

# Design, Prototipação e Construção

Prof<sup>a</sup>. Juliana Pinheiro Campos  
E-mail: [jupcampos@gmail.com](mailto:jupcampos@gmail.com)

# Design

- A atividade de design inicia uma vez que o conjunto de requisitos tenha sido estabelecido.
- Existem dois tipos de design:
  - Conceitual: preocupa-se em desenvolver um modelo conceitual que capte o que o produto irá realizar e como irá se comportar.
  - Físico: apresenta detalhes do design, tal como tela, estruturas de menus, ícones e gráficos.
- O design evolui iterativamente em ciclos de design – avaliação – redesign.

# Design

- Para avaliar de modo eficaz o design de um produto interativo, os designers devem produzir uma versão interativa de suas idéias.
- Essas podem ser feitas com papel e cartolina (nos primeiros estágios de desenvolvimento) ; conforme o design progride, elas podem vir a ser partes (polidas) do software ou peças de metal e plástico que já se parecem com o produto final.
- Essa atividade, preocupada com construir essa versão interativa, é chamada de prototipação e construção.

# Prototipação e construção

- Os usuários não conseguem transmitir o que querem, mas se vêem algo e começam a utilizá-lo sabem o que não querem.
- Após coletar requisitos temos que experimentar nossas idéias construindo protótipos e experimentando iterativamente várias versões.
- Quanto mais iterações melhor será o produto final.

# Protótipo

- Pode ser:
  - um modelo em escala menor de prédio ou ponte;
  - Esboço de papel de uma tela;
  - Maquete tridimensional, de papel ou cartolina;
  - Simples “pilha” de telas vinculadas por hyperlinks;
  - Entre outros.

# Protótipo

- Um protótipo pode ser qualquer coisa desde um storyboard de papel a uma parte complexa de um software e de uma maquete de cartolina a um pedaço de metal moldado e prensado.
- Ele possibilita que stakeholders interajam com um produto imaginado visando adquirir alguma experiência de como utilizá-lo em um ambiente real e a explorar os usos para ele imaginados.
- É uma representação limitada de um design que permite aos usuários interagir com ele e explorar sua conveniência.



# Por que fazer protótipos?

- São úteis quando se estão discutindo idéias com stakeholders;
- Facilitam a comunicação entre os membros das equipes e também entre os desenvolvedores e stakeholders;
- É uma forma eficaz de testar as idéias para você mesmo.
- Eles respondem a questões e fornecem suporte aos designers para a escolha de uma dentre várias opções.

# Por que fazer protótipos?

- Logo, protótipos servem para vários fins:
  - Testar a viabilidade técnica de uma idéia;
  - Esclarecer requisitos vagos;
  - Realizar alguns testes com usuários e avaliações;
  - Verificar se um certo rumo que se tomou no design é compatível com o resto do desenvolvimento do sistema.
- Ter um desses fins como seu propósito irá influenciar o tipo de protótipo que você irá construir.



# Por que fazer protótipos?

- Ex: Se você está tentando esclarecer como os usuários podem realizar um conjunto de tarefas e se o aparelho que você propôs oferece suporte a eles nisso, você pode então fazer uma maquete de papel.
- Tipos de prototipação:
  - Prototipação de baixa-fidelidade;
  - Prototipação de média-fidelidade;
  - Prototipação de alta-fidelidade;

# Prototipação de baixa fidelidade

- É aquele que não se assemelha muito com o produto final.
- Ele utiliza, por exemplo, materiais muito diferentes da versão final pretendida. Ex: impressora feita de caixa de papelão.
- Vantagens: simples, baratos e de rápida produção.
- Por ser de rápida produção, também podem ser rapidamente modificados, dando suporte à exploração de designs e idéias alternativas.

