar algum desconforto no manuseio das funcionalidades da interface.

Por fim, estas regras de ouro fornecem algumas orientações importantes para o processo do projeto de interface. O delineamento de um processo geral de análise e projeto de interfaces perpassa por várias etapas a seguir:

- (1) construção de diversos modelos das funções do sistema;
- (2) descrição das tarefas orientadas à interação usuário-máquina através do processo estabelecido;
- (3) modelagem da interface;
- (4) a construção de protótipos para serem avaliados pelos usuários finais em termo de qualidade.

Vejamos, agora, os detalhes destas etapas: na etapa 1 são descritos 4 (quatro) tipos de modelos a seguir:

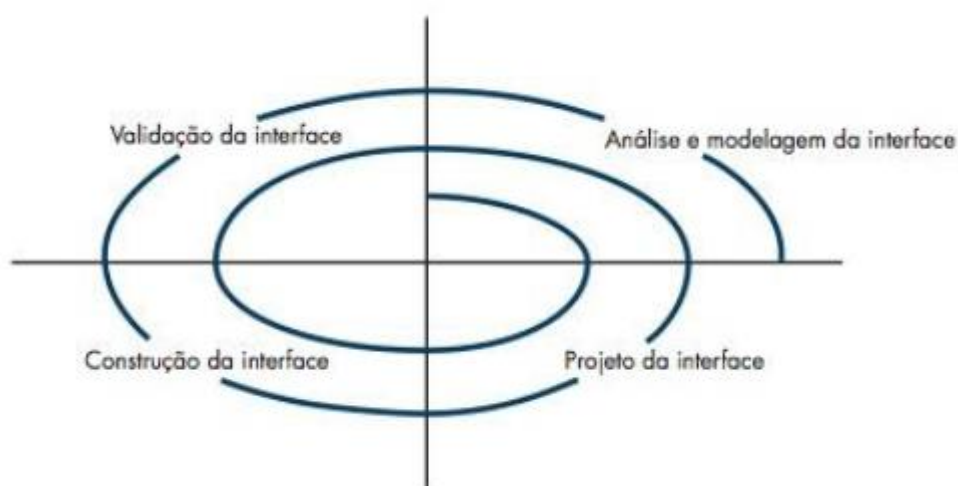
- **Modelo de usuário:** definir os perfis dos usuários finais do sistema para a construção de uma interface efetiva. Os usuários podem ser classificados em: *novato*, aqueles que não possuem nenhum conhecimento sobre a aplicação; *intermitente*, aquele que tem conhecimento razoável sobre a aplicação e *frequente* com conhecimento profundo sobre o sistema.
- **Modelo de projeto:** baseia-se nas informações contidas no modelo de usuário para a construção da interface.
- **Modelo mental:** aquele que se baseia na percepção que o usuário tem sobre o sistema.
- **Modelo de implementação:** agrega o *layout* visual da interface com todas as informações que descrevem os conteúdos das interfaces. Quando este modelo e o mental são coincidentes em suas especificações, os usuários se sentem à vontade com a aplicação e a utilizam de forma eficaz.



Estes modelos reconhecem as diferenças dos usuários e o projetista se propõe a utilizá-los para obter uma representação consistente da interface. Além disso, estes modelos são abstrações para que se consiga satisfazer um elemento-chave na construção do projeto de interface: “Conhecendo o usuário, você conhecerá as tarefas.”. Vejamos quais tarefas constituem o processo do projeto de interface!

O processo do projeto de interface foi estabelecido como iterativo, em que são realizadas várias iterações para se chegar à conclusão sobre a interface final e utilizam um modelo em espiral, pois possui 4 (quatro atividades principais) e ao final do modelo é construído um protótipo. Vale ressaltar que este processo engloba as etapas 2 (dois), 3(três) e 4 (quatro) estabelecidas acima (vide página 28) como parte do delineamento do processo de forma geral da interface do usuário.

Por conseguinte, vamos entender como funciona o modelo espiral e iterativo do projeto de interface do usuário (vide Figura 16):



**Figura 16 – Processo do projeto de interface**

**Fonte:** (PRESSMAN, 2016)

**Descrição da Imagem:** a imagem apresenta um plano cartesiano formado por quatro quadrantes. No primeiro tem-se o texto análise e modelagem da interface; no segundo tem-se projeto de interface; no terceiro a construção da interface e no quarto a validação da interface. Todos os quadrantes compostos por espirais azuis.

Com base na Figura 16, o modelo de processo começa do centro da espiral e engloba 4 (quatro) atividades principais:

**1. Análise e modelagem da interface:** tem como objetivo saber quais os perfis de usuário que irão interagir com a aplicação. Aqui os usuários são categorizados e algumas informações são





armazenadas como o conhecimento da área da aplicação, receptividade em relação ao sistema, nível de habilidade. De posse destes requisitos, é realizada uma análise de tarefas mais aprofundada para saber quais tarefas os usuários precisam realizar para alcançarem os objetivos do sistema. Além disso, uma análise do ambiente físico também é realizada nesta etapa. Depois de várias iterações é criado o modelo de análise para a interface.

**2. Projeto da interface:** Tendo como artefato de entrada o modelo de análise para a interface, a atividade de projeto é iniciada e assim são construídos todos os objetos e ações da interface que ajudam o usuário na realização de suas tarefas como forma de atender todas as metas de usabilidade definidas para a aplicação.

**3. Construção da interface:** esta etapa se inicia com a construção de um protótipo para avaliar os diferentes cenários de uso dos usuários. A cada iteração, alguns ajustes na interface poderão ser realizados conforme a validação deste protótipo. Vamos criar uma seção para explicar a você sobre protótipos.

**4. Validação da interface:** tem como objetivo fazer a verificação de alguns requisitos como: se a interface implementa de forma correta todas as tarefas dos usuários; se o grau de facilidade e uso está adequado para o aprendizado da interface pelos seus usuários e se o nível de aceitação está em um nível adequado para que os usuários percebam que a implementação da interface pode ser considerada como uma ferramenta essencial para que eles consigam realizar suas tarefas de forma fácil e eficaz.

Vale ressaltar que a atividade de projeto de interface, estabelece que cada etapa ocorra por diversas iterações para refinar as informações obtidas na etapa anterior. Por conseguinte, as etapas do projeto de interface são:

1. Utilizar as informações desenvolvidas na etapa de análise da interface e definindo quais os objetos e ações irão compor a interface;
2. Modelar o comportamento do estado atual da interface definindo quais eventos estão atrelados à mudança de estado do sistema;
3. Realizar a representação da forma como cada estado do sistema aparecerá para o usuário final;
4. Indicar como o usuário interpretará cada estado do sistema considerando as informações exibidas na interface.



Perceba que todas as etapas envolvidas no projeto da interface estão englobando a criação de objetos e ações da interface e de que forma poderão ser implementados e disponibilizados os diversos estados do sistema com base em variados contextos de uso do usuário.

Uma forma de começar as atividades do projeto da interface é realizar esboços desses estados da interface, ou seja, a forma que a interface é apresentada para o usuário sob diversas circunstâncias, para somente depois definir os objetos, ações e outras informações relevantes para o projeto. Desta forma, no sentido inverso, pode também funcionar dependendo de como a análise projetou a interface.

Vale ressaltar que a criação de objetos e ações são baseados em descrições de cenários de uso pelos projetistas. Sendo assim, os substantivos tornam-se objetos e os verbos ações. Veja esse exemplo: Um ícone de relatório (objeto) é arrastado e solto sobre um ícone de impressora (objeto). A ação consiste em criar um relatório impresso. Após a criação de todos os objetos e ações é possível construir o *layout* de tela, assim como, todas as outras atividades do projeto de interface.

Portanto, é importante ressaltar que independente da sequência das tarefas do processo do projeto, deve-se sempre seguir as regras de ouro estabelecidas na seção anterior, realizar a modelagem da interface com base nas atividades de análise da interface e por fim, considerar todos os ambientes a serem utilizados na construção da interface como tecnologia, sistema operacional e ferramentas de desenvolvimento.

## 2.3 Protótipos

Vimos no modelo de processo do projeto de interface que para a construção da interface são criados protótipos, mas o que são? Podemos definir como um modelo funcional que é construído a partir das especificações realizadas na fase de **requisitos** em desenvolvimento de software.



Você pode entender um pouco sobre a fase de requisitos consultando este link:  
[https://edisciplinas.usp.br/pluginfile.php/3142953/mod\\_resource/content/2/Ala09-Requisitos.pdf](https://edisciplinas.usp.br/pluginfile.php/3142953/mod_resource/content/2/Ala09-Requisitos.pdf)

Mas não se preocupe, durante o nosso curso você verá isso de forma aprofundada.





Em um protótipo serão especificadas as principais funcionalidades do software que será construído, não há necessidade de tanto detalhamento nesta fase inicial. Com o protótipo é possível identificar, avaliar, interagir com as funcionalidades principais da interface do usuário.

Adicionalmente, um protótipo é uma representação da interface do usuário de forma interativa e que são destacadas informações relevantes e assim, os usuários podem aprovar e/ou sugerir melhorias. Podemos então, afirmar que um protótipo consiste em:

- Uma simulação da interface para o usuário que pode ser feita em papel;
- Um guia para o usuário ter uma visão ampla das funcionalidades do sistema;
- Uma versão inicial do programa a ser construído.

