

# DESCRIÇÃO

Processos de desenvolvimento de interfaces humano-computador, utilizando princípios, diretrizes e técnicas de design de interface.

# PROPÓSITO

Compreender os processos e as principais técnicas de desenvolvimento de interface humano-computador e saber como aplicá-los em sistemas de software são conhecimentos essenciais para os profissionais de desenvolvimento de sistemas, especialmente na Web.

# OBJETIVOS

## MÓDULO 1

Descrever os elementos a serem definidos em um projeto de design de interface humano-computador

## MÓDULO 2

Descrever técnicas de concepção e modelagem de interface humano-computador

## MÓDULO 3

Reconhecer o processo de design de interface humano-computador

## MÓDULO 4

Definir princípios e diretrizes para o design de interfaces humano-computador

# INTRODUÇÃO

A interface de um software pode, de fato, determinar seu sucesso ou fracasso. É muito difícil encontrar um software de sucesso cuja interface não tenha boas características de usabilidade. O design de uma interface tem que respeitar diversos critérios para que o software tenha aceitação por parte dos usuários. Por mais funcionalidades que o software implemente e por melhores que sejam, se a interface não é boa ele poderá até mesmo deixar de ser utilizado. Toda essa visão sobre a interface tem momento adequado para ser pensada durante o processo de desenvolvimento de um software. Em geral, temos o que podemos chamar de fase de projeto de interface. Saber identificar que características de usabilidade são importantes para aquele software e como aplicá-las é importante para que o design da interface saia o mais ajustado possível às necessidades do usuário.

Entenderemos como projetar o design de uma interface respeitando as necessidades de todos os envolvidos e as capacidades técnicas das ferramentas disponíveis para construção do software. Veremos as técnicas de concepção e de modelagem dessas interfaces assim como os processos para sua implementação. Por fim, apresentaremos e discutiremos princípios e diretrizes da área de design de interface em IHC.

O ponto mais importante neste conteúdo é que teremos uma visão geral de todas as partes do processo de desenvolvimento de software que envolvem o desenvolvimento de interfaces. O desenvolvimento de interfaces é muito abrangente e por isso possui vários desdobramentos.

# MÓDULO 1

---

- ⦿ Descrever os elementos a serem definidos em um projeto de design de interface humano-computador

## PROJETO DE DESIGN DE INTERFACE HUMANO-COMPUTADOR



O projeto de design de IHC visa elaborar um modelo conceitual de entidades e atributos do domínio e do sistema, estruturar as tarefas e projetar as interações e a interface de um sistema interativo que apoie os objetivos do usuário. Visa especificar como será a interação entre o usuário e o sistema. Para isso, precisamos não só entender o problema do usuário, mas também como este executa suas atividades e quais são suas necessidades que devem ser atendidas com o uso do sistema.

O projeto de design de interface se preocupa com muitos elementos, como:

Parte visual.

Usabilidade.

Arquitetura da informação.

Navegação.

Transição de telas, entre outros.

Tudo deve ser pensado e planejado para que a experiência dos usuários seja de satisfação durante o uso do sistema. Vem daí a ideia de **experiência do usuário ou UX**.

No projeto de design da interface, o usuário é o centro das atenções no desenvolvimento do sistema.

Para desenvolvermos bem um projeto de design de uma interface, alguns conceitos precisam estar bem sólidos.

Alguns desses, como os conceitos de **usabilidade**, **Engenharia Semiótica**, **interação** e de **interface**, são objeto de estudo da **Ergonomia em IHC**.

Neste módulo, estudaremos os conceitos de **affordance** e de **comunicabilidade**.

## AFFORDANCE

Trata-se de um termo em inglês sem tradução no português, significando **o conjunto de características do software perceptíveis pelo usuário**, as quais indicam que tipos de operações podem ser realizadas com o sistema interativo, bem como as formas de realizá-las manipulando a interface.

Na área de IHC, segundo Norman (1998), a affordance de um objeto é exatamente o conjunto das características capazes de mostrar aos usuários as operações que eles podem fazer com ele.

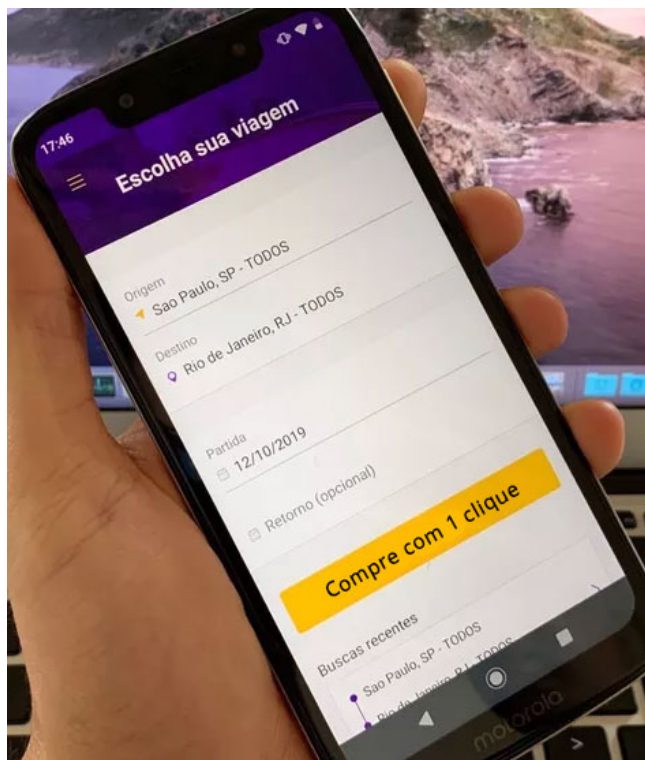
Affordance é a capacidade que um objeto tem de ser reconhecido e utilizado exatamente para o que foi projetado, mas sem a necessidade de uma explicação prévia.

Em geral, todo esse conhecimento é baseado em padrões. Padrões são úteis, pois somos capazes de reduzir a carga cognitiva dos usuários quando interagem com um determinado elemento.

Para entender um pouco melhor, vejamos os 4 (quatro) tipos existentes de affordance com exemplos a seguir:

## EXPLÍCITO

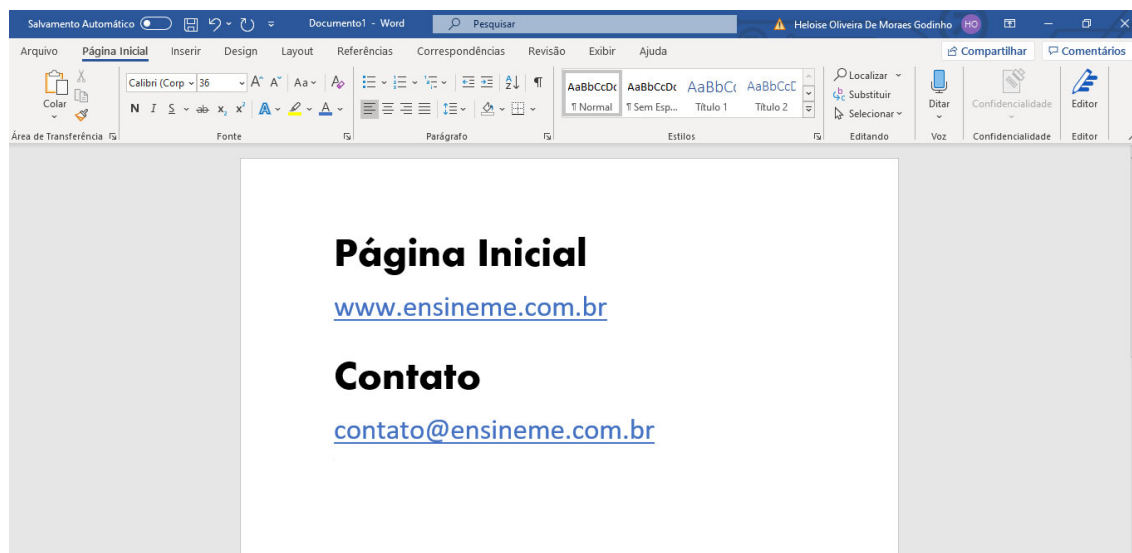
Objeto acompanhado de um texto indicando exatamente o que o usuário deve fazer. Exemplo: Um botão de “Compre com 1 clique”.



## CONVENCIONAL/PADRÃO

Baseado em experiências anteriores do usuário ou em informações que são conhecidas pela maioria dos usuários.

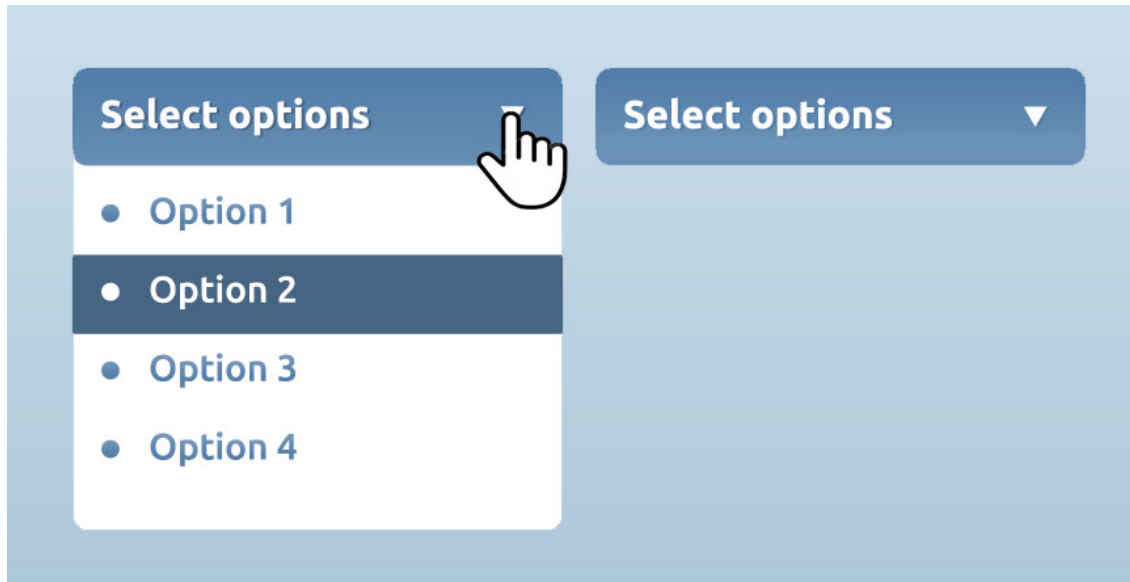
Exemplo: um link sublinhado com a cor azul.