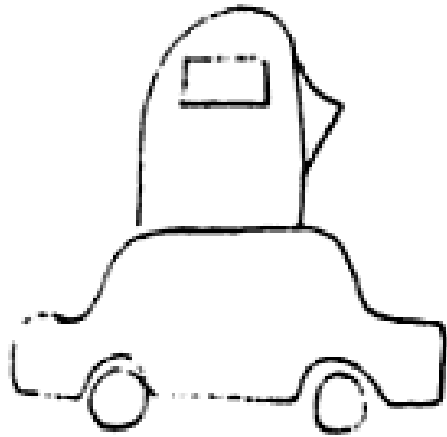
# Prototipação de baixa fidelidade

- Isso é importante nos primeiros estágios de desenvolvimento (durante design conceitual), uma vez que os protótipos usados para explorar idéias devem ser flexíveis e encorajar a exploração e modificação.
- Eles nunca são projetados para ser mantidos e integrados ao produto final; servem apenas para exploração.

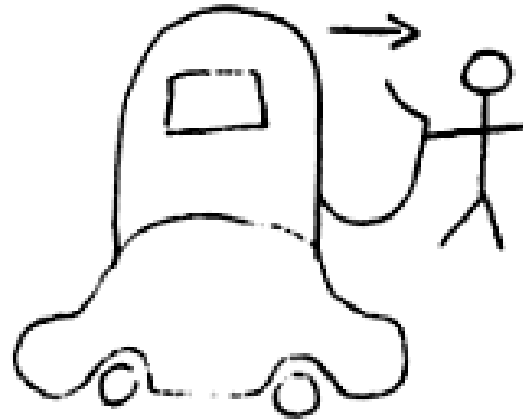
# Prototipação de baixa fidelidade

- **Storyboards:** Consiste em uma série de desenhos mostrando como o usuário pode progredir em uma tarefa utilizando o produto que está sendo desenvolvido.
- **Esboços:** A prototipação de baixa fidelidade quase sempre se baseia na idéia de esboços. As pessoas tem dificuldade pois se sentem inibidas devido à qualidade de seus desenhos. Você pode desenvolver seus próprios ícones para elementos que queira esboçar e praticar.

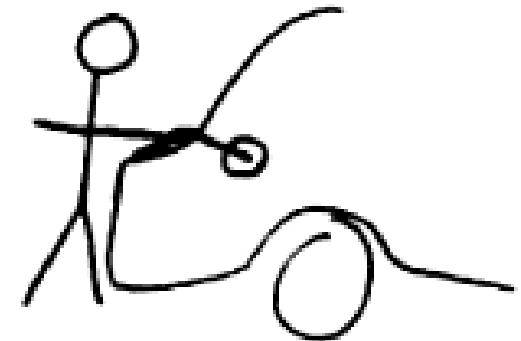
# Prototipação de baixa fidelidade



Conduza o carro até a bomba de gasolina



Retire a mangueira da bomba...



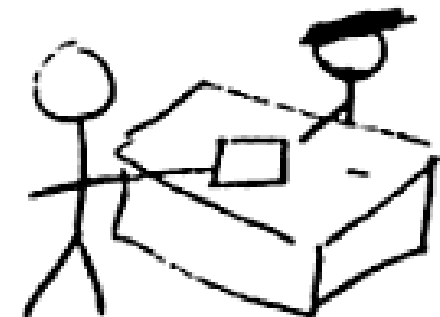
...e insira-a no tanque



Aperte o gatilho da mangueira até o tanque ficar cheio



Recoloque a mangueira na bomba quando o tanque estiver cheio



Pague no caixa

# Prototipação de alta fidelidade

- Utiliza materiais que você espera que estejam no produto final e realiza um protótipo que se parece muito mais com algo acabado.
- São representações executáveis (código), construídos com o uso de alguma linguagem de programação (ou ferramenta de apoio) e contém as principais funcionalidades presentes na interface do futuro sistema.
- Ex: Um protótipo de software desenvolvido em linguagem Java apresenta maior fidelidade do que o de uma maquete de papel.