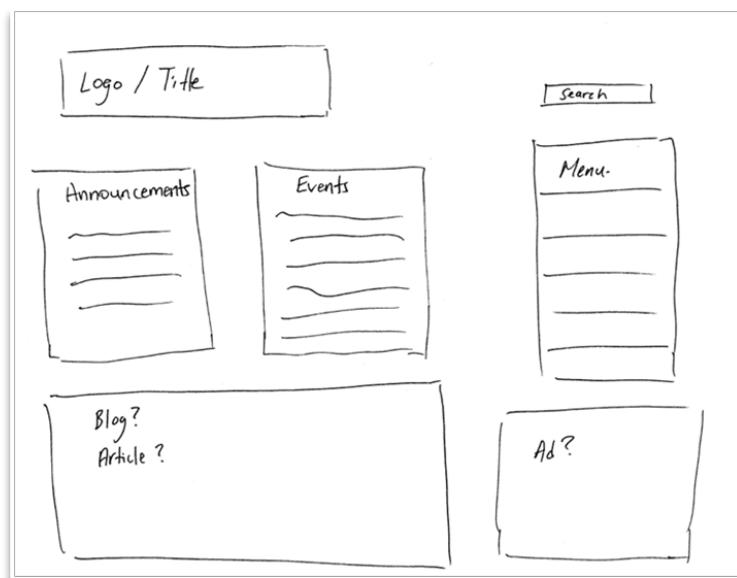
Desta forma, o protótipo tem como principal função ajustar os diferentes objetivos do projetista de interface e do usuário, no que diz respeito às diversas funcionalidades do sistema. E o protótipo é utilizado na fase inicial do projeto, em que procura se entender os problemas que os usuários possuem em suas atividades rotineiras e como um sistema pode auxiliar a minimizar tais problemas.



É nesta fase de construção do protótipo que o escopo, validação de ideias e requisitos devem ser feitas junto ao cliente para evitar futuras modificações impactantes em seu projeto, no que diz respeito à *interface*.

Agora que o conceito de protótipo está esclarecido a você, vejamos como eles se classificam. Existem 3 (três) categorias de protótipos que são:

**1 – Protótipo de baixa fidelidade:** Aqui são utilizados recursos básicos como lápis e papel e então você vai rabiscar tudo o que o cliente deseja visualizar na interface solicitada. A figura abaixo demonstra um exemplo disso:



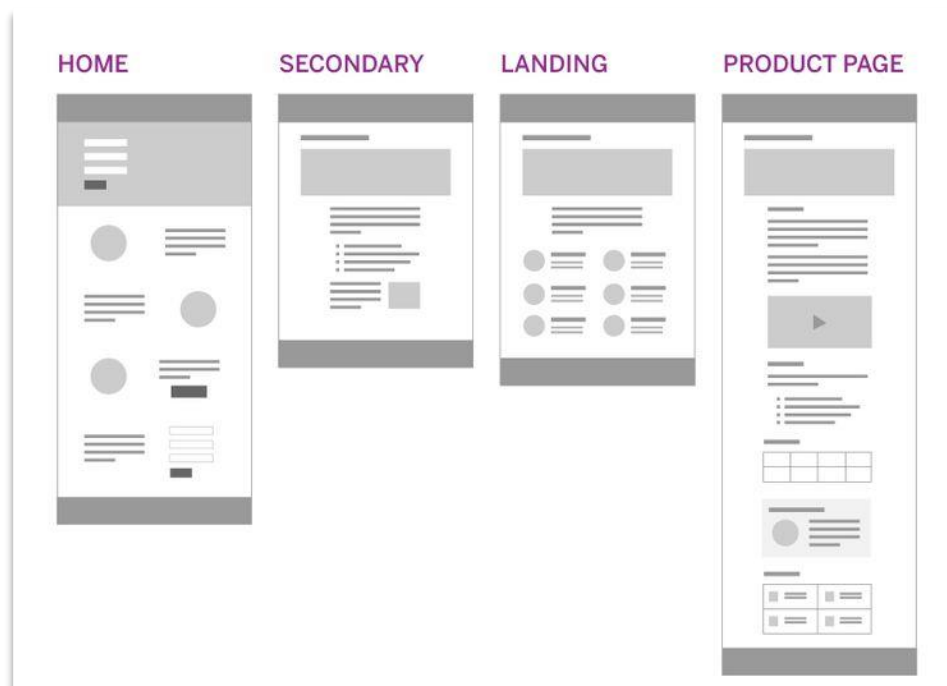
**Figura 17 - Exemplo de protótipo de baixa qualidade.**

**Fonte:** Samara (2010)

**Descrição da imagem:** a imagem apresenta um quadrado contendo outros retângulos representando componentes e textos que são necessários para construir uma interface.

Perceba que nesta figura temos o rascunho de como serão dispostos elementos importantes na interface como: *Logo*, *Menu*, Caixa de busca, Eventos, entre outros. Tudo isso faz com que o usuário tenha uma pré-visualização dos componentes que serão dispostos na interface a ser construída.

**2 – Protótipo de média fidelidade:** Para fazer este tipo de protótipo alguns softwares como *Photoshop*, *Illustrator* servem para esboçar a sua interface. Eles são ferramentas práticas, pois não precisam de tanto detalhamento e isso encurta o tempo para serem produzidos os protótipos. Vejamos um exemplo disso na figura abaixo:

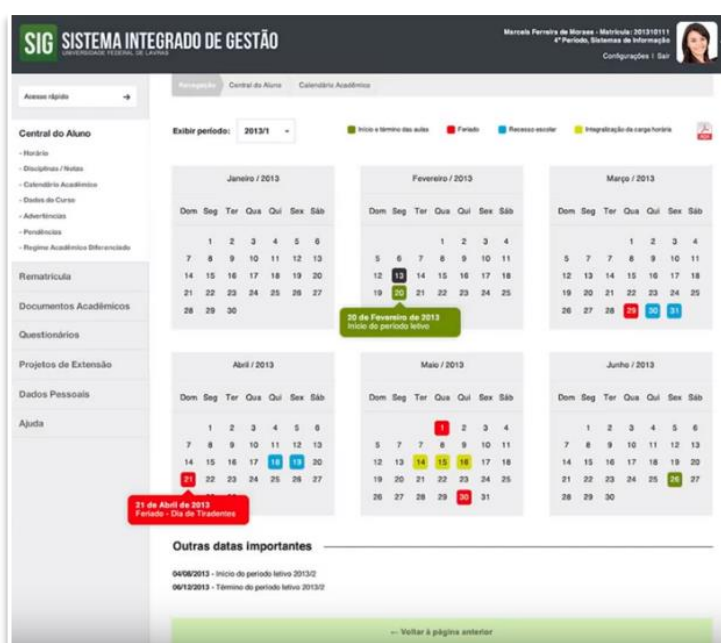


**Figura 18 - Exemplo de protótipo de média fidelidade feito no Illustrator.**

**Fonte:** Samara (2010)

**Descrição da imagem:** a imagem apresenta quatro retângulos em formato paralelo contendo outras formas gráficas como círculos e quadrados além de títulos em cada retângulo, especificando os tipos de página que o site deve possuir.

**3 – Protótipos de alta fidelidade:** É o tipo de protótipo que faz a maior aproximação com a interface real do cliente. É possível validar todas as funcionalidades, áreas que farão parte da interface e todos os requisitos junto ao seu cliente. A figura abaixo demonstra este tipo de protótipo:





**Figura 19 - Exemplo de protótipo de alta fidelidade feito no Photoshop.**

**Fonte:** Samara (2010).

**Descrição:** a imagem apresenta um quadrado que compõe um sistema integrado de gestão. Nela temos textos e calendários informativos.

Perceba que neste protótipo, para um sistema integrado de gestão, temos a visualização da interface real que será entregue para o cliente, com a criação de calendários, *menu* lateral esquerdo entre outros aspectos da interface. Cuidado ao utilizar protótipos de alta fidelidade, pois o cliente pode pensar que a programação do site esteja pronta!



Você pode encontrar algumas nomenclaturas como *wireframe*, *mockup* e isso dependem de como os autores os definem. O importante é você saber que todos são protótipos e que eles desempenham funções importantes para o projeto com seus inúmeros benefícios. Logo, use aquele tipo de protótipo que lhe for mais conveniente mediante a situação do seu projeto.

Por fim, é recomendável que você utilize os protótipos na fase inicial do projeto, já que o custo para realizar mudanças, é menor nessa fase do que em outras. Além disso, os protótipos são peças fundamentais para o sucesso do projeto, fazendo com que o esforço de desenvolvimento seja melhor utilizado e otimiza o tempo dos envolvidos.



## 3. Competência 03 | Aplicação de conceitos de usabilidade na construção de uma interface móvel

Estamos chegando a nossa última competência desta disciplina. Espero que os conhecimentos adquiridos tenham fluído de forma satisfatória e que você tenha agregado a base teórica suficiente para que possamos aplicar na construção de uma interface móvel. Mas antes disso, vejamos alguns assuntos que são interessantes para o objetivo desta competência: a elaboração da interface *mobile*.

### 3.1. Recursos para construção de interfaces IOs e Android

Nesta seção vamos abordar alguns componentes que podem ser utilizados para a construção de interfaces para os dois sistemas operacionais mais utilizados: *iOs*, da empresa *Apple*, e *Android*, pertence à empresa de tecnologia *Google*. Preparado? Vamos começar!

- **Botões:** os telefones *Android* têm um botão para trás, que pode ser usado para voltar a telas anteriores no aplicativo. Os *iPhones* não têm este botão, então é necessário uma forma de voltar para a tela anterior. Isto é normalmente feito por um botão "para trás" no topo esquerdo da tela.