# OCULTO/ESCONDIDO

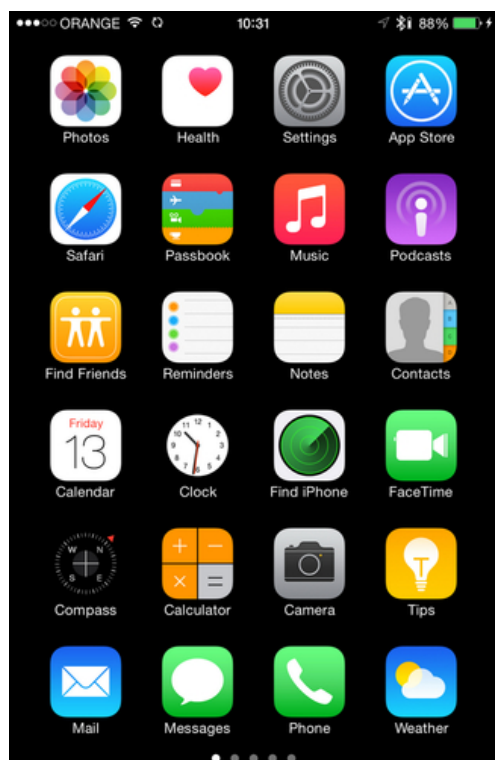
É utilizado para diminuir a complexidade de uma interface, porém pode tornar a interface difícil de usar se o usuário não estiver acostumado.

Exemplo: Menu dropdown, que só mostra seu conteúdo quando passamos o mouse por cima de um item do menu.



# METAFÓRICO

Utiliza objetos do mundo real como metáforas para ajudar no entendimento da interface. A intenção é ajudar a comunicar ideias abstratas sem muitas instruções. Exemplo: Ícones de aplicativos em telas de smartphones.



## COMUNICABILIDADE

É a propriedade de transmitir ao usuário, de forma eficaz e eficiente, as intenções da interação com a interface.

Os sistemas que têm essa característica expressam bem a intenção e a lógica do design do sistema, permitem aos usuários expressar sua intenção de uso e respondem com modelos de comunicação adequados ao contexto de uso. A intenção do designer deve ser sempre remover as barreiras da interface que impedem o usuário de interagir, tornando o uso fácil e comunicando ao usuário as suas concepções e intenções ao conceber o sistema interativo.

O conceito de comunicabilidade foi proposto pela **Engenharia Semiótica** (elemento que também é objeto da área de Ergonomia em IHC).

## ENGENHARIA SEMIÓTICA

A Engenharia Semiótica vê a Interação Humano-Computador (IHC) como uma comunicação mediada por computador entre designers e usuários em tempo de interação. O sistema fala em nome de seus criadores em vários tipos de conversas especificado em tempo de design. Essas conversas comunicam aos designers a compreensão de quem são os usuários, o que eles sabem que os usuários querem ou precisam fazer, quais as suas formas preferidas e por quê. A mensagem dos designers para os usuários inclui até mesmo os idiomas interativos que os usuários terão para se comunicar com o sistema a fim de atingir seus objetivos específicos. Então, o processo é, na verdade, uma comunicação sobre a comunicação, ou metacomunicação.

Fonte: Wikipédia

A comunicabilidade diz respeito à capacidade da interface de comunicar ao usuário a lógica do design, as intenções do designer e os princípios de interação resultantes das decisões tomadas durante todo o processo de design.

**SE UM USUÁRIO É CAPAZ DE COMPREENDER A LÓGICA UTILIZADA NA CONCEPÇÃO DO SISTEMA INTERATIVO, TERÁ MAIORES CHANCES DE FAZER UM USO CRIATIVO, EFICIENTE E PRODUTIVO DELE.**

(PRATES; BARBOSA, 2007)

O entendimento dessa lógica de design permite que os usuários usem melhor o sistema.

## **★ EXEMPLO**

Não precisamos saber como funcionam os recursos de estilos de formatação ou numeração automática de um editor de texto para utilizá-lo, mas se sabemos isso certamente podemos usar esse software de forma mais eficiente e obter menos erros.

Segundo Prates *et al.* (2000), a intenção é que o usuário saiba:



**PARA QUE SERVE O SISTEMA.**

**QUAL A VANTAGEM DE UTILIZÁ-LO.**

**COMO ELE FUNCIONA.**

**QUAIS SÃO OS PRINCÍPIOS GERAIS DE INTERAÇÃO COM O SISTEMA.**

## **FASES DE UM PROJETO DE INTERFACE HUMANO-COMPUTADOR**

Segundo Rogers, Sharp e Preece (2013), as fases de um projeto de IHC são principalmente:

### **IDENTIFICAR AS NECESSIDADES E DEFINIR OS REQUISITOS**

Nesta atividade, deverão ser entendidos os processos de trabalho dos usuários, verificadas quais as suas reais necessidades frente ao uso do sistema para apoio aos seus processos de trabalho e listados todos os requisitos de negócio que esse sistema deverá suportar.

### **DESENVOLVER DESIGNS ALTERNATIVOS**

Nesta atividade, devem ser explorados vários aspectos com relação ao visual e usabilidade do software. Cenários de interação devem ser criados para serem avaliados e escolhidos pelos usuários.

# CONSTRUIR VERSÕES INTERATIVAS DOS DESIGNS

Nesta atividade, são desenvolvidos protótipos que permitem ao usuário vivenciar a visão da interface em um software próprio para prototipação. Isso ajuda muito a esclarecer os requisitos da interface.

## AVALIAR O DESIGN

Nesta atividade, as alternativas de design apresentadas nos protótipos são avaliadas e ranqueadas através de critérios definidos previamente, como: Aparência, quantidade de requisitos satisfeitos, usabilidade, entre outros.

# DEFININDO AS REPRESENTAÇÕES DO PROJETO

Para registro das informações coletadas no processo de levantamento de requisitos precisamos de representações.

Aqui falaremos sobre diversas representações e modelos utilizados para registrar, organizar, refinar e analisar os dados coletados, dentre eles: Perfil de usuário, personas, cenários de análise ou de problema e modelos de tarefas.

## 1) PERFIL DO USUÁRIO

Precisamos definir o perfil do usuário. É para ele que construiremos o sistema. Devemos fazer uma descrição detalhada de suas características. Precisamos conhecer seu domínio sobre a área de atuação do sistema e sobre as tarefas que deverá realizar utilizando o sistema.

A intenção desse levantamento de informações sobre cada usuário é poder categorizar os mesmos em grupos, com base em semelhanças, principalmente verificando que usuários poderão realizar que tarefas no sistema.

## 2) PERSONAS

É um termo que representa um grupo hipotético de usuários.

Personas são definidas, principalmente, por seus objetivos de atividade no sistema em geral.

Uma boa prática citada por muitos autores é termos poucas personas para um sistema. Muitas delas em paralelo geram muitos conjuntos e isso pode complicar muito as funcionalidades do sistema, pois vamos ter que “agradar” muitos pontos de vista diferentes.

Uma persona deve ter:

## IDENTIDADE

A persona deve ter nome, idade, um avatar (foto) e dados demográficos.



## STATUS

A persona deve ter uma categoria como stakeholder, usuário primário, administrador ou outras.



## OBJETIVOS

A persona deve ter seus objetivos definidos com relação ao sistema.

## HABILIDADES

A persona deve ter especificadas as suas habilidades e competências específicas.



## TAREFAS

A persona deve ter definidas as tarefas que realiza normalmente, com quais frequências.

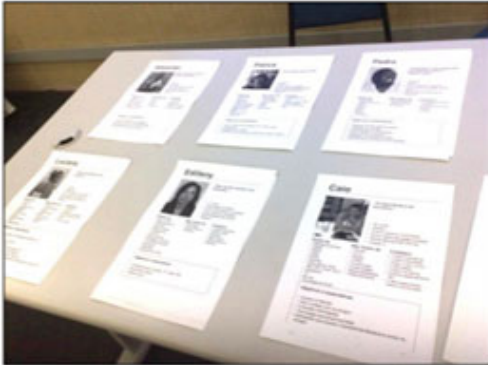


## RELACIONAMENTOS

A persona deve ter elencadas as outras personas com quem se relaciona.

O importante é que todos na equipe tenham a mesma visão sobre quais são os tipos de usuários do sistema e que possam entender como irão atuar nele. Cada projeto deve possuir seu próprio **elenco de personas**.

Cada elenco de personas possui ao menos uma **persona primária** que é foco principal do design.



## Caio



*O importante é ser exclusivo*

- 22 anos
- Solteiro
- Ensino superior incompleto
- Zona urbana da região sudeste
- Renda Mensal R\$7.000,00
- Classe A

<b>Gosto de</b> <ul style="list-style-type: none"> <li>• Bebidas</li> <li>• Festas</li> <li>• Status</li> <li>• Viajar</li> <li>• Flertar</li> <li>• Esportes</li> <li>• Video games (Xbox 360, PS3, Guitar Hero)</li> <li>• Importados</li> <li>• TV</li> <li>• Blu-ray</li> <li>• Tecnologia de ponta</li> </ul>	<b>Não Gosto de</b> <ul style="list-style-type: none"> <li>• Estudar</li> <li>• Livros</li> <li>• Regras</li> <li>• Limites</li> <li>• Compromisso</li> <li>• Coisas antigas</li> <li>• Responsabilidades</li> <li>• Festas familiares</li> </ul>	<b>Cotidiano</b> <ul style="list-style-type: none"> <li>• Ir para a academia</li> <li>• Ir para a faculdade</li> <li>• Dirigir o carro (novo)</li> <li>• Ir para festas</li> <li>• Ir para o shopping</li> <li>• Cinema (especialmente 3D)</li> <li>• Ir para restaurantes, bares e lanchonetes</li> </ul>
--	---	--

**Objetivos e expectativas**

- Acesso a internet
- Fácil contato com os amigos
- Consultar informações
- Tecnologia exclusiva/importada
- Aplicações que possam impressionar/destacá-lo entre os amigos

Exemplo de cartões de personas.

Fonte: CARVALHO, 2001.

### 3) CENÁRIOS

Trata-se de uma história sobre pessoas executando atividades a fim de realizar algo ou atingir um objetivo. É uma narrativa que inclui contexto sobre uma situação de uso de um sistema.

Os cenários podem ser utilizados para descrever como algo acontece, com bastante riqueza de detalhes, de modo que promova o melhor entendimento sobre a situação. Mas o mais importante é que os cenários descrevem o comportamento dos atores (personas).

Em geral, possui um enredo que inclui sequências de ações e eventos: O que os usuários fazem, o que acontece com eles, que mudanças ocorrem no ambiente, entre outras coisas. Tem sempre um ator principal e um objetivo principal. Tem também um título que descreve brevemente a situação.

Os elementos característicos de um cenário são:

# AMBIENTE

Detalhes da situação.