# Prototipação de alta fidelidade

- Problemas de projetos realizados mediante prototipação de alta fidelidade:
  - Levam muito tempo para ser construídos;
  - Os revisores e aplicadores de teste tendem a comentar aspectos superficiais, em vez do conteúdo;
  - Os desenvolvedores relutam em mudar algo no qual trabalharam artesanalmente por horas;
  - Um protótipo em software pode elevar demais as expectativas.
  - É necessário apenas um bug no protótipo para interromper o teste.



# Prototipação de alta fidelidade

- É útil para vender idéias a pessoas e para testar questões técnicas.
- Oferece ao usuário interação direta com a solução proposta.
- No entanto, o uso da prototipação em papel e outras idéias deveriam ser encorajadas para a exploração de questões de conteúdo e estrutura.

# Alta fidelidade x Baixa fidelidade

**Tabela 8.1** Eficácia relativa de protótipos de baixa- vs. alta-fidelidade (Rudd *et al.*, 1996)

Tipo	Vantagens	Desvantagens
<b>Protótipo de baixa-fidelidade</b>	<ul style="list-style-type: none"><li>• Custo mais baixo de desenvolvimento.</li><li>• Avalia múltiplos conceitos de <i>design</i>.</li><li>• Instrumento de comunicação útil.</li><li>• Aborda questões de leiaute de tela.</li><li>• Útil para identificação de requisitos de mercado.</li><li>• <i>Proof-of-concept</i> (demonstrações de que o conceito funciona).</li></ul>	<ul style="list-style-type: none"><li>• Verificação limitada de erros.</li><li>• Especificação pobre em detalhe para codificação.</li><li>• “Uso” conduzido pelo facilitador.</li><li>• Utilidade limitada após estabelecimento dos requisitos.</li><li>• Utilidade limitada para testes de usabilidade.</li><li>• Limitações de fluxo e navegação.</li></ul>
<b>Protótipo de alta-fidelidade</b>	<ul style="list-style-type: none"><li>• Funcionalidade completa.</li><li>• Totalmente interativo.</li><li>• Uso conduzido pelo usuário.</li><li>• Define claramente o esquema de navegação.</li><li>• Uso para exploração e teste.</li><li>• Mesmo <i>look and feel</i> do produto final.</li><li>• Serve como uma especificação viva.</li><li>• Ferramenta de venda e <i>marketing</i>.</li></ul>	<ul style="list-style-type: none"><li>• Desenvolvimento mais caro.</li><li>• Sua criação demanda tempo.</li><li>• Ineficiente para <i>designs proof-of-concept</i> (demonstrações de que o conceito funciona).</li><li>• Não serve para coleta de requisitos.</li></ul>

# Prototipação de média fidelidade

- Mais recentemente, alguns autores introduziram a noção de protótipos de média fidelidade.
- É uma classe intermediária que agrega vantagens dos protótipos de baixa fidelidade e de média fidelidade.
- Consiste na implementação computadorizada de uma aplicação limitada funcionalmente, contendo apenas as funções essenciais para avaliar alguns cenários específicos.

# Prototipação de média fidelidade

- Representação do sistema (ou parte do sistema) usada para apoiar estudos com o usuário, sendo tipicamente utilizados para testar prematuramente conceitos de projeto no processo de concepção da interface.
- Características mantidas dos protótipos de baixa fidelidade:
  - Rapidez e facilidade de construir/modificar
  - Possibilidade de explorar diferentes alternativas de projeto
  - Melhoria na comunicação da equipe

# Prototipação de média fidelidade

- Características mantidas dos protótipos de alta fidelidade:
  - Interação direta entre usuário e o sistema;
  - Manutenção do histórico do projeto;
  - Reuso de partes do projeto;
  - Possibilidade de realização de testes de usabilidade e treinamento.

# Prototipação de média fidelidade

- Resumindo: utilização de uma técnica de prototipagem de baixa-fidelidade (esboços, storyboards) com um suporte computacional, de modo a possibilitar a simulação de seu comportamento tal como um protótipo de alta-fidelidade, favorecendo a prática da prototipagem exploratória e evolutiva.