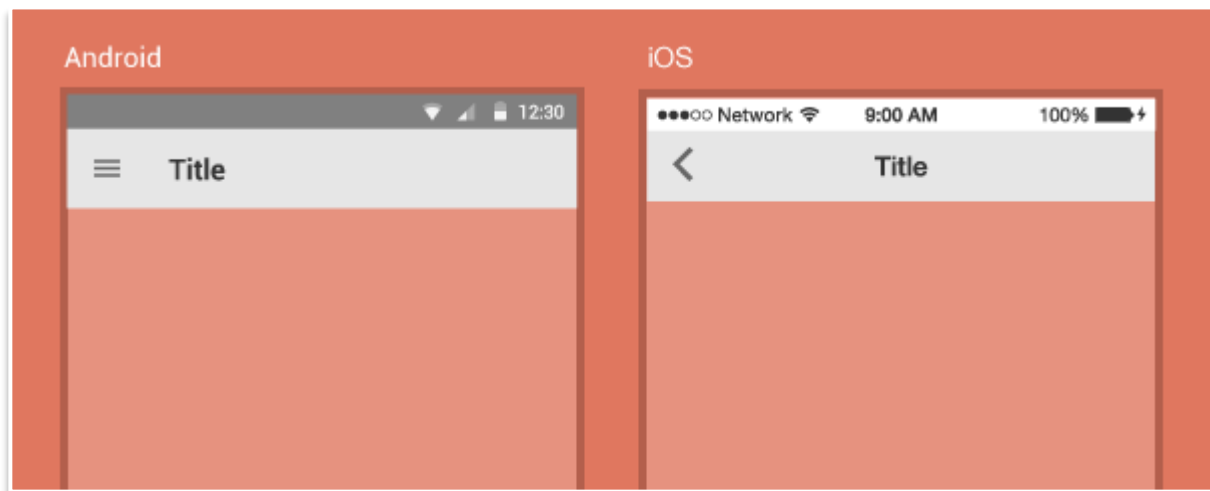


Figura 20. Botão Voltar Android

Fonte: (ANDROID MATERIAL DESIGN, 2018)

**Descrição da imagem:** a imagem apresenta um fundo preto com uma seta na direção para a esquerda na cor cinza.

- **Elementos globais:** pode-se citar alguns elementos globais como a barra de estado, que são diferentes em cada plataforma. Em *Android*, o texto é alinhado à esquerda, enquanto que em *iOS* é centralizado. Além disso, em *iOS*, muitas empresas substituem o título da página principal com o logo da sua empresa, mas isto não é a melhor abordagem para *Android*. A barra de estado (com rede, bateria, e informação temporal) é um componente nativo, e não precisa realizar alterações no *design*.



**Figura 21 - Barra de estado**

**Fonte:** a autora

**Descrição da imagem:** a imagem apresenta um fundo laranja e do lado esquerdo temos a tela do android para representação da barra de status e do lado direito a tela do iOS.

- **Barra de navegação:** talvez o maior aspecto de diferença entre as plataformas *iOS* e *Android* seja a navegação. O padrão de navegação primária em *Android* é um menu caixa. Os usuários de *Android* naturalmente vão para estes itens de *menu*, que tende a ser onipresente durante a sua experiência. Já os padrões da *Apple* são mais a favor de uma barra tabular, que está localizada no fundo da tela, e permite um fácil acesso às áreas de alto nível da aplicação. Na imagem abaixo, existem dois padrões de navegação: um menu gaveta para *Android* e uma barra tabular para *iOS*.

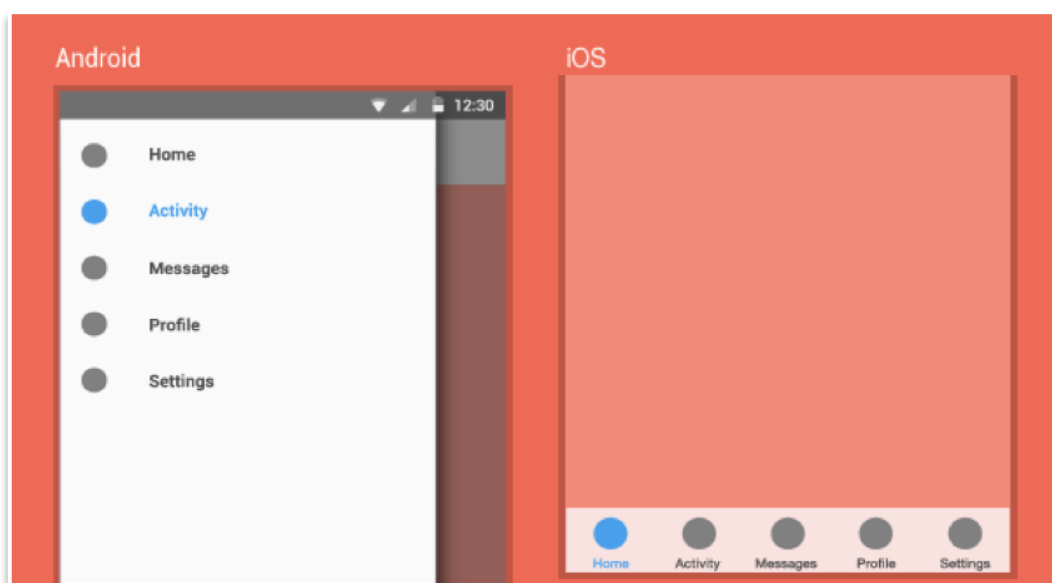




Figura 22 - Barra de navegação.

Fonte: a autora

**Descrição da imagem:** a imagem apresenta um fundo laranja e do lado esquerdo temos a tela do android para representação da barra de navegação com uma lista de opções pertencentes a um menu e do lado direito a tela do iOS em que a barra de navegação está na parte inferior da tela.

- **Tipografia:** refere-se ao tipo de letra em que é utilizado no iOS, o tipo Helvetica Neue e em Android é Roboto. Embora os estilos dos tipos de letra sejam visivelmente diferentes, a visualização do tipo de letra é muito parecida. Podemos citar algumas generalizações: (1) o Android usa espaço branco amplo no layout e um tamanho de letra maior em seu design; (2) Em iOS, há uma variação de tamanho menor, mas há uma pequena variação no peso do tipo de letra. Tipicamente, ambas as plataformas usam pesos mais leves no tipo de letra. No entanto, no exemplo abaixo, o design de Android está usando as propriedades light e regular de Roboto, enquanto que o design iOS está usando bold e regular de Helvetica Neue. Perceba que a diferença é bem sucinta.

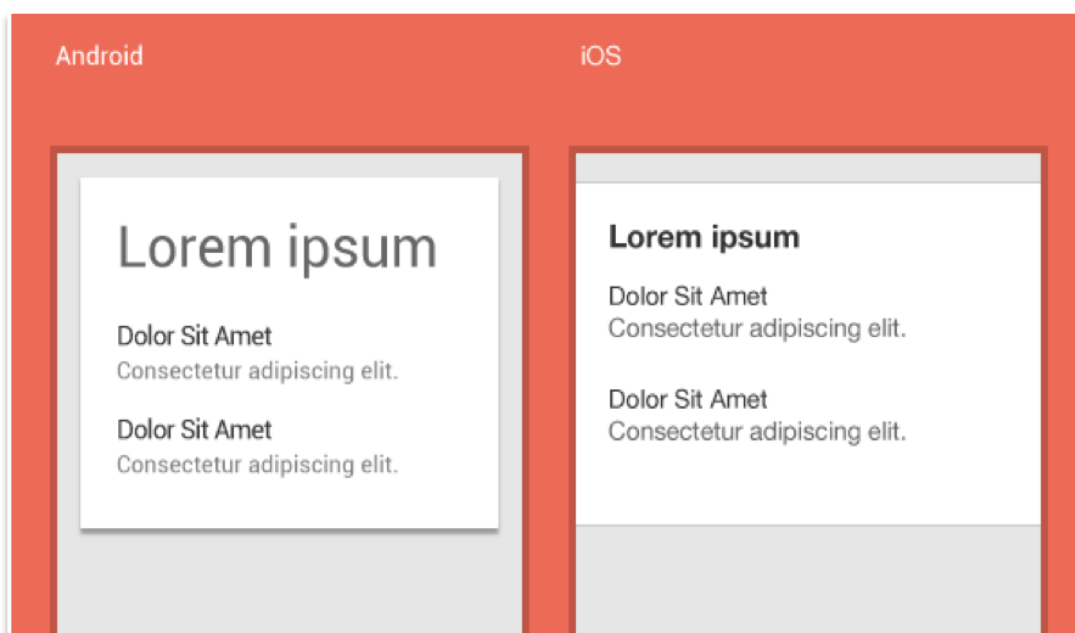


Figura 23 - Tipografia para Android e IOS

Fonte: a autora

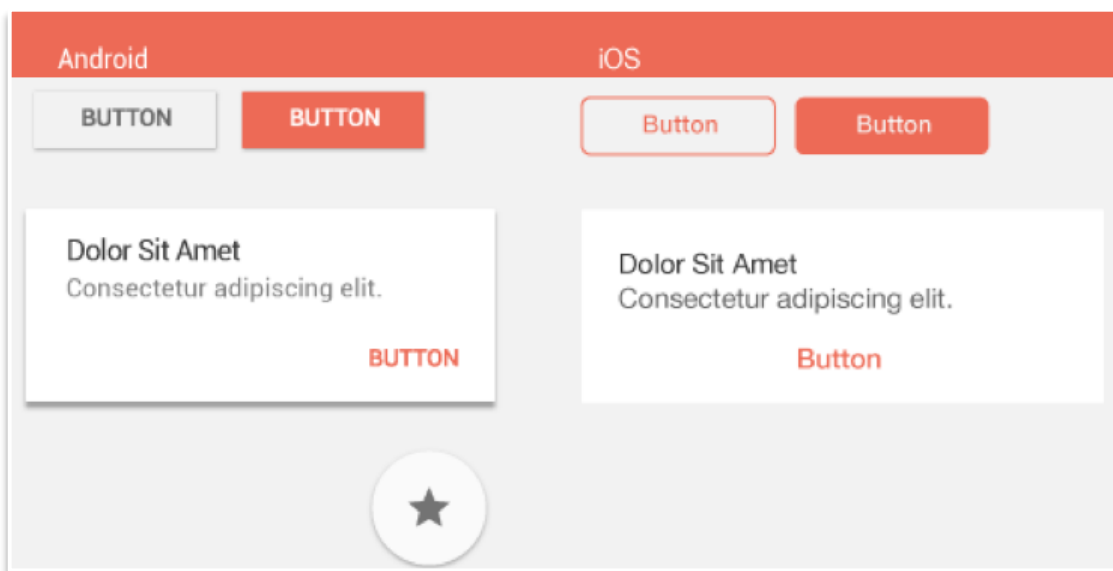
**Descrição da imagem:** a imagem apresenta um fundo laranja e dois espaços com fundo cinza e com um texto. Cada espaço desse apresenta a fonte utilizada em cada um dos sistemas operacionais: Android e iOS.