# ATORES

Pessoas que interagem no cenário.



# OBJETIVOS

O que motiva as ações dos atores.



# PLANEJAMENTO

atividades que transformam o objetivo em comportamento.

# AÇÕES

Comportamentos.



## EVENTOS

Ações externas ou reações internas.



## AVALIAÇÃO

interpretação da situação

**OS CENÁRIOS PODEM INCLUIR EXCEÇÕES, OU SEJA, EVENTOS QUE ACONTECEM RARAMENTE.**

## ★ EXEMPLO

Sérgio, professor de Computação de uma universidade, precisa utilizar o sistema desta universidade para cadastrar as notas dos alunos no final do semestre.

Como ele é novo na universidade, está fazendo isso pela primeira vez. Antes de iniciar, ele precisa corrigir as provas e ter uma nota única por aluno. No sistema só é permitido um lançamento por bimestre. Ele precisa também descrever se o aluno está tendo algum problema de aprendizagem ou não.

Ele recebeu um treinamento neste sistema assim que foi admitido na universidade. Ele teve que fazer todas as configurações para seu perfil, que é sem privilégios de administração no sistema. Sua disciplina e os alunos já estavam cadastrados no sistema pelo administrador da secretaria.

No entanto, durante as aulas do primeiro semestre, teve a notícia da suspensão das aulas presenciais por uma semana por causa de uma olimpíada de Matemática que iria acontecer na universidade e muitos dos alunos de Computação estavam inscritos.

Com isso, teve que adiar as provas e o calendário de suas aulas. Como o calendário era diferente para cada disciplina, afetada ou não pela olimpíada, cada professor ficou responsável por esta alteração.

Para realizar esta ação ele vai na função de calendário, vai até o mês atual e atualiza datas e horários das suas aulas em função das datas que recebeu das olimpíadas.

Os alunos de Computação podem estar participando e, portanto, com esta alteração passam a ter a visão das mudanças de suas aulas presenciais.

## **4) ANÁLISE DE TAREFAS**

Esta análise é utilizada para se ter um entendimento melhor sobre qual é o trabalho dos usuários. Nesse tipo de análise, o trabalho não é apenas elencar tarefas, mas entender como o sistema irá afetar esse trabalho realizado.

Um dos primeiros passos numa análise de tarefas é coletarmos o conjunto de objetivos das pessoas.





Num segundo passo, para cada objetivo, devemos elaborar uma lista das ações realizadas para alcançar esse objetivo.

Existem alguns métodos para realizar análise de tarefas utilizados em IHC, como a Análise Hierárquica de Tarefas, o GOMS e o ConcurTaskTrees. Essas técnicas serão detalhadas no próximo módulo deste tema.

## REQUISITOS EM IHC

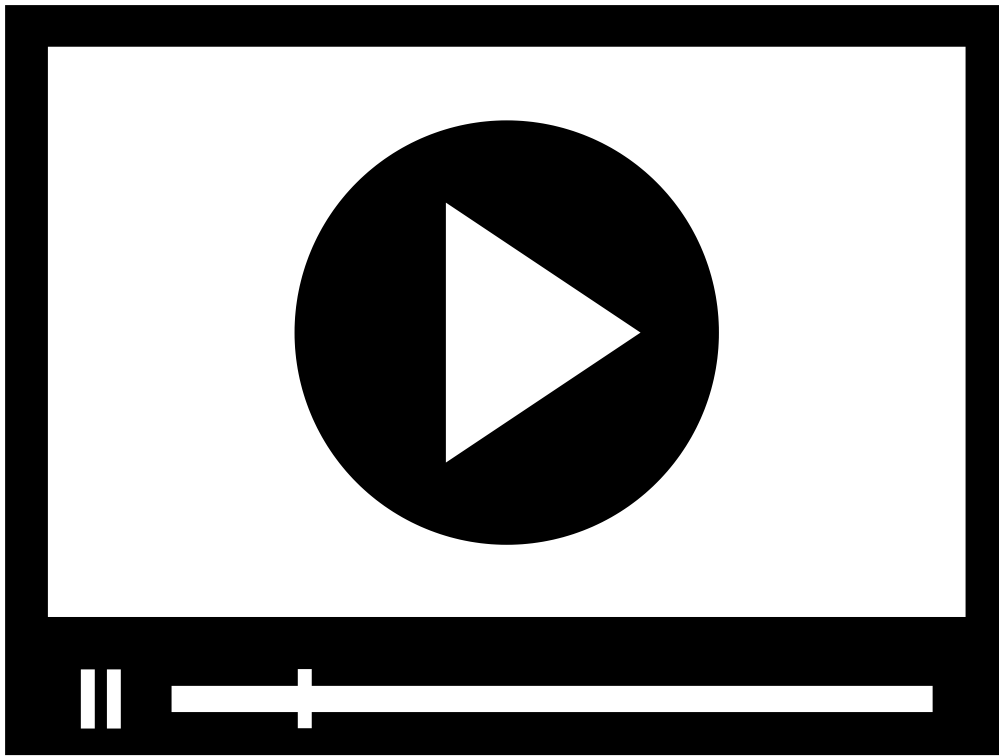
No processo de desenvolvimento de software, existe uma fase inicial chamada de levantamento de requisitos. É nessa fase que entendemos o funcionamento dos processos de trabalho dos usuários e suas necessidades para o sistema.

Nessa fase, o analista se reúne com os clientes e outros stakeholders com o objetivo de coletar informações sobre o que desejam para o sistema. O levantamento de requisitos é a fase em que se definem todas as funcionalidades que o sistema vai ter e, como é muito importante, costuma-se assinar um documento contendo todos os requisitos do software, como um acordo entre o usuário e o desenvolvedor.

### ATENÇÃO

O objetivo é alinhar as expectativas do cliente e de quem está desenvolvendo o software. Outro benefício de termos esse documento é o controle do projeto. Caso tenhamos alguma alteração, ele deve ser ajustado e acordado novamente.

Existem diversas técnicas para o levantamento de requisitos e, nestas, algumas práticas são bem interessantes para IHC. Vamos falar delas no próximo módulo.



## PROJETO DE DESIGN DE INTERFACE



## VERIFICANDO O APRENDIZADO

## MÓDULO 2

# TÉCNICAS DE CONCEPÇÃO E MODELAGEM DE INTERFACE HUMANO-COMPUTADOR

Agora falaremos sobre algumas técnicas usadas para identificação das necessidades do usuário dentro da concepção de IHC. A intenção dessas técnicas é tornar o projeto de interface humano-computador mais eficiente.



## TÉCNICAS DE LEVANTAMENTO DE REQUISITOS EM IHC

Na fase de levantamento de requisitos, é importante usarmos técnicas adequadas. Essas técnicas podem ser caracterizadas quanto ao seu objetivo, suas vantagens e o nível de esforço necessário para sua aplicação.

Dentre as técnicas mais utilizadas para levantar os requisitos dos usuários, podemos citar:

# ENTREVISTAS

É uma conversa guiada por um roteiro de perguntas, na qual um entrevistador busca obter informação de um entrevistado.

As perguntas podem ser **abertas** (exploratórias) ou **fechadas** (já se conhece as respostas prováveis).



As entrevistas podem ser **estruturadas** (roteiro seguido fielmente) ou **não estruturadas** (perguntas flexíveis).

Deve ser elaborado um roteiro previamente, que pode ser em formato de perguntas ou tópicos para lembrar os assuntos a serem vistos. Uma entrevista costuma seguir o seguinte roteiro:

Uma apresentação.

Perguntas de fácil resposta.

Perguntas mais extensas e complexas (parte principal).

Conclusão.

## GRUPOS DE FOCO

É um grupo com diversas pessoas reunidas numa espécie de entrevista coletiva, guiada por um moderador.

Costuma ser uma boa prática, pois fornece muitas informações em pouco tempo e envolve múltiplos pontos de vista. **O moderador é muito importante para garantir que todos participem e para evitar conflitos.**

# QUESTIONÁRIOS

É um formulário com perguntas que os usuários e demais participantes devem responder para fornecer os dados necessários em uma pesquisa.

Os questionários permitem coletar dados de muitas pessoas, até mesmo geograficamente dispersas. Podem conter perguntas abertas e fechadas, mas as perguntas fechadas são mais comuns, pois são de preenchimento rápido e de fácil análise.

As pessoas podem responder questionários a seu tempo e do seu local. Um dos inconvenientes é que o entrevistador não terá como tirar dúvidas sobre as perguntas. Além disso, devem conter instruções muito claras, dado que não haverá a presença do entrevistador para tirar dúvidas.

# BRAINSTORMING

É utilizado para coletar informações sobre as características que os usuários querem em um produto. Serve para qualquer produto. Seu resultado é uma lista priorizada de necessidades dos usuários.

Diferente de um grupo de foco, uma sessão de brainstorming busca levantar de forma bastante livre um conjunto grande e abrangente de opiniões dos participantes em torno de um tema. A sessão começa com uma pergunta resumo do objetivo e segue detalhando características do produto, de necessidades e desejos dos usuários, e também pode ser dividida em duas etapas.

Deve haver um moderador responsável por fazer perguntas e para esclarecer o que ficou definido no final. Para organizar as informações de um brainstorm, é recomendado o uso da técnica de **Diagrama de afinidade**. Essa técnica organiza ideias de acordo com seus relacionamentos naturais.

Seus passos são:

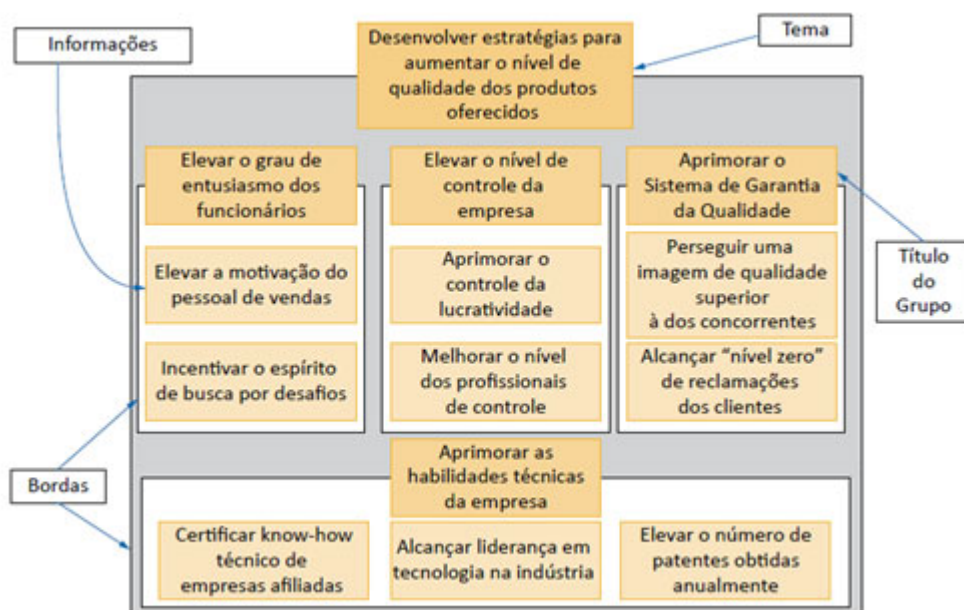
- Listar os dados coletados no brainstorm.