# Prototipação de média fidelidade

- Ferramentas usuais tais como editores de HTML, Microsoft Power Point ou Microsoft Visio podem ser utilizadas para construir protótipos de média-fidelidade.
- No entanto, ferramentas específicas vêm sendo construídas desde a década de 90 a fim de possibilitar a construção de esboços de interface com o auxílio de recursos computacionais.



# Prototipação de média fidelidade

- Ex de algumas ferramentas:

- Pencil project: <http://pencil.evolus.vn/>

- Mockingbird: <https://gomockingbird.com/>

- Cacao:

- [https://cacao.com/lang/pt\\_br/;jsessionid=8FB0](https://cacao.com/lang/pt_br/;jsessionid=8FB0)

- Axure: [axure.com](http://axure.com)

- Proface:

- <http://sourceforge.net/projects/proface/>

# Prototipação de média fidelidade

- Protótipo navegável com Axure:

<http://www.prototiponavegavel.com.br/>

- Exercício: prototipar sistema para biblioteca

# Concessões na prototipação

- Os protótipos envolvem concessões: a intenção é produzir algo rapidamente a fim de testar algum aspecto do produto.
- O tipo de questões ou escolhas a que qualquer protótipo permite ao designer responder é, portanto, limitado.
- Ele deve ser projetado e construído tendo as questões principais em mente.
- Ex: Sabemos que um protótipo de papel não irá funcionar; os ícones de um protótipo podem ser rudimentares

# Concessões na prototipação

- Duas compensações comuns que geralmente devem ser contrabalanceadas são a amplitude da funcionalidade (prototipação horizontal) versus a profundidade (prototipação vertical)
  - Prototipação horizontal: fornece ampla gama de funções, mas com poucos detalhes.
  - Prototipação vertical: fornece muitos detalhes para poucas funções.

# Concessões na prototipação

- Um dos perigos de se produzirem protótipos executáveis, isto é, com os quais se pode interagir automaticamente, é o fato de os usuários poderem acreditar que ele constitui o próprio sistema.
- O perigo para os desenvolvedores é o protótipo levá-los a considerar menos alternativas.

# Construção: do design à implementação

- Quando o design tiver passado pelos ciclos de iterações o número de vezes suficiente para que se tenha segurança de que se enquadra nos requisitos, tudo o que foi aprendido nos passos iterados da prototipação e da avaliação deve ser integrado na realização do produto final.

# Construção: do design à implementação

- Filosofias de desenvolvimento:
  - Prototipação evolutiva: envolve fazer o protótipo evoluir para o produto final. Os protótipos devem ser submetidos a testes rigorosos durante o desenvolvimento.
  - Prototipação descartável: utiliza os protótipos como pontos de apoio para o design final. Os protótipos são jogados fora e o produto final é construído a partir do zero.



# Construção: do design à implementação

- Quando a equipe de um projeto está sob pressão e possui um protótipo complexo que preenche muitos requisitos (ou um conj. de protótipos verticais que juntos preenchem os requisitos), pode ser tentador colocá-los juntos e apresentar o resultado como produto acabado.
- Isso irá adiar os problemas de manutenção e teste para mais tarde. O que comprometerá a qualidade do produto.

# Construção: do design à implementação

- Fazer com que o protótipo final evolua para o produto final mediante um processo definido de prototipação evolutiva pode levar a um produto final resistentes.
- Mas isso deve ser claramente planejado e projetado desde o início.

# Design conceitual: dos requisitos ao primeiro design

- O design conceitual preocupa-se com transformar os requisitos e as necessidades do usuário em um modelo conceitual.
- Modelo conceitual é uma descrição do sistema sobre o que ele deveria fazer, como se comportar e com o que se parecer – que seria compreensível pelos usuários da maneira pretendida.
- A base para se projetar esse modelo é o conjunto de tarefas do usuário ao qual o produto irá oferecer suporte.