- **Estilo de botões:** As aplicações iOS são tipicamente [flat](#) em aparência, não usando profundidade ou sombras. Os botões primários têm uma cor de preenchimento, enquanto que os botões secundários são revertidos, usando um contorno da mesma





cor, também tem um botão com texto simples, mas não compartilha com o estilo de letras maiúsculas do Android, e é mais leve no peso do tipo de letra.



**Figura 24 - Estilos de botões Android e iOS.**

**Fonte:** a autora.

**Descrição da imagem:** a imagem apresenta uma parte superior com fundo laranja contendo os textos Android e iOS.

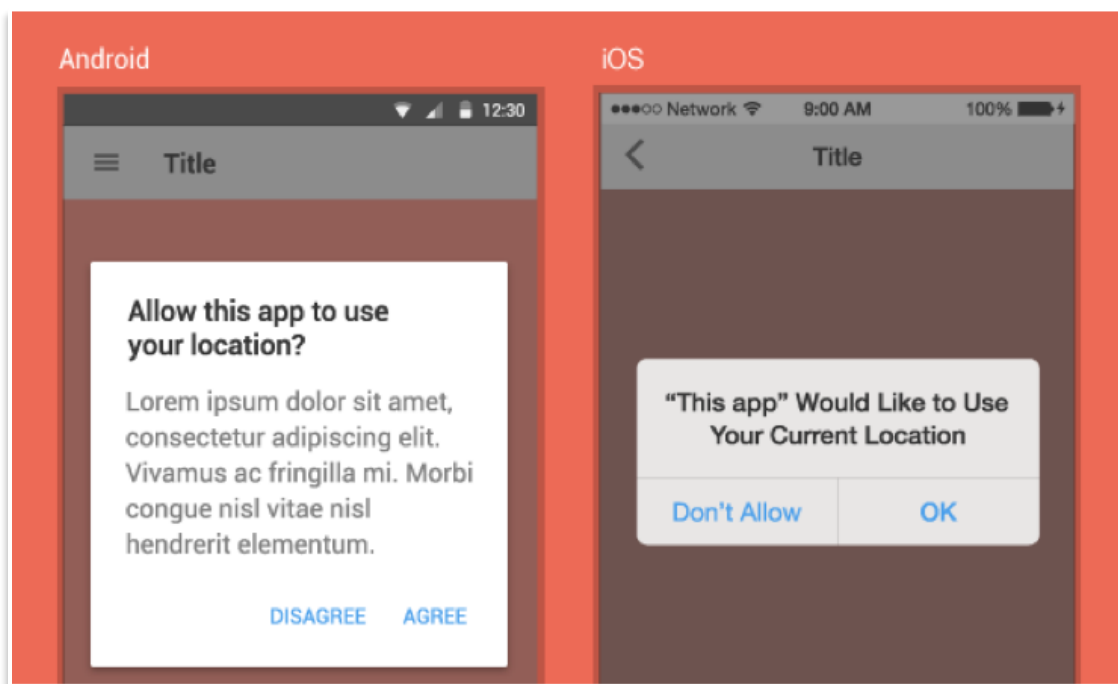
Abaixo, tem-se a representação de dois botões para cada sistema operacional Android e iOS: um com fundo cinza e outro com fundo laranja. Em seguida, tem-se duas caixas de texto para representar um conteúdo e abaixo um botão em letra maiúscula para Android e somente a inicial maiúscula para iOS.



O termo flat significa Plano e refere-se à utilização de uma única cor, sem profundidade ou efeito no componente utilizado. Quem utilizou bastante o estilo Flat foi a empresa Microsoft com as interfaces Metro. Há quem defenda e há quem recrimine este estilo Flat, pois não se deixa claro para o usuário o que é clicável.

- **Alertas:** É importante obter o estilo dos alertas corretamente, porque eles podem controlar ações importantes como inscrever, aceitar termos, ou até confirmar pagamento. Os alertas em *Android* usam o estilo de botões flat, e as ações estão no lado direito inferior do alerta. Os "botões" são na verdade inteiramente baseados em texto, utilizando letras maiúsculas. Em iOS, as ações são separadas por divisores, normalmente eles estão em frase ou título a partir dos blocos separados.



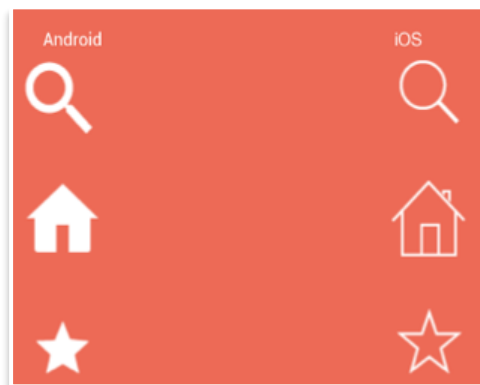


**Figura 25 - Alertas em Android e iOS.**

**Fonte:** a autora.

**Descrição da imagem:** a imagem apresenta uma parte superior com fundo laranja contendo os textos Android e iOS. Abaixo e do lado esquerdo, tem-se a representação de uma caixa de texto com fundo branco com textos em negrito e na cor cinza, além da representação dos alertas com as palavras DISAGREE e AGREE na cor azul. Já do lado direito temos um texto em negrito e dois botões na cor azul para representar um alerta em iOS.

- **Ícones:** Os ícones do sistema *Android* têm contornos mais espessos, ou são inteiramente sólidos. Anteriormente, os ícones Android usavam perspectiva ou uma visão tridimensional, mas agora os padrões para Android especificam ícones de duas dimensões. Para iOS, os ícones são mais finos e não possuem bordas espessas e nem preenchimento.



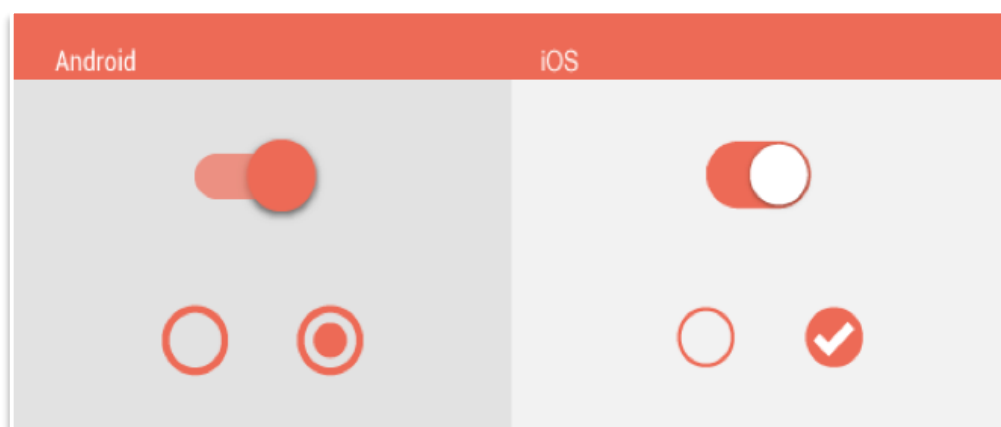
**Figura 25 - Ícones em Android e iOS.**

**Fonte:** a autora.

**Descrição da imagem:** a imagem apresenta um fundo laranja e do lado esquerdo 3 (três) ícones para Android e do lado

direito outros 3 (três) ícones para iOS.

- **Controladores de interface comuns:** Botões do tipo rádio, caixas de seleção, e switches são componentes funcionais denominados de controladores de interface. Use componentes nativos sempre que possível para estes, para que as pessoas saibam como usá-los, e confiem que os usuários saberão manuseá-los. Na imagem abaixo vemos primeiramente botões switch e rádio equivalentes para Android e iOS.



**Figura 26 - Controladores em Android e iOS.**

**Fonte:** a autora.

**Descrição da imagem:** a imagem apresenta um fundo laranja com os nomes Android e iOS. Em seguida, do lado direito tem-se um plano de fundo cinza para representar um botão switch e outro do tipo rádio para Android. Do lado direito, tem-se a mesma representação de botões.

Portanto, conhecer estas propriedades permite ao design manter-se ajustado aos requisitos de interfaces para as plataformas *Android* e *iOS* fazendo com que seja executado o projeto de forma correta.

## 3.2 Orientações sobre usabilidade

Após entender quais recursos são importantes para criação de interfaces *iOS* e *Android*, vamos prosseguir com um assunto interessante e imprescindível para que sua interface tenha uma boa experiência de usuário: usabilidade.

Antes de iniciarmos a descrever este assunto, é importante que você entenda o que é **heurística**, a qual prediz um conjunto de regras e métodos que guiam para a descoberta, invenção ou solução dos problemas, ou seja, é um método com a finalidade de encontrar soluções para os problemas relacionados à interface.



Uma outra definição de **heurística** pode ser prescrita como a criação de atalhos, regras ou até mesmo premissas que ajudam o usuário a compreender alguma funcionalidade nova. Desta forma, o projetista procura pensar em interfaces que facilitem o processo cognitivo do usuário para que ele memorize de forma mais rápida as funcionalidades e também faça com que o uso da interface seja intuitivo e mais esclarecido para o usuário.