Coloque todos os dados em uma área que seja visível a todos.

Agrupe os dados, contendo características em comum.

Dê um nome para cada grupo.

Se precisar, agrupe os grupos em grupos maiores.



Exemplo de Diagrama de afinidade.

Fonte: SANTOS, 2016.

## CLASSIFICAÇÃO DE CARTÕES (CARD SORTING)

É utilizada para guiar o projeto da arquitetura de informação de um produto. Essa técnica pode ser utilizada para:

Explorar como as pessoas pensam sobre alguns pontos.

Descobrir categorias de informação para saber o que pode ser agrupado.

Coletar listas de palavras utilizadas no contexto.

É uma técnica que não depende de tecnologia. Consiste em escrever as categorias em papel e espalhá-las em uma área para visualmente fazer a classificação.



Exemplo de card sorting.

Fonte: SANTOS, 2016.

## ESTUDOS DE CAMPO

Inclui um conjunto de atividades relacionadas com usabilidade (investigação contextual, entrevistas no ambiente do usuário e observações simples).

Durante um estudo de campo, são visitados os ambientes onde ocorrem os processos em que o usuário participa. Trata-se de uma investigação da realidade dos usuários, e não de suposições. O objetivo é tornar explícitos os processos do ambiente do usuário.

## INVESTIGAÇÃO CONTEXTUAL

Tem como objetivo explicitar todos os aspectos da prática do trabalho do usuário.

A investigação contextual ocorre no local onde o usuário trabalha, de forma que se possa observar o usuário enquanto ele trabalha e conversar com ele sobre o seu trabalho.

# STORYBOARD

Técnica que detalha cenários do sistema por meio de uma sequência de desenhos. Os desenhos também podem ser feitos em papel e colocados em uma área visível aos outros membros das sessões de discussão.

Por meio dessa exposição, os desenhos podem ser avaliados e discutidos entre os usuários e designers e devem estar baseados em princípios de usabilidade.



Exemplo de storyboard.

Fonte: SANTOS, 2016.

## PROTÓTIPO/MAQUETE

É uma concretização do projeto de interface que permite aos usuários e envolvidos interagirem com ele e explorarem sua aderência às suas necessidades. É um modelo, uma representação do que pode ser o produto.

São usados para simular e testar interações e para demonstrar requisitos de layout das interfaces.

Para sua elaboração podemos seguir os seguintes passos:

### PASSO 1

Elaboração das estruturas gráficas das telas.



## PASSO 2

Definição das navegações entre as telas.

## PASSO 3

Alocação dos componentes nas telas.

## PASSO 4

Verificação de combinação de cores e outros elementos gráficos.

## PASSO 5

Verificação do atendimento de necessidades gerais do usuário quanto às interfaces.

Esse tipo de técnica pode ser usada em papel ou em softwares que são próprios para isso. Os protótipos podem ter baixa e alta fidelidade.

Um protótipo de **baixa** fidelidade é aquele que não se parece muito com o produto, mesmo assim é útil porque é simples, de rápida produção, barato e fácil de ser modificado para novos testes. Em geral, são os mais utilizados porque são descartáveis e servem mesmo só para acelerar o processo de levantamento de requisitos.



Já o protótipo de **alta** fidelidade utiliza ferramentas com linguagens mais rápidas de desenvolvimento, e com isso já mostra ao usuário como vai ficar o produto mais próximo da realidade. Este é útil e mais utilizado quando queremos vender ideias ou testar questões mais técnicas.



Protótipo de interface.

Fonte: Shutterstock.com

# TÉCNICAS DE MODELAGEM DE INTERFACE

Estas técnicas são um conjunto de etapas e atividades para a definição de elementos concretos, partindo de elementos abstratos.

As principais são: **Bridge** e **Design centrado no usuário**.

A **técnica de Bridge** trata da criação de tarefas de modelos de interação. É baseada numa sequência de sessões com várias pessoas envolvidas no projeto, com a intenção de criar “pontes” entre os requisitos dos usuários e da organização. Em geral, os usuários apresentam seu fluxo de trabalho, que são transformados em objetos de tarefas (caixas de diálogo e caixas de mensagens). Esses objetos, por sua vez, são testados pelos usuários participantes para verificar se atendem e correspondem às atividades que fazem atualmente e que serão transferidas para o sistema.

Já a técnica de **Design centrado no usuário** é uma abordagem **iterativa** que foca especificamente no uso do sistema. Essa técnica pressupõe que os designers irão prever como será o uso do sistema e com isso farão testes de validade desse uso.

# ITERATIVA

A iteração se refere à repetição, ciclos, em que compreende a execução destes ciclos ou das etapas em uma rotina maior.

Em geral, cada iteração nesta abordagem segue basicamente 4 etapas:

Especificação do contexto de uso onde são definidas as personas, os seus objetivos com o uso do produto e em que condições vão utilizá-lo.



Especificação dos requisitos que deverão ser atendidos pelo sistema.



Criação da solução de design da interface.



Avaliação quanto à qualidade através de testes de usabilidade.

## TÉCNICAS PARA ANÁLISE DE TAREFAS

No módulo anterior, citamos algumas técnicas para análise de tarefas utilizadas em IHC, dentre elas a **Análise Hierárquica de Tarefas** (HTA – Hierarchical Task Analysis), o **GOMS** (Goals, Operators, Methods, and Selection) e o **ConcurTaskTrees** (CTT).

Falaremos aqui um pouco mais sobre cada uma delas.

## ANÁLISE HIERÁRQUICA DE TAREFAS

Foi desenvolvida para entender as competências e habilidades necessárias em tarefas complexas e não repetitivas. Ajuda a relacionar o que as pessoas fazem, por que fazem, e quais as consequências caso não o façam corretamente. Uma tarefa é qualquer parte do trabalho que precisa ser realizada.

Em HTA, tarefa se aproxima do conceito de atividade.

A definição das tarefas começa pelo estabelecimento dos objetivos das pessoas. Um objetivo é um estado final que se quer atingir.

A HTA examina os objetivos de alto nível e os decompõe em subobjetivos, buscando identificar quais **subobjetivos** são mais difíceis de atingir (ou que geram mais erros).

Os subobjetivos de um objetivo e as relações entre eles são estabelecidos em um **plano**.

Um plano define os subobjetivos necessários para alcançar um outro **objetivo maior**.

No plano, cada subobjetivo é alcançado por uma **operação**, que é a unidade fundamental em HTA.

Uma operação é especificada pelas circunstâncias nas quais o objetivo é ativado (input ou entrada), pelas atividades ou ações (actions) que contribuem para atingi-lo e pelas condições que indicam o seu atingimento (feedback).

Uma ação pode ser entendida como uma instrução para fazer algo sob certas circunstâncias, o input como estados e o feedback como testes ou avaliação do estado final.

Dessa maneira, a análise visa a identificar principalmente como um sistema possibilita ou impede as pessoas de alcançarem seus objetivos. Essa análise permite ainda identificar problemas potenciais de cada ação, bem como elaborar recomendações para evitá-los.

Em geral, a Análise Hierárquica de Tarefas consiste nos seguintes passos:

Definir os objetivos.



Obter consenso.



Identificar as fontes de informação.



Criar uma tabela ou diagrama de decomposição dos objetivos.



Validar a decomposição com os envolvidos.



Identificar as operações a serem realizadas.



Testar o desempenho de uso do sistema.

## GOMS (GOALS, OPERATORS, METHODS, AND SELECTION RULES)

Esta técnica descreve uma tarefa e o conhecimento do usuário sobre como realizar essa tarefa em termos de **objetivos (goals)**, **operadores (operators)**, **métodos (methods)** e **regras de seleção (selection rules)**.

Os **objetivos** são as ações que o usuário quer realizar. Os operadores permitem que estas ações aconteçam como seleção de menus e o clique de um botão, por exemplo.

Os **métodos** são as sequências de subobjetivos e operadores que fazem com que o usuário atinja seus objetivos.

 **ATENÇÃO**

Quando há mais do que um método para atingir um mesmo objetivo, temos então as regras de seleção, que serão as tomadas de decisão dos usuários sobre qual método utilizar.

O **GOMS** trabalha com o conhecimento que uma pessoa tem sobre os processos que executa. É mais utilizado quando os usuários executam tarefas sobre as quais já têm bastante conhecimento.

## CTT (CONCURTASKTREES)

As árvores de tarefas concorrentes servem para apoiar a avaliação do design da interface.

O CCT trabalha com as:

Tarefas do **usuário** (realizadas fora do sistema).



Tarefas do **sistema** (sem interação com o usuário).



Tarefas **interativas** (com diálogos usuário-sistema).



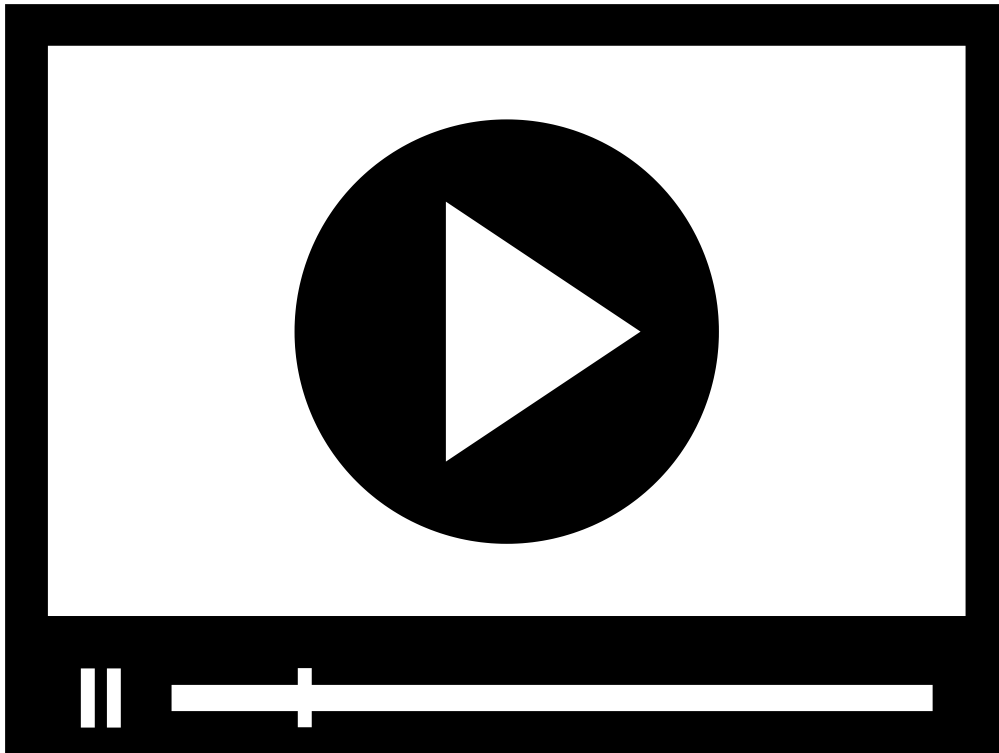
Tarefas **abstratas** (composição de tarefas).