# Design conceitual: dos requisitos ao primeiro design

- Não há uma transformação fácil que se aplique a um conjunto de requisitos para a produção do “melhor” modelo conceitual ou mesmo de um “bom o suficiente”.
- A partir dos requisitos e da experiência com os dados e com os usuários, uma idéia acerca do que você pretende que seja a experiência dos usuários utilizando o novo produto irá emergir.
- As idéias podem emergir durante a coleta de dados; você deve separar os requisitos reais das idéias para soluções.

# Design conceitual: dos requisitos ao primeiro design

- Princípios-chave norteadores do design conceitual:
  - Manter a mente aberta, mas nunca esquecer os usuários e seu contexto.
  - Discutir idéias com outros stakeholders o máximo possível.
  - Utilizar prototipação de baixa-fidelidade para obter rápido feedback.
  - Iterar, iterar, iterar (“para ter uma boa idéia, tenha muitas idéias”).

# Design conceitual: dos requisitos ao primeiro design

- Considerar alternativas e pensar repetidamente sobre perspectivas diferentes auxilia a expandir o espaço da solução e pode ajudar a inspirar percepções (acerca da experiência do usuário).

# Três perspectivas para desenvolver um modelo conceitual

## ■ Que modo de interação?

- O modo de interação é a forma como o usuário invoca ações quando interage com o aparelho.
- Ele vai depender das atividades que o usuário pretende realizar quando o utiliza.
- Existem 2 tipos de modos de interação: baseados em atividades (estilos: instruir, conversar, manipular e navegar, explorar e pesquisar) e em objetos.
- O melhor depende do domínio de aplicação e do tipo de sistema que está sendo desenvolvido.



# Três perspectivas para desenvolver um modelo conceitual

## ■ Que modo de interação?

- A maioria dos modelos conceituais será uma combinação de modos, e poderá associar partes diferentes da interação com os diferentes modos.
- Ex: pense nos modos de interação apropriados para o design da agenda eletrônica.
- Os modelos baseados em objetos oferecem uma perspectiva diferente, uma vez que são estruturados em torno de objetos do mundo real.

# Três perspectivas para desenvolver um modelo conceitual

- Existe uma metáfora de interface adequada?
  - As metáforas são destinadas a combinar conhecimento familiar com conhecimento novo, de uma maneira que ajude o usuário a entender o sistema.
  - Para escolher uma boa metáfora é importante ter um bom entendimento acerca dos usuários e de seu contexto.

# Três perspectivas para desenvolver um modelo conceitual

- 3 passos para escolha de uma boa metáfora:
  - Entender o que o sistema irá realizar
  - Entender quais partes do problema podem causar complicações aos usuários
  - Gerar metáforas.
- Procurar metáforas na descrição das tarefas feitas pelos usuários representa um bom ponto de partida.

# Três perspectivas para desenvolver um modelo conceitual

- 5 perguntas para avaliar interfaces:
  - A metáfora proporciona uma boa estrutura?
  - Quanto da metáfora é relevante para o problema?
  - A metáfora é fácil de representar?
  - Seu público entenderá a metáfora?
  - Quão extensível é a metáfora?

# Três perspectivas para desenvolver um modelo conceitual

- Que paradigma de interação usar?
  - Os paradigmas são filosofias de design que ajudam você a pensar sobre o produto que está sendo desenvolvido.
  - Paradigmas de interação: computação ubíqua, computação pervasiva, computação vestível, bits tangíveis, realidade aumentada e integração física/virtual, ambientes atentos, workday world.
  - Pensar sobre requisitos do ambiente é relevante quando se considera paradigmas de interação.

# Expandindo o modelo conceitual

- Além das 3 questões anteriores, outras idéias devem ser pensadas antes de ser prototipadas:
  - Que tecnologias usar?
  - A que funções o produto irá oferecer suporte?
  - Como as funções se relacionam uma com a outra?
  - Que informação será exigida para obter o suporte?