As heurísticas de usabilidade ou de Nielsen, como comumente são chamadas, tem esse nome, pois o inventor se chama Jakob Nielsen, o qual é um cientista da computação e grande pesquisador da área interação humano - computador, construindo padrões para realizar as avaliações das interfaces.



Vamos conhecer um pouco mais sobre Nielsen em:  
[https://www.infopedia.pt/\\$jakob-nielsen](https://www.infopedia.pt/$jakob-nielsen)

Adicionalmente, um dos parâmetros que precisam ser considerado são as heurísticas que podem ser utilizadas durante ou após o projeto. No decorrer do projeto, as heurísticas têm o intuito de que o projetista realize um melhor desenho da interface com uma boa navegação, interação fluida entre usuário e interface e que aumente de forma positiva a experiência do usuário.

Já na fase de manutenção da interface, ou seja, quando a interface já está em momento de produção para o usuário, as heurísticas são utilizadas para realizar técnicas de avaliação das interfaces como forma de identificar problemas de usabilidade baseando – se nas heurísticas.

Desta forma, pode-se ressaltar a importância das heurísticas no que se refere à melhoria da usabilidade, da produtividade, já que o uso será mais intenso por parte dos usuários e aumenta também a satisfação e o engajamento dos usuários em relação à interface. Ajudam também a pensar de forma prioritária como os usuários vão interagir com as interfaces para que erros possam ser prevenidos durante a manipulação de recursos da interface.

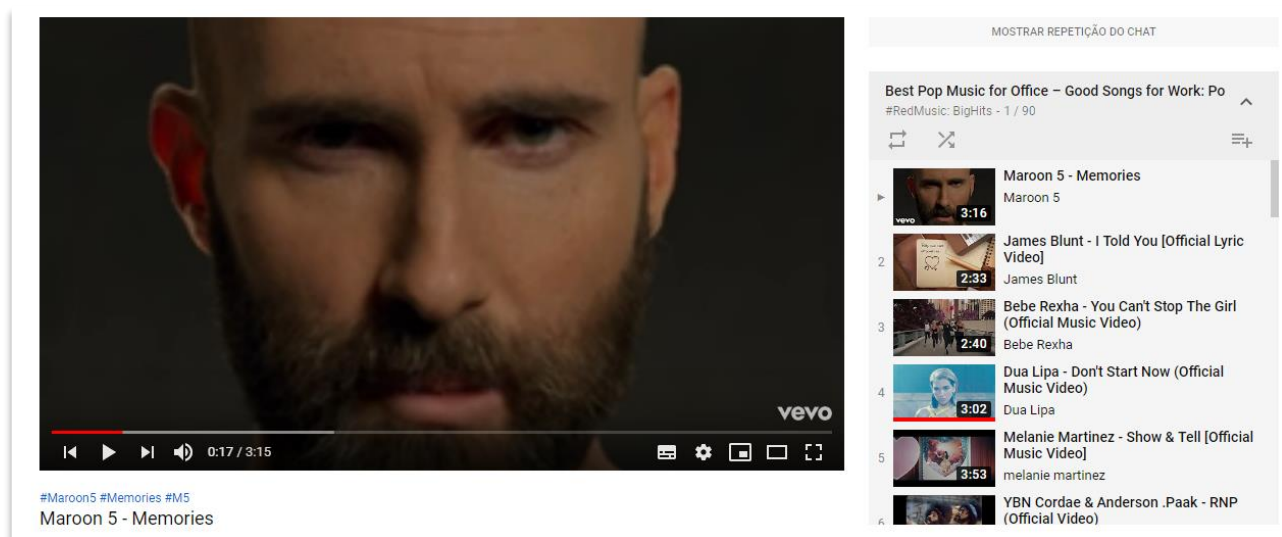
Agora, que você já sabe tudo sobre heurística, vejamos quais são as 10 heurísticas de usabilidade preconizadas por Nielsen:

## 1. Visibilidade do estado do sistema

Esta heurística informa que devem ser demonstrados para o usuário toda informação que ocorre mediante qualquer ação do usuário, ou seja, refere-se à informação para o usuário de tudo o que ocorre no sistema.

Todo sistema precisa ter configurado seus diferentes estados conforme várias ações ocorrem durante as interações dos usuários com as interfaces. Sendo assim, a visibilidade é demonstrada quando é informado ao usuário a informação que ele precisa no momento da interação. Uma forma de mostrar isso é por meio de feedbacks instantâneos.

Na Figura 27 tem-se uma *playlist* de músicas que podem ser assistidas e/ou ouvidas pelo *Youtube*. A *playlist* do lado direito emprega perfeitamente a heurística nº1 de Nielsen quando do lado direito da tela, o usuário sabe qual vídeo está sendo assistido, qual é o próximo e até mesmo quais ainda não foram vistos pelo usuário.



**Figura 27 - Playlist Youtube**

**Fonte:** <https://www.youtube.com/watch?v=SlPhMPnQ58k&list=PLKUA473MWUv2jmkqlxzQR3YL4kuPArj4G>

**Descrição da imagem:** a imagem apresenta um fundo preto e do lado direito um vídeo do canal Youtube e do lado direito uma lista de músicas a serem tocadas numa playlist chamada Best Pop Music for Office.

## 2. Correspondência entre o mundo real e o sistema

Para realizar essa correspondência, a linguagem utilizada no sistema deve atender ao vocabulário do usuário. Algumas convenções e metáforas do mundo real para o virtual devem ser utilizadas. Um exemplo é a barra de ferramentas do Photoshop que associa ferramentas do mundo real para o virtual como a ação aumentar a visualização da imagem utilizando a lupa.



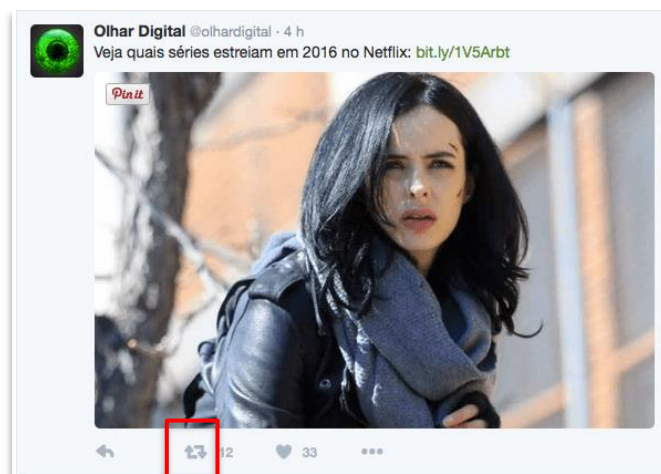
**Figura 28 - Barra de ferramentas Photoshop.**

**Fonte:** Photoshop.

**Descrição da imagem:** a imagem apresenta uma barra de ferramentas do lado esquerdo e do lado direito um fundo preto com texto que serve como plano de fundo.

### **3. Controle e liberdade do usuário**

O sistema precisa permitir qualquer cancelamento das operações do sistema e deve controlar as entradas de dados na interface, ou seja, fornecer a liberdade para o usuário das decisões e ações que ele pode realizar. O ideal é sugerir o que deve ser feito. Um exemplo disso, é quando o usuário cria um tweet, mas não pode editá-lo, afinal caso alguém dê um retweet no tweet original, é interessante que o seu conteúdo não seja modificado para que o usuário não esteja replicando conteúdos que não queira.



**Figura 29 - Barra de ferramentas Photoshop.**

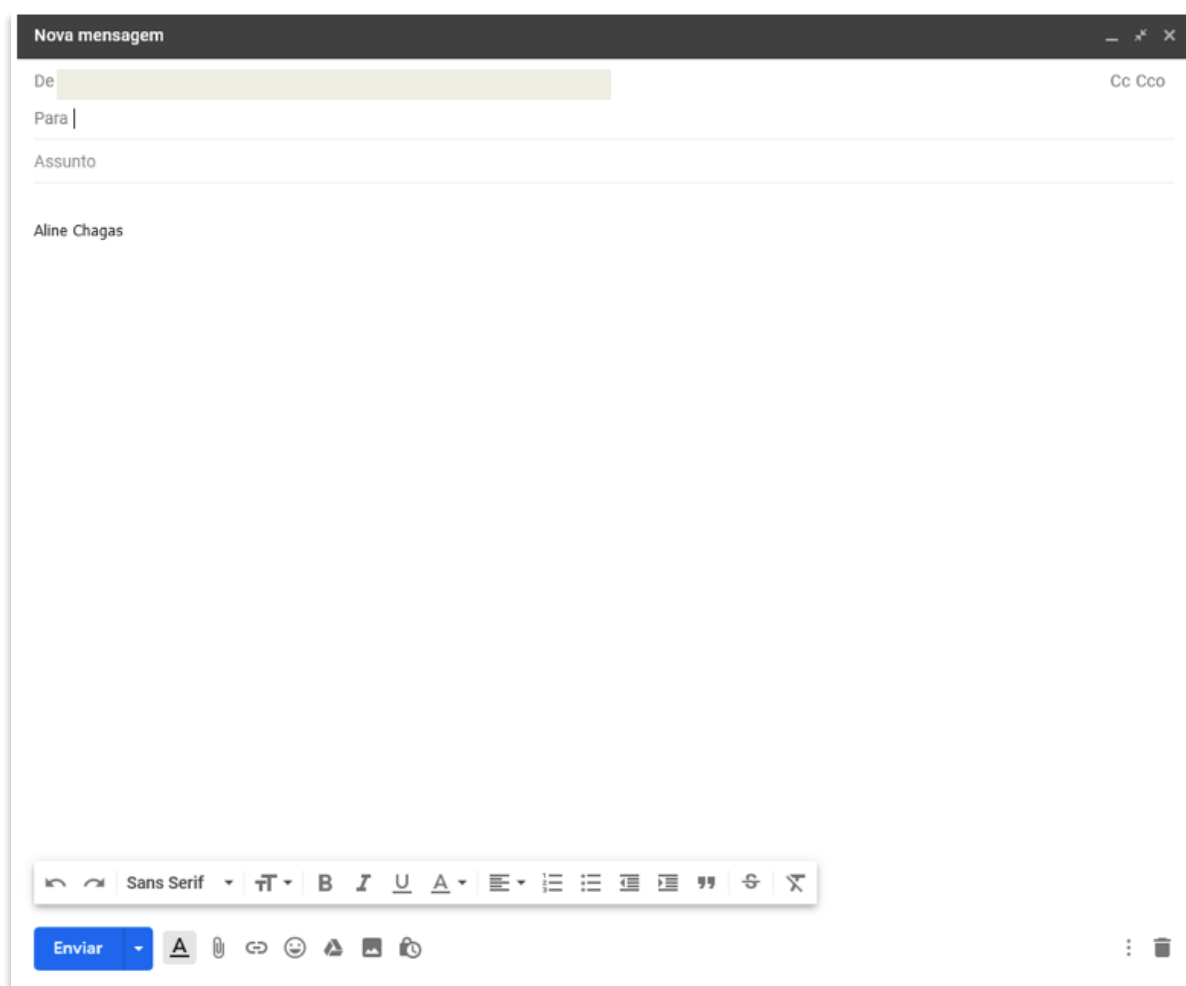
**Fonte:** Photoshop.