A maior contribuição dessa técnica é explicitar a hierarquia entre as tarefas a serem executadas para atingimento dos objetivos dos usuários.

## ★ EXEMPLO

Para marcar a Tarefa A como realizada, as Tarefas B e C devem ter sido realizadas também. A vantagem dessa técnica é que nela há a possibilidade do registro explícito das relações entre

as tarefas.



## **TÉCNICAS DE CONCEPÇÃO E MODELAGEM DE INTERFACE HUMANO- COMPUTADOR**



## **VERIFICANDO O APRENDIZADO**

# MÓDULO 3

---

## 🕒 Reconhecer o processo de design de interface humano-computador **Processo de Design de Interface Humano-Computador**

Um sistema interativo tem o propósito de apoiar os usuários a alcançarem seus objetivos. Para isso, são desenvolvidos projetos de interface como vimos no módulo anterior.

Agora entenderemos quais são os processos que podem ser utilizados para o desenvolvimento desses projetos, mas antes aprenderemos um pouco mais sobre design de interface.

## DESIGN

Toda vez que inserimos um novo artefato em um contexto, estamos fazendo uma intervenção nesse contexto. Isso pode ter influências positivas ou negativas.

## ★ EXEMPLO

Exemplo: Sair de um apartamento alugado para um próprio tem muitos aspectos positivos – mais conforto, satisfação e facilidades –, mas também tem outras interferências não tão satisfatórias, como custos de impostos e taxas de condomínio, seguros, entre outros. Portanto, para qualquer interferência que venhamos a fazer, devemos analisar prós e contras. Assim é também o processo de design.

Quando analisamos um cenário, podemos encontrar problemas, características desagradáveis ou algo que podemos melhorar.

Nesse contexto, a atividade de design atua:

Na análise da situação atual.

Na síntese de uma intervenção para melhoria.

Na avaliação da nova situação para identificar as melhorias com a intervenção.



A análise da situação atual depende de diversos fatores, por exemplo, o domínio, os objetivos dos usuários, o tempo, os recursos, entre outros. Ela aponta as necessidades e as oportunidades de melhoria para as quais será projetada uma intervenção.

A diferença entre a situação atual e a situação desejada é a motivação principal para projetarmos uma intervenção. Por fim, esta deve ser analisada para verificar se surtiu os efeitos esperados.

## OS PROCESSOS DE DESIGN

Cada processo de design tem atividades diferentes, mas, em geral, todos definem: Como executar cada atividade; a sequência em que elas devem ser executadas; quais podem se repetir, e por quais motivos; e os artefatos usados e produzidos em cada uma delas.

Segundo Barbosa *et al.* (2004), uma característica básica dos processos de design de IHC é a execução das atividades de forma iterativa, permitindo refinamentos sucessivos da situação atual e da proposta de intervenção. Dessa forma, o designer pode aprender mais sobre o problema a ser resolvido e sobre a solução que irá definir.

Os processos de design de IHC começam analisando a situação atual. Quando se tem conhecimento suficiente sobre isso, ele vai para a fase seguinte, concebendo, modelando e construindo a intervenção. Enquanto projeta uma intervenção, pode precisar revisitar a situação atual. Assim, ele volta à atividade anterior para ampliar, refinar ou reformular sua interpretação, numa nova iteração.

Tendo uma proposta de intervenção em mãos, o designer passa para o processo de avaliá-la. Durante essa avaliação, este pode perceber que ainda precisa rever sua análise ou sua proposta de intervenção. Esse processo iterativo se repete quantas vezes forem necessárias, até o designer obter uma intervenção satisfatória.

### ATENÇÃO

É muito importante envolver os usuários durante todas essas atividades para que participem das decisões. Quanto mais cedo os usuários forem envolvidos no processo de design, mais

cedo será possível aprender sobre suas necessidades e, assim, gerar a solução, bem como identificar e corrigir problemas.

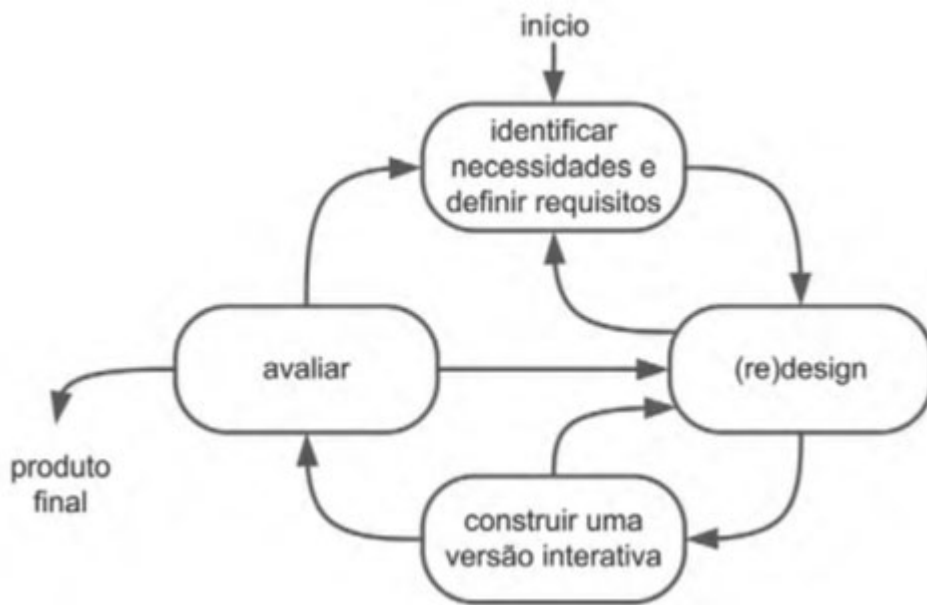
Os processos de design de IHC propostos na literatura são: Modelo de ciclo de vida simplificado, ciclo de vida estrela, Engenharia de Usabilidade de Nielsen, Engenharia de Usabilidade de Mayhew, design contextual, design baseado em cenários, design dirigido por objetivos, design centrado na comunicação.

## **MODELO DE CICLO DE VIDA SIMPLIFICADO**

Preece, Sharp e Rogers (2002, 2007) organizaram um processo simples, que mostra a importância do design centrado no usuário. Neste modelo, durante o (re)design da interação e da interface, o designer explora diferentes ideias para elaborar uma solução para as necessidades e aos requisitos definidos na atividade de análise.

O resultado dessa atividade de design pode ser registrado em descrições textuais da interação (cenários), esboços de interface (desenhos de tela) ou em qualquer modelo ou representação da interface e da interação usuário-sistema. O designer constrói versões interativas das propostas de solução que simulem o funcionamento da interface. A iteração entre as atividades ocorre quantas vezes for necessária, limitada apenas pelo orçamento, tempo e recursos disponíveis.

Idealmente, o processo de design é concluído com uma avaliação de que a solução de IHC atende às necessidades e aos requisitos identificados. O produto é uma especificação da interação e da interface. Uma ideia de como este processo funciona pode ser vista na figura a seguir.



Modelo simples de processo de design de IHC. Fonte: PREECE *et al.*, 2007.

## CICLO DE VIDA ESTRELA

É um processo composto por seis atividades, segundo Hix e Hartson (1993):

A análise de tarefas, usuários e funções é a atividade na qual fazemos o levantamento das necessidades dos usuários.

Na especificação de requisitos, entendemos os problemas que precisamos resolver.

A solução é concebida na atividade do projeto conceitual e especificação do design.

A prototipação é a atividade na qual são elaboradas as versões de interações possíveis.

Na atividade de implementação, a solução escolhida é finalmente desenvolvida.

A atividade de avaliação verifica se os dados coletados na atividade de análise e os requisitos especificados estão de acordo com as necessidades dos usuários e se não existem erros de usabilidade no sistema.

Nesse tipo de processo, o designer pode iniciar o trabalho pela atividade que preferir, dependendo da necessidade do projeto. A única exigência é que, após concluir cada uma delas, ele deve avaliar os resultados obtidos para verificar se está no caminho correto.

Como pode ser visto na figura a seguir, todas as atividades estão interligadas pela atividade de avaliação, ou seja, o que quer que se faça sempre necessita de uma avaliação, ao concluir uma atividade e antes de iniciar outra.



Ciclo de vida em estrela.

Fonte: BARBOSA et al., 2004.

# ENGENHARIA DE USABILIDADE DE NIELSEN

Segundo Nielsen (1993), é um conjunto de atividades que ocorrem durante todo o ciclo de vida do produto. Neste processo, o seguinte conjunto de atividades é proposto:

## CONHECER SEU USUÁRIO

Conhecer o usuário significa compreender suas necessidades e suas tarefas, assim como entender porque cada uma existe e se de fato fazem sentido.

## REALIZAR UMA ANÁLISE COMPETITIVA

A análise competitiva trata de examinar produtos prontos com funcionalidades semelhantes ou complementares à intenção. Além da comparação, identifica o que funciona e o que não funciona no ambiente.

## DEFINIR AS METAS DE USABILIDADE

A definição de metas entende que precisam ser definidos os fatores de qualidade de uso priorizados no projeto, assim como será avaliado e também qual a faixa de valor aceitável para cada um.

# **FAZER DESIGNS PARALELOS**

A parte de design paralelo consiste em elaborar diferentes alternativas de design para o projeto. Ao final dessa etapa, as soluções alternativas são analisadas e um design consolidado é elaborado.

## **ADOTAR O DESIGN PARTICIPATIVO**

No design participativo, são feitas discussões com os usuários envolvidos. Nessa fase está previsto que a equipe de design vá ter acesso permanente a um conjunto de usuários representativos.

## **FAZER O DESIGN COORDENADO DA INTERFACE**

A próxima fase é a de design coordenado de interface. Ela inclui não apenas os elementos de interface propriamente ditos, mas também toda a documentação, o sistema de ajuda e os tutoriais produzidos sobre o sistema.