# Utilizando cenários no design conceitual

- Cenários são uma descrição narrativa informal.
- Eles podem ser utilizados no design conceitual visando explicar situações de trabalho ainda que sejam mais comumente utilizados para expressar situações propostas ou imaginadas.
- Papéis sugeridos para cenários:
  - Como base para um design completo
  - Para implementação técnica
  - Como meio de cooperação dentro de equipes de design
  - Como meio de cooperação entre fronteiras profissionais

# Utilizando protótipos no design conceitual

- A prototipação é usada para se ter feedback dos designs emergentes.
- O feedback pode ser de usuários ou colegas e pode lhe informar que determinada técnica é inviável.
- Diferentes tipos de protótipos são utilizados em momentos diferentes das iterações de desenvolvimento e com pessoas diferentes.
- Que tipo de protótipo você faria se estivesse esperando a aprovação das pessoas que farão julgamento do protótipo?



# Design físico: tornando-o concreto

- O design físico envolve considerar questões mais concretas e detalhadas acerca do projeto de uma interface (como design de tela ou teclado, quais ícones utilizar, como estruturar menus)
- Não há uma fronteira rígida entre design conceitual e físico.
- O design de interação é inerentemente interativo e, portanto, algumas questões que envolvem detalhes surgirão durante o design conceitual; e durante o design físico será necessário revisitar decisões tomadas durante o design conceitual.

# Design físico: tornando-o concreto

- O design diz respeito a fazer escolhas e tomar decisões, e o designer deve se esforçar para contrabalançar requisitos ambientais, de usuário, de dados e de usabilidade com requisitos funcionais.
- Frequentemente esses requisitos entram em conflito.

# Recomendações (guidelines) para o Design físico

- A maneira como projetamos a interface física do produto não pode entrar em conflito com os processos cognitivos (atenção, percepção, memória, etc) dos usuários envolvidos.
- Uma ampla gama de recomendações, princípios e regras foi desenvolvida para ajudar os designers a assegurar-se de que seus produtos são utilizáveis.
- As recomendações de Nielsen foram apresentadas anteriormente em forma de heurísticas.

# Recomendações (guidelines) para o Design físico

- Outras recomendações são as regras de ouro de Shneiderman:
  1. Esforce-se pela consistência
  2. Possibilite que usuários frequentes utilizem atalhos
  3. Ofereça feedback informativo
  4. Projete diálogos para encerrar as ações.
  5. Ofereça prevenção contra erros e manuseio fácil dos mesmos
  6. Permita uma reversão fácil das ações
  7. Faça com que o usuário se sinta no controle
  8. Reduza a carga de memória de curto prazo.

# Tipos diferentes de widgets

- As interfaces são feitas de widgets: elementos como caixas de diálogo, menus, ícones, barras de ferramentas, etc.
- Cada elemento deve ser projetado ou escolhido a partir de um conjunto pré-projetado de widgets.
- Algumas vezes, essas decisões são tomadas mediante um guia de estilo.
- Um guia de estilo pode ser produzido comercialmente como os do Windows, ou ser internos de uma empresa.

# Tipos diferentes de widgets

- Um guia de estilo dita a aparência e o modo de atuar (look and feel) de uma interface, isto é, que elementos devem ser utilizados para determinados propósitos e como eles são.
- Consideraremos 3 aspectos principais do design da interface:
  - Design do menu
  - Design dos ícones
  - Design da tela

# Design do menu

- Menus proporcionam escolhas que podem ser de comandos ou opções relacionadas a um comando.
- Fornecem meios pelos quais o usuário pode realizar ações relacionadas à tarefa em questão.
- O modo como projetar um menu pode parecer óbvio, mas se você quiser fazer com que a aplicação seja fácil de utilizar e proporcionar satisfação ao usuário, alguns pontos devem ser levados em conta.
  - funções mais comumente utilizadas devem estar no topo.
  - opções opostas como “fechar” e “salvar” devem ser claramente separadas.

# Design do menu

## ■ Parte da ISO 9241 (ergonomia de software de escritório)

### 5.2 Agrupando opções em um menu

As opções do menu devem ser agrupadas de modo a refletir as expectativas do usuário e facilitar a escolha das opções.