**Descrição da imagem:** a imagem apresenta um tweet do Olhar digital em que tem a chamada para as séries que vão estreiar na Netflix. O destaque da imagem é para a função retweet que pode ser feita sobre a publicação original que não terá seu conteúdo modificado.

## 4. Consistência e padrões

Estabelecer um padrão para determinadas funcionalidades da interface garante que o usuário crie hábitos no manuseio destas funcionalidades. Além disso, a consistência de elementos visuais é importante para manter uma interface mais harmônica na composição de seus elementos. Um exemplo é o envio de e-mail pelo Gmail que tem seu *layout* sempre consistente e padroniza algumas ações como o botão enviar sempre do lado esquerdo:



**Figura 30 - Envio de e-mail pelo Gmail.**

Fonte: gmail.com.

**Descrição da imagem:** a imagem apresenta um fundo branco em que na parte inferior tem-se uma barra de ferramentas e um botão com fundo azul com o texto Enviar.



## 5. Prevenção de erros

Caso ocorram exceções, o sistema deve informar mensagens claras sobre o erro. O ideal é que os erros sejam tratados de forma preventiva, como é o recurso do auto completar em buscadores que tenta prever um resultado para o usuário não errar.



**Figura 31 - Sugestões de palavras pelo buscador Google.**

**Fonte:** google.com.

**Descrição da imagem:** a imagem apresenta um fundo branco com uma caixa de texto para que o usuário possa inserir o texto a ser buscado.

## 6. Reconhecimento no lugar de memorização

Todos os elementos da interface devem estar bem visíveis para o usuário, não requerendo que o usuário memorize o caminho que seguiu durante a navegação na interface. Um exemplo é o recurso de informar caminhos que o usuário utilizou para se chegar em determinada tela.



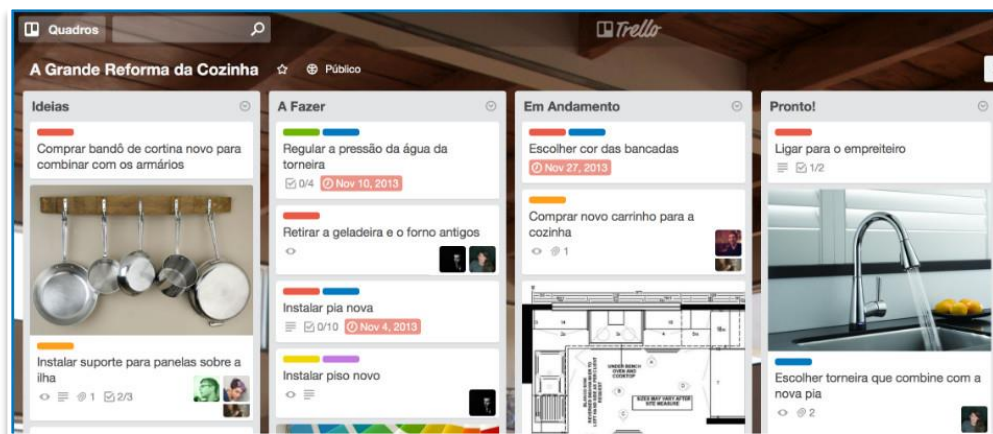
**Figura 32 - Compra de jogos do Playstation 4 no site submarino.**

**Fonte:** submarino.com.

**Descrição da imagem:** a imagem apresenta um fundo branco com vários links que informam o caminho que o usuário seguiu para chegar a página de jogos do Playstation 4.

## 7. Flexibilidade e eficiência de uso

A interface deve ser flexível em suas funcionalidades, sendo eficiente no seu uso tanto para usuários experientes quanto não experientes. Um exemplo é o Trello que possui uma interface “clean” com funcionalidades bem definidas e a movimentação dos *post-its* é muito ágil potencializando a eficiência da ferramenta.



**Figura 33 - Utilização do quadro no Trello para realizar uma tarefa.**

**Fonte:** trello.com

**Descrição da imagem:** a imagem apresenta um fundo na cor marrom com 4 (quatro) caixas que possuem os respectivos títulos: ideias, a fazer, em andamento e pronto.

## **8. Estética e design minimalista**

Compreende que tudo o que for demonstrado ao usuário podendo ser um aplicativo, sistema ou site, para que seja breve e direto, dispondo somente os elementos necessários para expor um design “clean”. Um exemplo é a utilização de ícones nos aplicativos que já representam a funcionalidade agregada:



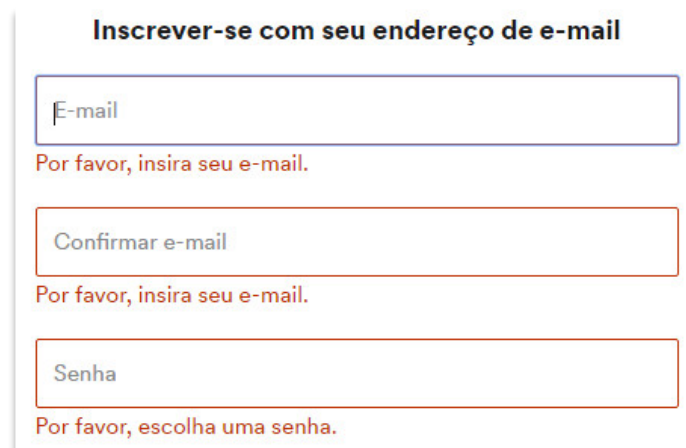
**Figura 34 - Ícones com design minimalista.**

**Fonte:** a autora

**Descrição da imagem:** a imagem apresenta vários ícones com fundos coloridos que representam funções quando utilizadas em sites e/ou aplicativos.

## **9. Ajude os usuários a reconhecer, diagnosticar e sanar erros**

As mensagens de erro do sistema precisam ter um texto simples e claro para que o usuário não fique intimidado com o erro e assim seja fornecida uma saída construtiva ou possível solução. Um exemplo são as mensagens de ajuda ao preencher um formulário errado do Spotify.



Inscrever-se com seu endereço de e-mail

E-mail

Por favor, insira seu e-mail.

Confirmar e-mail

Por favor, insira seu e-mail.

Senha

Por favor, escolha uma senha.

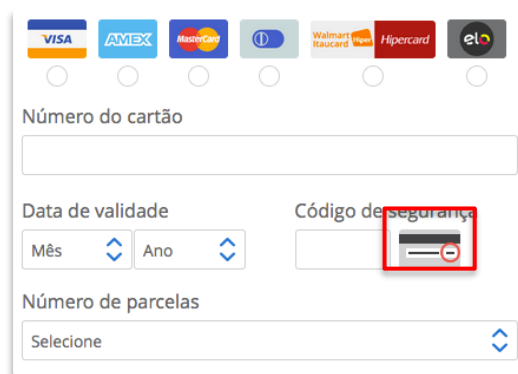
**Figura 35 - Tela de cadastro do Spotify.**

**Fonte:** <https://www.spotify.com/br/signup/>

**Descrição da imagem:** a imagem apresenta um fundo branco com 3 (três) caixas de texto que precisam ser preenchidas pelo usuário. Caso o usuário erre, são fornecidas mensagens na cor vermelha.

## 10. Ajuda e documentação

É necessário evitar a necessidade de ajuda na utilização do sistema. De toda forma, toda a ajuda e documentação deve ser fácil de localizar. Um exemplo é a opção de pagamento através de sites de comércio eletrônico que fornece a ajuda para o usuário localizar o código de segurança do cartão através de uma imagem.



Visa AMEX MasterCard Itau Walmart Itaucard HiperCard elo

Número do cartão

Data de validade

Mês Ano

Código de segurança

Número de parcelas

Selecione

**Figura 36 - Tela de pagamento.**

**Fonte:** a autora.

**Descrição da imagem:** a imagem apresenta um fundo branco com ícones que apresentam a bandeira de cada cartão, além de campos para preenchimento de um formulário para realizar o pagamento de uma compra pelo *e-commerce*.

Por fim, estes princípios definem pontos importantes para a composição das interfaces, melhoram a forma como os usuários interagem com a interface, além de oferecer uma navegação muito mais agradável, eficiente e satisfatória aos seus usuários. Agora, vamos aprofundar



um pouco mais dentro do assunto de usabilidade, explicando a você o conceito de acessibilidade digital.