## **APLICAR DIRETRIZES E ANÁLISE HEURÍSTICA**

Em seguida, vem a fase de análise heurística, nas quais são seguidas as diretrizes para o design das interfaces. A cada interface projetada deve ser feita uma avaliação heurística para verificar se as diretrizes estão sendo seguidas.

## **FAZER PROTÓTIPOS**

Antes de iniciar a implementação da interface com o usuário, recomenda-se fazer protótipos dos sistemas finais, para que sejam avaliados junto a usuários e modificados caso se tenha problemas.

## **REALIZAR TESTES EMPÍRICOS**

A partir dos protótipos, são feitos testes empíricos, que consistem principalmente na observação dos usuários ao utilizarem os protótipos para realizar certas tarefas.

## **PRATICAR DESIGN ITERATIVO**

Por fim, temos o design iterativo. A cada iteração de design e avaliação, alguns problemas são corrigidos e o processo deve ser repetido até o alcance das metas.

# ENGENHARIA DE USABILIDADE DE MAYHEW

Outro ciclo de vida para a Engenharia de Usabilidade foi proposto por Deborah Mayhew (1999). Ele reúne e organiza diferentes atividades em IHC.

Análise de requisitos

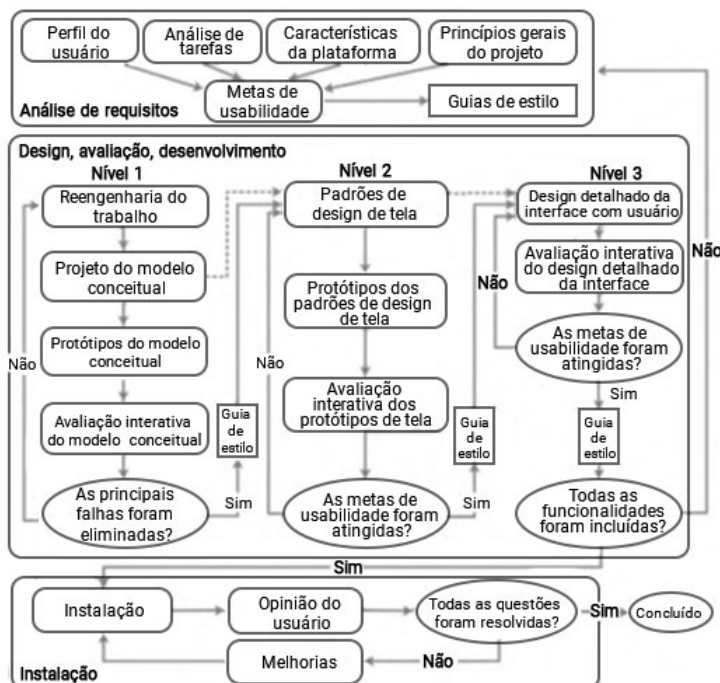
Na fase de análise de requisitos são definidas as metas de usabilidade dos usuários.

Design/avaliação/desenvolvimento

Na fase de design, avaliação e desenvolvimento é concebida a solução de IHC que atenda às metas de usabilidade estabelecidas na fase anterior

Instalação

Na fase de instalação, o designer deve coletar opiniões dos usuários depois de algum tempo de uso. A intenção é melhorar o sistema em versões futuras.



Ciclo de vida para a engenharia de usabilidade.

Fonte: MAYHEW, 1999.

# DESIGN CONTEXTUAL

O objetivo deste processo é compreender as necessidades dos usuários através de uma investigação detalhada do contexto de uso. Isso é fundamental para o designer elaborar uma solução adequada.

As principais atividades do design contextual são:

## INVESTIGAÇÃO CONTEXTUAL

Na qual se busca conhecer os usuários e sua forma de trabalho.

## MODELAGEM DO TRABALHO

Na qual são registrados e compartilhados com a equipe o que se aprendeu na investigação contextual.

## CONSOLIDAÇÃO

Na qual são organizados os perfis, reprojeto do trabalho caso necessário, projeto do ambiente do usuário onde se tem as informações de contexto, prototipação e teste com usuários.

## DESIGN BASEADO EM CENÁRIOS

Este processo utiliza diferentes tipos de cenários durante todas as atividades envolvidas na criação da solução.

Um cenário, como já vimos anteriormente, é uma história sobre pessoas executando uma atividade. Ao escrever, ler e revisar cenários, a equipe de design e os usuários têm a

oportunidade de discutir e analisar como as atividades dos usuários são afetadas pelo sistema que será desenvolvido. Assim, podemos dizer que os cenários são uma ferramenta útil e barata para gerar e avaliar diversas ideias durante as atividades de design.

Nesse tipo de processo de desenvolvimento de um projeto de IHC, a equipe de design explora ideias para a solução elaborando três tipos de cenários:

De atividade

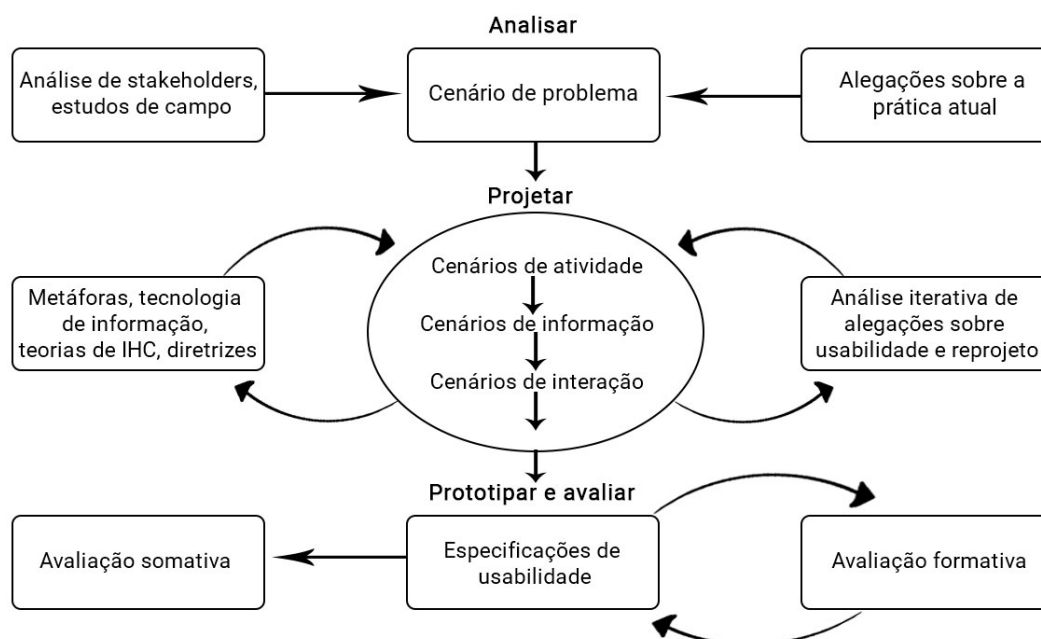
De informação

De interação

Um cenário de atividade é uma narrativa sobre as tarefas típicas e críticas que os usuários vão executar com ajuda do sistema.

Um cenário de informação é uma elaboração de um cenário de atividade que descreve as informações fornecidas pelo sistema ao usuário durante a interação.

Um cenário de interação especifica em detalhes as ações do usuário e as respectivas respostas (feedback) do sistema necessárias para executar as tarefas apoiadas pelo sistema.



Atividades do design baseado em cenários.

Fonte: BARBOSA *et al.*, 2004.



# DESIGN DIRIGIDO POR OBJETIVOS

É um processo que incentiva o designer a explorar as tecnologias para oferecer aos usuários soluções mais criativas, inovadoras e eficientes de alcançarem seus objetivos.

É dividido em seis fases:

## Pesquisa

O designer, em geral, está interessado em conhecer o usuário, o domínio do sistema e o contexto de uso.

## Modelagem

Tem por objetivo organizar e registrar o conhecimento adquirido na fase de pesquisa através da elaboração de modelos do usuário, domínio e contexto de uso.

## Definição de requisitos

O designer interpreta as informações coletadas e estruturadas nos modelos para definir os requisitos do usuário, do negócio e técnicos.

## Projeto conceitual

O designer concebe uma solução de interação e um esboço de interface pouco detalhado; sua preocupação está na concepção da estrutura e no comportamento da interface.

## Refinamento

O foco é detalhar a solução de interface, definindo todas as características dos elementos da mesma, tais como tamanho, cores e ícones.

## Coerência

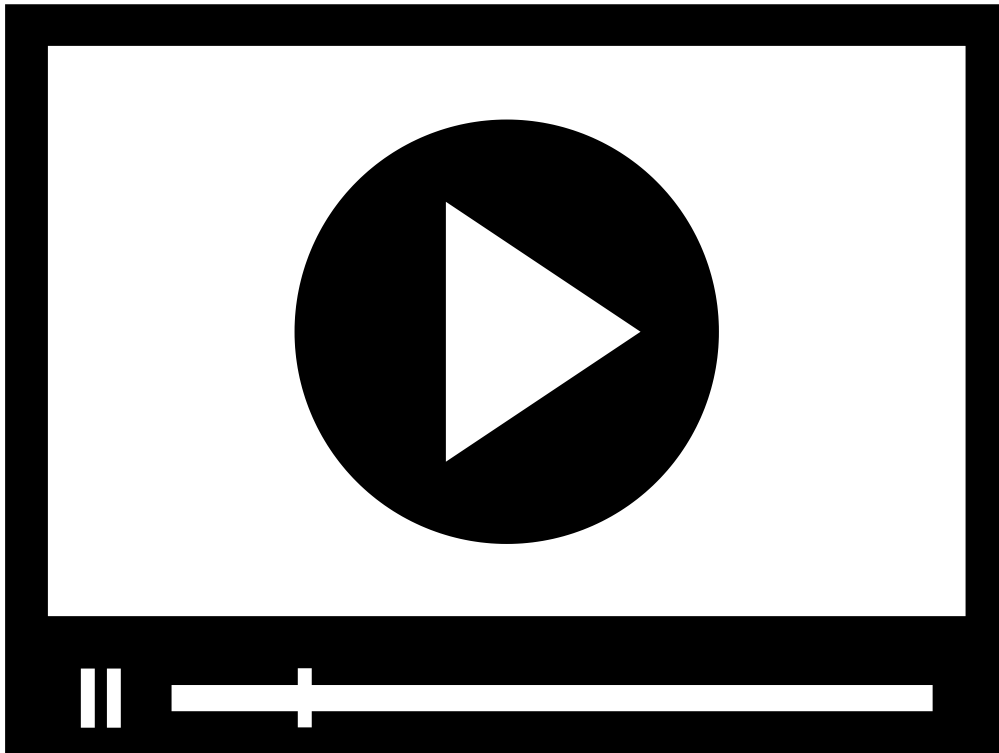
Manter a coerência da solução proposta enquanto acomoda as limitações técnicas imprevistas.