#### 5.2.1 Grupos lógicos

Se o menu contém um grande número de opções (8 ou +) e essas opções podem ser logicamente agrupadas, elas devem ser agrupadas por função ou por outras categorias lógicas que sejam significativas para os usuários:

Ex: agrupar comandos em um processador de texto, em categorias como personalizar, compor, editar, imprimir

#### 5.2.2 Grupos arbitrários

Se 8 ou + opções são agrupadas arbitrariamente em um painel de menu, elas devem ser organizadas em grupos igualmente distribuídos utilizando a seguinte equação:

$g = \text{raiz de } n$  onde  $g$  é o número de grupos e  $n$  é o número de opções no painel

Ex: dadas 19 opções, organize-as em 4 grupos de 5



# Design dos ícones

- Projetar um bom ícone demanda tempo.
- Eles são culturais e específicos do contexto.
- Os designers deveriam sempre lançar mão de tradições ou padrões existentes e certamente não contradizê-los.
- Objetos ou coisas concretas são mais fáceis de representar por um ícone, mas ações são mais difíceis.
- Ícones devem ser projetados de maneira que os usuários possam prontamente perceber o seu significado e devem ser distinguíveis uns dos outros.

# Design da tela

- Existem dois aspectos para o design de tela: como a tarefa é dividida nas diferentes telas e como as telas individuais são projetadas.
- Pode-se oferecer suporte ao primeiro aspecto fazendo-se referência à análise de tarefas, que dividiu a tarefa em subtarefas e planos de ação.
- Um plano de partida para o design de tela consiste em traduzir a análise de tarefas nas telas, de forma que cada tarefa ou subtarefa disponha de sua própria tela.
- Uma das ponderações que devem ser feitas: dispor de muitas telas simples tornar-se-ia tão frustrante quanto a apresentação das informações todas em uma única tela.

# Design da tela

- Uma outra questão que afeta a divisão de uma tarefa em várias telas diz respeito ao fato de que todas as informações pertinentes devem estar facilmente acessíveis nos momentos relevantes.
- As recomendações para o design individual de telas:
  - Projetar tela de forma que a atenção dos usuários seja dirigida imediatamente para pontos importantes (elementos apropriados e úteis para a tarefa em questão).
  - Utilizar cores, movimento, caixas e agrupamentos a fim de ajudar no entendimento e na clareza das informações.
  - Animações podem ser dispersivas se não forem relevantes para a tarefa, mas eficazes se usadas com bom senso.

# Design da tela

- Uma boa organização auxilia os usuários a compreender e interpretar a interação dentro de seu próprio contexto.
- Deve haver um equilíbrio entre telas não muito cheias, que disponham de bastante espaço vazio, e telas sobrecarregadas com muitos e complicados ícones.
- Se a tela estiver sobrecarregada, o usuário ficará confuso e irá distrair-se. Mas muito espaço vazio e conseqüentemente maior número de telas poderá levá-lo a mudar com frequência de tela e a realizar uma série de interações desconexas.

# Design da tela: exibição da informação

- Certificar-se de que as informações relevantes para a realização de uma tarefa estão disponíveis constitui um dos aspectos deste item.
- Diferentes tipos de informação requerem diferentes tipos de apresentação:
  - Dados por natureza discretos, como os indicativos de vendas do último mês, poderiam ser exibidos graficamente utilizando-se a técnica digital.
  - Dados por natureza contínuos, como os que dizem respeito ao aumento na porcentagem de vendas no último mês, são melhor exibidos usando-se um dispositivo analógico.

# Referências

- Sharp, H.; Rogers, Y.; Preece, J.; Design de Interação: além da interação homem-computador. Ed. Artmed, 2005. ISBN: 8536304944.
- Aguiar, Y. P. (2007) SMILE – uma Ferramenta para Geração Automática, Edição e Simulação de Protótipos de Interface do Usuário. *Dissertação (Mestrado em Informática) – COPIN/UFCG, Campina Grande, Brasil.*