## 3.2.1 Acessibilidade digital

Os conceitos de usabilidade e acessibilidade estão muito interligados. Quando você está utilizando uma interface com eficiência e eficácia, dizemos que a usabilidade foi bem aplicada e quando durante o uso, não existem barreiras digitais, ou seja, dificuldades que comprometam a experiência do usuário, podemos afirmar que a interface tem uma boa acessibilidade digital. Mas o que seria, de fato, uma acessibilidade digital? Assimile este conceito:



*“Flexibilidade proporcionada para o acesso à informação e à interação, de maneira que usuários com diferentes necessidades possam acessar e usar sistemas ou aplicações. A acessibilidade digital é mais específica e refere-se ao acesso a qualquer recurso da Tecnologia da Informação.”*

Sendo assim, podemos perceber que o grande objetivo da acessibilidade digital é não excluir os usuários em virtude de suas necessidades especiais. A acessibilidade é vista como um parâmetro de qualidade para websites e aplicações que estejam preocupados com a inclusão social.

Vale ressaltar que a acessibilidade só passou a ter uma maior ênfase a partir dos anos 2000 com o estabelecimento de algumas leis federais. E em 2008, o Decreto Legislativo nº 186 publicado no diário oficial da união ratificou o texto da Convenção sobre os Direitos das Pessoas com Deficiência assinado em *Nova York*, em 30 de março de 2007 e tornou obrigatório que todos os sites eletrônicos de administração pública ou privada tenham que ser acessíveis a pessoas com deficiência.



E como podemos promover a acessibilidade digital em nossas interfaces?



Na internet, acessibilidade refere-se principalmente às recomendações do WCAG (*World Content Accessibility Guide*) do W3C e no caso do Governo Brasileiro ao e-MAG (Modelo de Acessibilidade em Governo Eletrônico). O e-MAG está alinhado às recomendações internacionais, mas estabelece padrões de comportamento acessível para sites governamentais.

Todas estas recomendações estão atreladas aos códigos em HTML (*Hypertext Markup Language*), uso de imagens e ícones, folhas de estilo (CSS), além de proporcionar a devida interpretação das páginas pelas tecnologias assistivas utilizadas pelas pessoas com alguma deficiência.

HTML, CSS? Calma! Isso você verá durante o nosso curso. Aprenderá a realizar a programação das páginas *web* com essas linguagens de programação, que ainda são muito utilizadas na *web*. Mas eu quero explicar um outro termo que apareceu para você: tecnologias assistivas. Vamos prosseguir!

Você pode ter construído a melhor interface, o maravilhoso projeto de experiência do usuário, se ele não está atrelado às diretrizes da acessibilidade digital, isto continuará a ser uma grande barreira para as pessoas que tem algum tipo de deficiência. E como você não quer excluir os seus usuários, é bom pensar em como as tecnologias assistivas podem fazer uso do seu projeto de interface.

O nome de tecnologias assistivas refere-se a um conjunto de ferramentas (ex.: uma bengala, um cão guia) ou algum recurso (treinamento em braile) que possa ampliar as habilidades funcionais das pessoas com deficiência e assim, promova uma maior autonomia e esta pessoa não se sinta excluída devido à sua limitação.

No caso das páginas *web*, por exemplo, uma pessoa com deficiência visual (total ou parcial) a leitura das páginas é feita através de um programa denominado leitor de tela (*screen readers*) que emite uma voz ao ler as páginas *web*.

Vale ressaltar que ao projetar interfaces, a preocupação não se deve apenas a como serão utilizados os recursos computacionais para a produção das interfaces e sim, como os usuários, em suas diversas limitações, podem interagir com esta interface.

O funcionamento de uma interação com uma pessoa que tem a visão sem deficiência é feito através de um *software* (navegador) que lê e interpreta o código *HTML* e o exibe formatado para o usuário. Já as pessoas com alguma deficiência visual, precisam utilizar softwares denominados “programas leitores de tela” (*screen readers*) associados a outros programas chamados de



“sintetizadores de voz”. Os programas leitores de tela captam e interpretam o código relacionado à informação exibida na tela do computador e, por meio dos sintetizadores de voz, disponibilizam a informação em forma de som.



E quais os softwares existentes no mercado?

Hoje se utilizam, para o ambiente *Windows*, o *Dosvox/Webvox* (desenvolvido pelo NCE da UFRJ e distribuído gratuitamente), o *Virtual Vision* (desenvolvido pela empresa *Micropower*) e o *Jaws* for *Windows* (desenvolvido pela empresa *Freedom Scientific*). Há também leitores para os ambientes *Linux* e *Unix*, como o *Emacspeak* e o *Gnopernicus*, suporte pré-instalado a pessoas que têm deficiência visual aguda.



**Para saber sobre os sistemas mencionados acesse os links a seguir:**

Dosvox: <http://intervox.nce.ufrj.br/dosvox/>

Webvox: <http://intervox.nce.ufrj.br/~bernard/nossaturma/WEBVOX.HTM>

Virtual Vision: <https://www.virtualvision.com.br/>

Jaws: <https://support.freedomscientific.com/Downloads/JAWS>

Emacspeak: <http://emacspeak.sourceforge.net/>

Outro exemplo de leitor de telas, que vem ganhando popularidade é o NVDA (*NonVisual Desktop Access*), lançado em 2007, pelo australiano *Michael Curran*. O NVDA possui código-fonte aberto, gratuito e distribuído como software livre utilizando a licença pública geral versão 2, que permite o compartilhamento e alteração do código-fonte. Além da versão para instalação no computador, o NVDA tem a vantagem de possuir a mesma versão pronta para ser executada diretamente através de pendrive. Entre suas características encontram-se: sintetizador de voz gratuito e de código aberto, chamado eSpeak, uso de bips para comunicar ao usuário que barra de progresso está se movendo entre outras. Você pode encontrar outras informações através do link: <http://cegosbrasil.net/downloads/download-do-nvda>

Enfim, a área de acessibilidade digital já está sendo bastante difundida entre os projetistas de interface e há uma preocupação em como a interação pode ser melhorada para todos os tipos de usuários em suas diversas dificuldades. Pense nisso, em seu próximo projeto de interface, certo?





**Recomendações da W3C disponível em:**

[http://www.acessibilidade.net/web/ine/lista\\_verificacao\\_wai.html](http://www.acessibilidade.net/web/ine/lista_verificacao_wai.html)

Modelo de Acessibilidade em Governo Eletrônico e-MAG disponível em:

<https://www.governodigital.gov.br/documentos-e-arquivos/e-MAG%20V3.pdf>

## 3.3 Modelagem de protótipo

Já foi exposto a você todos os conceitos referentes à usabilidade e protótipos, vamos a partir de agora, aplicar esses conteúdos para que você possa iniciar um bom projeto de interface junto ao cliente.

Durante a construção de um software temos vários processos que fazem a congruência para a construção da aplicação a ser entregue para o cliente. A primeira etapa, e uma das mais importante, é a fase de **requisitos**, a qual serão estabelecidos todos os detalhes do que o cliente precisa e isso será documentado em formato texto, para que depois esse documento seja transformado em linguagem acessível para a programação.



**Você pode entender um pouco mais sobre Requisitos em:**

<https://www.devmedia.com.br/introducao-a-requisitos-de-software/29580>

Contudo, não se preocupe, pois durante o nosso curso você verá isso de forma bem mais detalhada.

De posse dos requisitos obtidos junto ao cliente, você pode começar a validar quais componentes atenderão as funcionalidades já descritas na fase de requisitos. Isso serve para que o cliente já visualize como ficará a sua interface com todos os requisitos já propostos. Isso pode ser feito através dos protótipos, que tem um grande objetivo: esclarecer ao cliente como será o seu projeto de interface em termos de visualização dos elementos das páginas.

É interessante você explicar ao seu cliente que o protótipo entregue a ele é apenas um esboço para validação das principais funcionalidades do sistema. Toda a parte gráfica será feita pelo *design* que receberá o protótipo como base para a produção das interfaces vinculadas ao sistema.





Bom, mas vamos agora colocar em prática todos os conceitos vistos até o presente momento e que ajudarão você a ter uma base prática para a criação de protótipos. Começaremos primeiramente com nossa situação hipotética:



O cliente precisa de um programa de fidelidade para sua loja virtual. Tal programa visa oferecer pontos para cada produto comprado na loja do cliente. Em sua página principal com os conteúdos informativos e menu de acesso às funcionalidades da página web (sobre a loja, produtos, contato) deve conter um link para o programa de fidelidade. O cliente acessa o programa através de um código recebido pelo número do seu telefone em uma página informativa sobre as vantagens do programa.

Com base nesse contexto, podemos estipular alguns requisitos sobre as interfaces:

## **1. Criar página inicial da loja.**

- 1.1 Na página deve conter o menu (Sobre a loja, produtos, contato) e os conteúdos informativos.
- 1.2 Criar botão Programa de Fidelidade nesta página.