# DESIGN CENTRADO NA COMUNICAÇÃO

O design centrado na comunicação, segundo Barbosa *et al.* (2004), é baseado na Engenharia Semiótica. Entende a interação humano-computador como um processo de comunicação entre o usuário e o designer do sistema, através da sua interface.

Quando o usuário tem acesso a essa metacomunicação, acredita-se que ele tenha melhores condições de aprender e usar o sistema de forma produtiva, eficiente e criativa.

Seu principal objetivo é elaborar uma solução de IHC que transmita a metacomunicação do designer para se construir um sistema com alta comunicabilidade. Para isso, o designer é orientado a se posicionar como um dos interlocutores das conversas que ocorrem durante a interação.



## PROCESSO DE DESIGN DE INTERFACE HUMANO COMPUTADOR



# VERIFICANDO O APRENDIZADO

## MÓDULO 4

---

- ⦿ Definir princípios e diretrizes para o design de interfaces humano-computador

## PRINCÍPIOS E DIRETRIZES DE DESIGN DE INTERFACE HUMANO-COMPUTADOR

Existem diversos princípios, diretrizes e heurísticas na área de IHC. Os mais conhecidos são os de Norman (1988) e de Nielsen (1993) e as regras de ouro de Shneiderman (1998). A existência deles não substitui os processos de análise, design e avaliação. De toda forma, esses princípios e diretrizes podem ser usados como apoio aos processos de design.

Algumas das diretrizes são desenvolvidas especificamente para determinados ambientes ou dispositivos. Alguns servem muitas vezes para definir padrões ou assegurar um determinado comportamento, mesmo sendo genéricas e não contextualizadas. Cabe ao designer considerar se e quais diretrizes são adequadas à sua situação de design, e como elas devem se manifestar na solução.

Os princípios e as diretrizes mais utilizados falam dos seguintes temas:

Correspondência com as expectativas dos usuários.

Simplicidade nas estruturas das tarefas.

Equilíbrio entre controle e liberdade do usuário.

Consistência e padronização.

Promoção da eficiência do usuário.

Antecipação das necessidades do usuário.

Visibilidade e reconhecimento.

Conteúdo relevante e expressão adequada.

Projeto para erros.

Sendo assim, abordaremos em seguida alguns dos principais pontos de extrema importância no tocante à qualidade de interface.

## **CORRESPONDÊNCIA COM AS EXPECTATIVAS DOS USUÁRIOS**

Esta diretriz trata de verificar se os usuários conseguem estabelecer relações principalmente entre ações e seus efeitos no sistema. Por exemplo, ao projetar um sistema de comércio eletrônico, devemos examinar como as pessoas fazem suas compras em lojas físicas. Precisamos fazer com o que o sistema permita ao usuário executar todas as ações que executa na loja.

### **★ EXEMPLO**

Em uma loja física, não nos identificamos ao entrar na loja e olhar os produtos, somente fazemos isso quando decidimos comprar e precisamos pagar. Então, um sistema que implementa um processo de comprar não deveria exigir que o usuário se identificasse para entrar.

O designer também deve projetar a interface utilizando o idioma do usuário, com palavras, expressões e conceitos que lhe são familiares. Também é recomendado o uso de metáforas,

que criam imagens na mente.

# **SIMPLICIDADE NAS ESTRUTURAS DAS TAREFAS**

Outra recomendação muito interessante é simplificar a estrutura das tarefas.

Para isso, os designers devem seguir quatro abordagens:

## **ABORDAGEM 01**

manter a tarefa, mas fornecendo diversas formas de apoio para que os usuários consigam aprender e realizar a mesma.

## **ABORDAGEM 02**

usar tecnologia para tornar visível o que seria invisível, melhorando o feedback e a capacidade de o usuário se manter no controle da tarefa.

## **ABORDAGEM 03**

automatizar a tarefa ou parte dela, mantendo-a igual.

## **ABORDAGEM 04**

modificar a natureza da tarefa.

# **EQUILÍBRIO ENTRE CONTROLE E LIBERDADE DO USUÁRIO**

O usuário deve estar no controle. Para isso, é necessário tentar reduzir o número de opções ou decisões que ele precisa tomar a cada instante.

Os usuários não devem ficar presos em um caminho de interação único para realizar uma atividade. Sempre deve ser fornecido a eles um caminho alternativo.

Mas é importante ressaltar que o usuário pode ter mais ou menos liberdade de acordo com seu perfil. Usuários sem muita experiência podem precisar de mais assistência e menos

alternativas, enquanto os mais experientes podem ter mais alternativas.

Shneiderman (1998) recomenda permitir que o usuário tenha controle total da interação.

Também é importante que eles possam cancelar ou desfazer suas ações, pois isso reduz o medo de errar e evita a necessidade de apresentar diversos diálogos pedindo confirmação das ações dos usuários.

## ATENÇÃO

Usar diálogos de confirmação em excesso e de forma indiscriminada não apenas aumenta o tempo de realização das tarefas, mas também pode tornar a comunicação ineficiente, pois muitos usuários acabam prosseguindo a interação sem mesmo ler o conteúdo desses diálogos.

Para aumentar o controle do usuário, o sistema também deve permitir que trabalhem de modo flexível, podendo o usuário configurar o sistema.

# PROMOÇÃO DA EFICIÊNCIA DO USUÁRIO

Uma diretriz importante e bastante recomendada é considerar a eficiência do usuário antes da eficiência do sistema.

Nessa diretriz, **recomenda-se manter o usuário ocupado, porque sempre há perda de produtividade quando ele precisa esperar.**

Sendo assim, é primordial que o projeto de software entenda que o sistema deve ser sensível às ações do usuário e deve interrompê-lo o mínimo possível. Um bom exemplo disso é que precisamos prever mecanismos para que o usuário nunca perca seu trabalho em caso de erro, seja este por uma falha na transmissão de rede, por um problema no fornecimento de energia para o computador ou por qualquer outra razão.

Devemos ter também mecanismos que permitam ao sistema “lembrar” de todas as ações do usuário antes do problema ocorrer. O sistema deve ser capaz de saber: se esse é o primeiro acesso; onde o usuário está no sistema; para onde ele está indo; o que o usuário tem feito durante a sessão de uso atual; onde ele estava quando deixou o sistema na última sessão; e outras informações semelhantes que permitam diminuir o trabalho do usuário e melhorar sua experiência de uso do sistema.

## COMENTÁRIO

A ideia principal é fornecer atalhos e aceleradores. Teclas de atalho e comandos ocultos são úteis a usuários experientes, e não prejudicam a interação dos usuários novatos.

## ANTECIPAÇÃO DAS NECESSIDADES DO USUÁRIO

A intenção desta diretriz é fazer com que o designer pense em como prever o que os usuários querem e precisam, em vez de esperar que estes busquem ou coletem informações.

Para isso, o sistema deve fornecer ao usuário todas as informações e ferramentas necessárias para cada passo do processo.

O software deve, portanto, tomar iniciativa e fornecer informações adicionais úteis – que tiverem relação com a pergunta –, em vez de apenas responder precisamente a uma pergunta do usuário.

Além disso, o software deve ser observador e se lembrar de quais ações o usuário realiza em sequência, para tentar antever o próximo passo a cada momento.

## COMENTÁRIO

Essa diretriz também trata, por exemplo, do preenchimento de campos automaticamente. É uma boa prática de design para campos que tenham valores default que estes já venham preenchidos.

## VISIBILIDADE E RECONHECIMENTO

Nesta diretriz, são tratadas as ações que devem estar explícitas para o usuário. Antes de executar uma ação, é necessário tornar visível para os usuários o que é possível realizar e

como as ações devem ser feitas.

Nesse sentido, a interface não deve oferecer opções que não estejam disponíveis ou não façam sentido em um determinado momento da interação.

Outro ponto importante é que, depois de uma ação realizada, o sistema deve mostrar ao usuário o novo estado, reconhecendo que a ação foi realizada e que a situação mudou.

O sistema não deve exigir que o usuário memorize muitas informações ou comandos durante a interação, devido à limitação humana do processamento de informação na memória de curto prazo.

## RECOMENDAÇÃO

Muitas vezes, o usuário pode não perceber a resposta do sistema, tamanha sua sutileza. Recomenda-se que comandos mais incisivos sejam aplicados. É necessário, ainda, que o sistema mantenha claro para o usuário o caminho que ele percorreu. Mapas mentais de caminhos não devem ser de responsabilidade do usuário.

# CONTEÚDO RELEVANTE E EXPRESSÃO ADEQUADA

Esta diretriz segue a ideia de que “menos é mais”. Isso está fortemente relacionado à simplicidade da interface.

A máxima da relação ou relevância afirma que **tudo o que for dito deve ter relação clara com os tópicos da conversa.**

A intenção é evitar a prolixidade e ambiguidade, buscar a concisão e ordenar adequadamente a conversa. Por isso, todos os diálogos devem ter somente as informações necessárias.

## RECOMENDAÇÃO



Recomenda-se que as mensagens de instrução sejam concisas e informativas e que rótulos e menus sejam claros e livres de ambiguidade.

Além do conteúdo, o designer deve se certificar de que o texto também seja legível.

Os **designers gráficos são os principais responsáveis pela identidade visual do sistema**, incluindo o layout – que define a disposição espacial dos elementos de interface – e a escolha de fontes, formas, cores, texturas, imagens e outros símbolos.

Alguns pontos importantes apontados por essa diretriz são: **Elegância e simplicidade; escala, contraste e proporção; organização e visual; imagens, entre outros.**

## INTEGRAÇÃO ENTRE IHC E ENGENHARIA DE SOFTWARE

As áreas de IHC e ES possuem, em alguns pontos, diferentes perspectivas sobre o que é importante em um sistema interativo ou sobre como desenvolvê-lo.

Para a Engenharia de Software, um sistema interativo é um artefato que tem uma interface que recebe dados, processa esses dados e retorna dados de saída. O que mais importa é aquilo que ocorre dentro do sistema. Tudo que ocorre na fronteira ou fora dele, inclusive a própria interface, acaba recebendo pouca atenção.



Na perspectiva de IHC, o objetivo principal é construir um sistema fidedigno que seja capaz de processar adequadamente os dados de entrada e de saída transmitidos através de uma interface bem definida.

As diferentes perspectivas de IHC e ES sobre o desenvolvimento de sistemas interativos deram origem a métodos, técnicas e processos próprios de cada área.

Atualmente, pesquisadores têm investigado a integração de métodos e técnicas entre IHC e ES. As principais abordagens de integração são:

Definição de características de um processo de desenvolvimento que se preocupa com a qualidade de uso.

Definição de processos de IHC paralelos que devem ser incorporados aos processos propostos pela ES.

Indicação de pontos em processos propostos pela ES nos quais atividades e métodos de IHC podem ser inseridos.

## MÉTODOS ÁGEIS EM IHC

Métodos ágeis de desenvolvimento, como **eXtreme Programming** e **Scrum**, podem ser interessantes em IHC porque buscam colaborar com o cliente através de pequenos ciclos de desenvolvimento de forma iterativa e incremental, para corrigir o rumo do processo. Porém, raramente as comunidades de métodos ágeis mencionam os usuários ou interface com usuário como um todo, muitas vezes negligenciando a experiência de uso.

## EXTREME PROGRAMMING

Programação extrema (do inglês eXtreme Programming), ou simplesmente XP, é considerada uma metodologia ágil, pois se ajusta bem a pequenas e médias em desenvolvimento de software com requisitos vagos e em constante mudança. Para isso, adota a estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de software.

Fonte: Wikipédia

## SCRUM

O Scrum (pron. [skrʌm]) é um framework de gerenciamento de projetos, da organização ao desenvolvimento ágil de produtos complexos e adaptativos com o mais alto valor possível, através de várias técnicas, usado desde o início de 1990 e que atualmente é utilizado em mais de 60% dos projetos ágeis em todo o mundo.

Este framework não é um processo linear, em vez disso, é um conjunto de conceitos e técnicas dentro do qual você pode empregar vários outros processos ou técnicas.

Fonte: Wikipédia

Algumas sugestões que podem integrar IHC em métodos ágeis:

O designer de IHC é responsável pelas decisões relacionadas com a qualidade de uso.

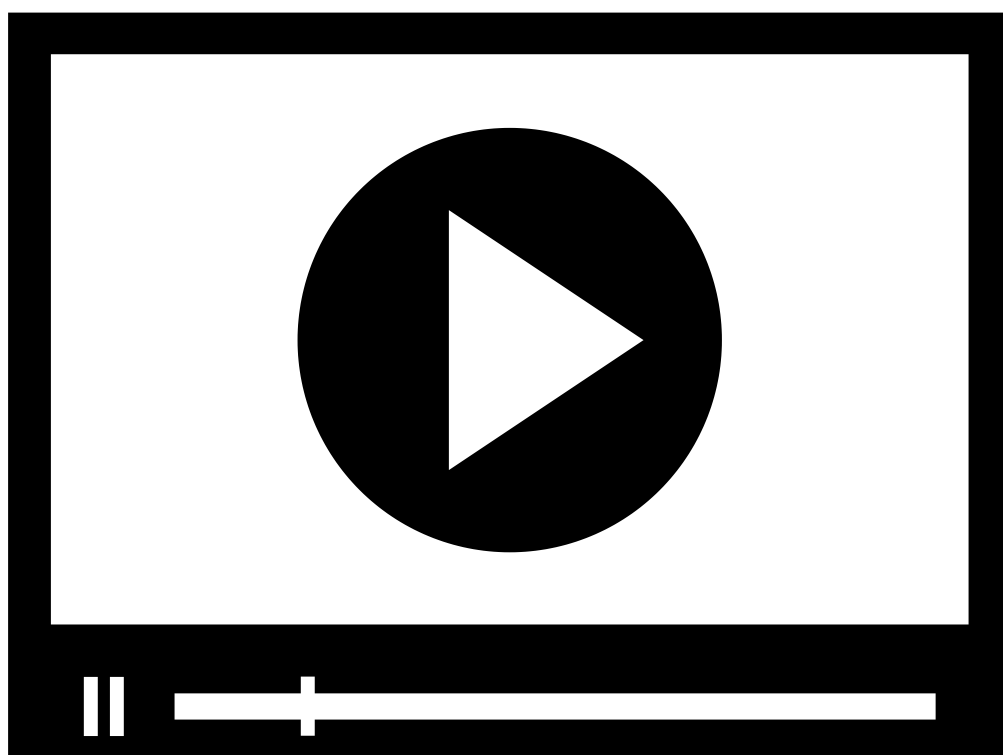
Deve-se equilibrar no cronograma do projeto o tempo para entregar um sistema e o tempo gasto para se ter qualidade de uso dele.

Buscar informações sobre o contexto de uso, e não apenas consultar os usuários e clientes sobre seus requisitos.

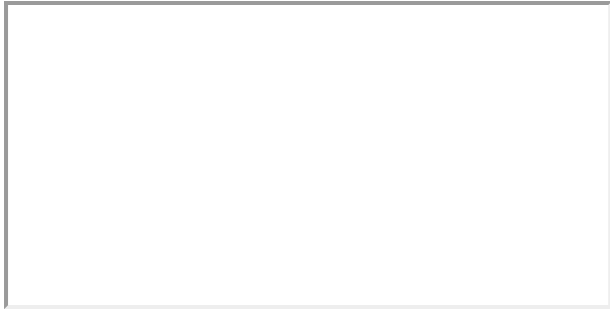
Realizar uma análise da situação atual do contexto de uso e não só dos casos de uso (use cases) amplamente utilizados em métodos ágeis.

O designer de IHC deve ser consultado na priorização das funcionalidades que serão desenvolvidas.

Realizar avaliações de IHC durante diferentes estágios do ciclo de desenvolvimento.



# PRINCÍPIOS E DIRETRIZES PARA DESIGN DE INTERFACE HUMANO-COMPUTADOR



## VERIFICANDO O APRENDIZADO

## CONCLUSÃO

## CONSIDERAÇÕES FINAIS

Neste tema, aprendemos como desenvolver um projeto de interface humano-computador. No primeiro módulo, apresentamos o conceito de affordance e comunicabilidade e como estes podem ser aplicados no projeto, bem como as fases deste e os produtos que são gerados em cada uma delas. Além disso, vimos os tipos de representação com perfil de usuário, personas, cenários e tarefas e, por fim, o conceito de requisitos de forma mais geral.

No segundo módulo, visitamos técnicas de concepção e modelagem. Entendemos como realizar uma entrevista, montar um grupo focal, fazer questionários, realizar classificação de cartões, elaborar um storyboard, fazer reuniões de brainstorm, entre outras técnicas, e em que situação de levantamento de requisitos de design cada uma é mais interessante de ser aplicada.

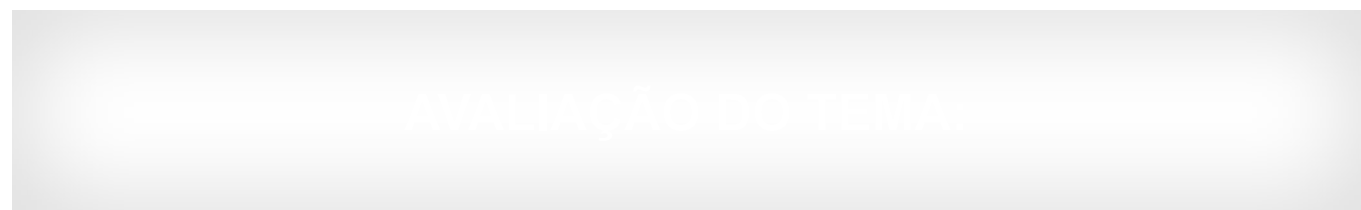
Em seguida, apresentamos os tipos de processo de design de interface e as fases de cada um destes. Falamos principalmente dos tipos de ciclo de vida simplificado e estrela, das engenharias de usabilidade de Nielsen e Mayhew e dos tipos de design dirigido por objetivos, contextual, baseado em cenários e centrado na comunicabilidade.

Por fim, conhecemos alguns dos principais princípios e diretrizes para o design, que são qualificados como boas práticas para modelagem de uma interface.



PODCAST

## ! PODCAST



## REFERÊNCIAS

BARBOSA, S. D. J.; SILVA, B. S. **Interação Humano-Computador**. Rio de Janeiro: Campus - Elsevier, 2010.

CARVALHO, C. R. M. *et al.* **Unindo IHC e negócios através do uso de personas**: Um estudo de caso no mercado de aplicativos móveis. *In*: Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction. Brazilian Computer Society, pp. 100-104, 2011.

HACKOS, J. T.; REDISH, J. C. **User and task analysis for interface design**. New York: John Wiley & Sons, 1998.

HIX, D.; HARTSON, H. **Developing user interfaces**: ensuring usability through product and process. New York: John Wiley & Sons, 1993.

MAYHEW, D. **The usability engineering lifecycle**: A practitioner's handbook for user interface design. San Francisco: Morgan Kaufmann, 1999.

NORMAN, D. A. **The design of everyday things**. New York: Basic Books, 1998.

PRATES, R. O.; BARBOSA, S. D. J.; SOUZA, C. S. de. **A case study for evaluating interface design through communicability**. Proceedings of the ACM International Conference on Designing Interactive Systems, DIS 2000, pp. 308–317, 2000.

PRATES, R. O.; BARBOSA, S. D. J. **Introdução à teoria e prática da Interação Humano-Computador fundamentada na Engenharia Semiótica**. *In*: T. Kowaltowski & K. Breitman (orgs.). Atualizações em informática 2007. XXVII Congresso da Sociedade Brasileira de Computação, JAI/SBC, 2007.

PREECE, J.; SHARP, H.; ROGERS, Y. **Interaction design**: Beyond human-computer interaction. New York: John Wiley & Sons, 2002.

PREECE, J.; SHARP, H.; ROGERS, Y. **Interaction design**: Beyond human-computer interaction. New York: John Wiley & Sons, 2007.

ROGERS, Y.; SHARP, H.; PREECE, J. **Design de interação**. Porto Alegre: Bookman, 2013.

SANTOS, F. G. **Engenharia de usabilidade**. 1. ed. Rio de Janeiro: Estácio, 2016.

SHNEIDERMAN, B. **Designing the user interface**: Strategies for effective Human-Computer Interaction. 3. ed. Boston: Addison-Wesley, 1998

---

## EXPLORE+

Para saber mais sobre os assuntos tratados neste tema, leia:

Recursos para um bom design, elaborado pela Professora Lucia Vilela Leite Filgueiras.

Fatores impactantes nos projetos de interface com o usuário de software de dispositivos móveis, da UNIP.

Pesquise na Internet:

Os slides da Universidade Federal do Piauí, disponibilizados na plataforma slide share, sobre o conceito de design e seus processos.

O site do Professor Walter de Abreu Cybis no LabUtil da UFSC (Universidade Federal de Santa Catarina).

---

# CONTEUDISTA

Claudia Cappelli

 **CURRÍCULO LATTES**