## **2. Criar uma página para cadastrar cliente.**

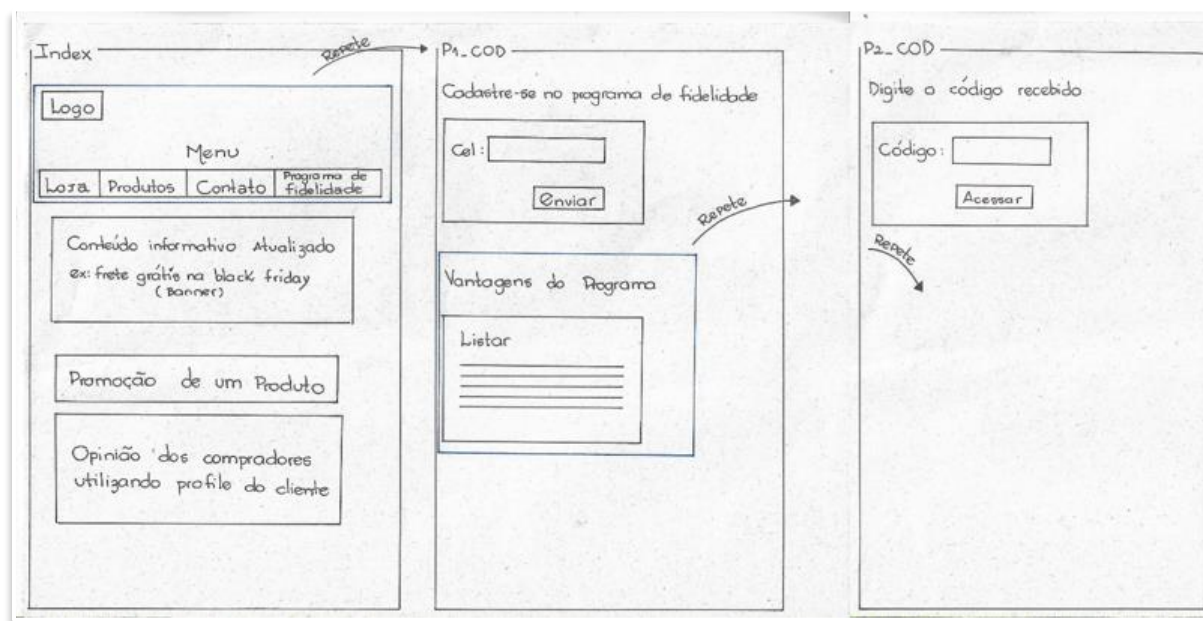
- 2.1 Criar campo para preencher o telefone do cliente e o botão Enviar.
- 2.2 Criar frame para demonstrar vantagens do programa.

## **3. Criar uma página para receber o código de validação do cliente, após ter clicado no botão enviar.**

- 3.1 Criar campo para recebimento do código e um botão chamado Acessar para ter acesso ao programa de fidelidade.

Para o item 1 criaremos uma tela chamada Index, para o item 2 uma tela chamada P1\_COD e para o 3 uma tela chamada P2\_COD. Os subitens serão agregados em cada página criada para organização dos componentes de cada tela principal.

Após receber esta lista de requisitos, podemos esboçar protótipos de baixa fidelidade conforme visto na Figura 37:



**Figura 37 - Três protótipos para representar três telas chamada Index e P1\_COD e P2\_COD**

**Fonte:** a autora

**Descrição da imagem:** na imagem temos três telas do sistema, uma chamada Index e outra P1\_COD. Na index temos o primeiro retângulo com a exposição de um menu e outros três retângulos com conteúdos variados do sistema. Na página P1\_COD temos um retângulo com campos para preenchimento do usuário e outro retângulo para descrever as vantagens do programa de fidelidade. Na P2\_COD temos um retângulo em que o usuário vai digitar um código recebido pelo celular para validação do seu cadastro.

O objetivo desse protótipo é realizar uma validação rápida com o cliente, apresentando as principais funcionalidades do sistema a ser desenvolvido. Além disso, visa especificar como será a interação do usuário com as telas do sistema.

Este protótipo foi produzindo utilizando apenas caneta e papel e traduzindo a informação disponibilizada nos requisitos e transformando para componentes de interfaces. Alguns componentes se repetiam nas telas seguintes e por isso, utilizamos a seta para a direita com o nome “repete” para informar que o componente destacado em azul vai se repetir na tela seguinte.

Para esclarecer melhor os componentes destacados em azul são o Menu (com Logo e opções do site) da página Index que vai se repetir nas telas seguintes P1\_COD e P2\_COD. Outro componente em azul que se repete é a caixa de texto nomeada como Vantagens do programa que está na página P1\_COD e se repetirá na página P2\_COD.

Por conseguinte, podemos também realizar um protótipo de alta fidelidade utilizando uma ferramenta chamada Figma que está disponível em: <https://www.figma.com/>. Esta aplicação permite a construção de protótipos de forma rápida e funcional. Um dos diferenciais dessa ferramenta é a colaboração em tempo real entre várias pessoas, o que ajuda vários usuários a



trabalharemos no *design* de uma nova aplicação, por exemplo, de forma remota e simultânea, comunicando-se entre eles e avançando no resultado.

Bom, como nós faremos somente uma parte das telas do sistema, poderemos usar algumas heurísticas de usabilidade em nossa página inicial como:

1 – Em nossa página inicial chamada *Index* vamos prover o controle e liberdade do usuário, em que o mesmo poderá, se quiser, acessar de forma direta o programa de fidelidade ou ele pode querer ver primeiramente os produtos em promoção.

2 – Para a heurística de consistência e padrões vamos manter nos botões que requerem ação do usuário, a mesma cor de fundo. Como o botão Enviar para que as informações do telefone sejam enviadas e o botão Acessar para que o usuário possa ter acesso ao programa de fidelidade. As cores podem até mudar no *design* final, mas o interessante é demonstrar essa consistência.

3 – A flexibilidade e eficiência de uso serão atribuídas na visibilidade das principais funcionalidades do sistema como ter um menu disponível, o acesso rápido ao programa de fidelidade, e o menu de promoções.

Antes de produzir o protótipo de alta fidelidade percebeu-se um pequeno erro no Menu. A palavra Contato e Programa de Fidelidade estavam na ordem errada. Então, alterou-se para a seguinte ordem: Sobre a loja, Produtos, Programa de Fidelidade e Contato. É importante frisar que mudanças podem ocorrer na fase de transição dos protótipos e isso é normal. Melhor que o erro ocorra nessa parte do desenvolvimento do sistema do que na programação, certo?

A visualização do protótipo de alta fidelidade feita no Figma pode ser conferida na Figura abaixo:



**Figura 38 - Tela Index feita no Figma.**

**Fonte:** figma.com

**Descrição da imagem:** a imagem apresenta uma página chamada Index, em que temos um retângulo cinza escuro com textos que formam um menu. Em seguida temos outro retângulo para inserir conteúdos do site. A seguir temos 6 quadrados que servem como categorias de promoções e opiniões de compradores.

Vale ressaltar que também utilizamos as dimensões de uma tela para celular com o sistema *Android* e suas demais configurações relacionadas à tipografia como: Fonte Roboto e uso de letras maiúsculas nos botões das interfaces do sistema. As demais telas do sistema relacionadas à interação com o programa de fidelidade poderão ser visualizadas em nossa videoaula desta competência. Confira agora em seu ambiente virtual de aprendizagem (AVA)!



## Conclusão

Enfim, chegamos ao final deste e-book e muito ainda poderia ser explicado sobre a área de interação humano – computador, mas pelo menos você compreendeu a boa parte que precisa para quem sabe, em um futuro, se tornar um bom projetista de interfaces do usuário.

Durante as três semanas de curso você pode compreender a finalidade das interfaces e a aplicabilidade em nosso dia-a-dia, como as interfaces evoluíram ao longo da história dos computadores, como ocorre um processo interativo e entendeu o que faz um usuário gostar de usar uma interface, a boa usabilidade que existe nela.

Por conseguinte, vimos alguns conceitos mais técnicos relacionados aos processos inerentes à construção do software e percebeu que para construir uma interface do usuário são necessárias diversas etapas para que se tenha sucesso, afinal, não pode haver distância semântica entre o que o cliente solicita e o que realmente é projeto no sistema. Caso isso ocorra, creio que será um tanto constrangedor!

Por fim, foram descritos os principais conceitos de protótipos e como modelar um protótipo com base nos requisitos do cliente, pode facilitar a vida de todos os envolvidos na construção dos sistemas e provavelmente você não escutará a frase do cliente: “Ah! Mas não foi desta forma que pensei que seria o sistema!”.

Portanto, espero que os conceitos visualizados neste e-book possam agregar em sua carreira profissional e que possamos cada vez mais humanizar as telas informatizadas para telas mais acessíveis a todos que queiram utilizá-las.

Bons estudos e nos veremos na próxima!



## Referências

NIELSEN, Jakob. *Projetando Websites*. CAMPUS, 2000.

KRUG, Steve. *Não me faça pensar*. ALTA BOOKS, 2006.

NIELSEN, J.; LORANGER, H. *Usabilidade na Web: Projetando Websites com Qualidade*. Rio de Janeiro: Campus, 2007. ISBN 8535221905.

NIELSEN J (1993) *Usability Engineering*. Academic Press.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016.

SOMMERVILLE, I. *Engenharia de software*. Tradução de Kalinka Oliveira e Ivan Bosnic. 9. ed. São Paulo: Pearson, 2011.





## Minicurrículo do Professor

### Aline Chagas Rodrigues Marques



Mestrado em Ciência da Computação no Centro de Informática (CIN) na Universidade Federal de Pernambuco (UFPE). Possui especialização em Análise de Sistemas pela Universidade Federal do Pará (2011) e graduação em Bacharelado em Ciência da Computação pelo Centro Universitário do Estado do Pará (2009). Atuou como Técnica em Informática no Instituto de Previdência e Assistência do Município de Belém (IPAMB). Atualmente é professora do curso técnico de Desenvolvimento de Sistemas, na modalidade EAD, na Secretaria de Educação do estado de Pernambuco, conteudista das disciplinas de Engenharia de Software e Interação Humano – Computador e tutora na Unyleya no curso de User Experience (UX) e User Interface (UI).

### Pedro Henrique Barboza da Silva



Mestre em Informática Aplicada (2017) e Licenciado em Computação (2015) pela Universidade Federal Rural de Pernambuco, onde atua como pesquisador do grupo TECNES - Informação Educação e Tecnologia Colaborativa em Saúde. Atualmente é professor do curso técnico de Desenvolvimento de Sistemas, na modalidade EAD, na Secretaria de Educação do estado de Pernambuco e professor do ensino superior nos cursos de Sistemas de Informação e Análise e Desenvolvimento de Sistemas.



