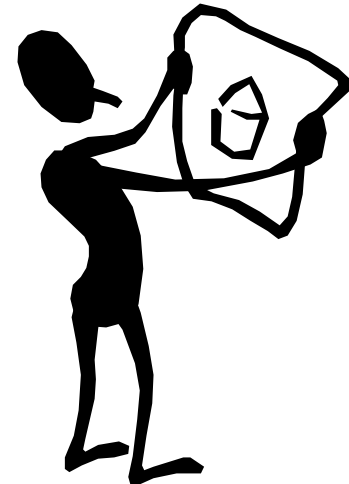


# Projeto de Software

Engenharia de Software

*Simone do Rocio Senger de Souza*  
*srocio@icmc.usp.br*





# Engenharia de Software

---

## 1- Definição ("o que")

- **Análise do sistema**
- **Planejamento do projeto de software**
- **Análise de requisitos**

## 2- Desenvolvimento ("como")

- ***Projeto de software***
- **Codificação**
- **Testes**

## 3- Manutenção ("alterações")

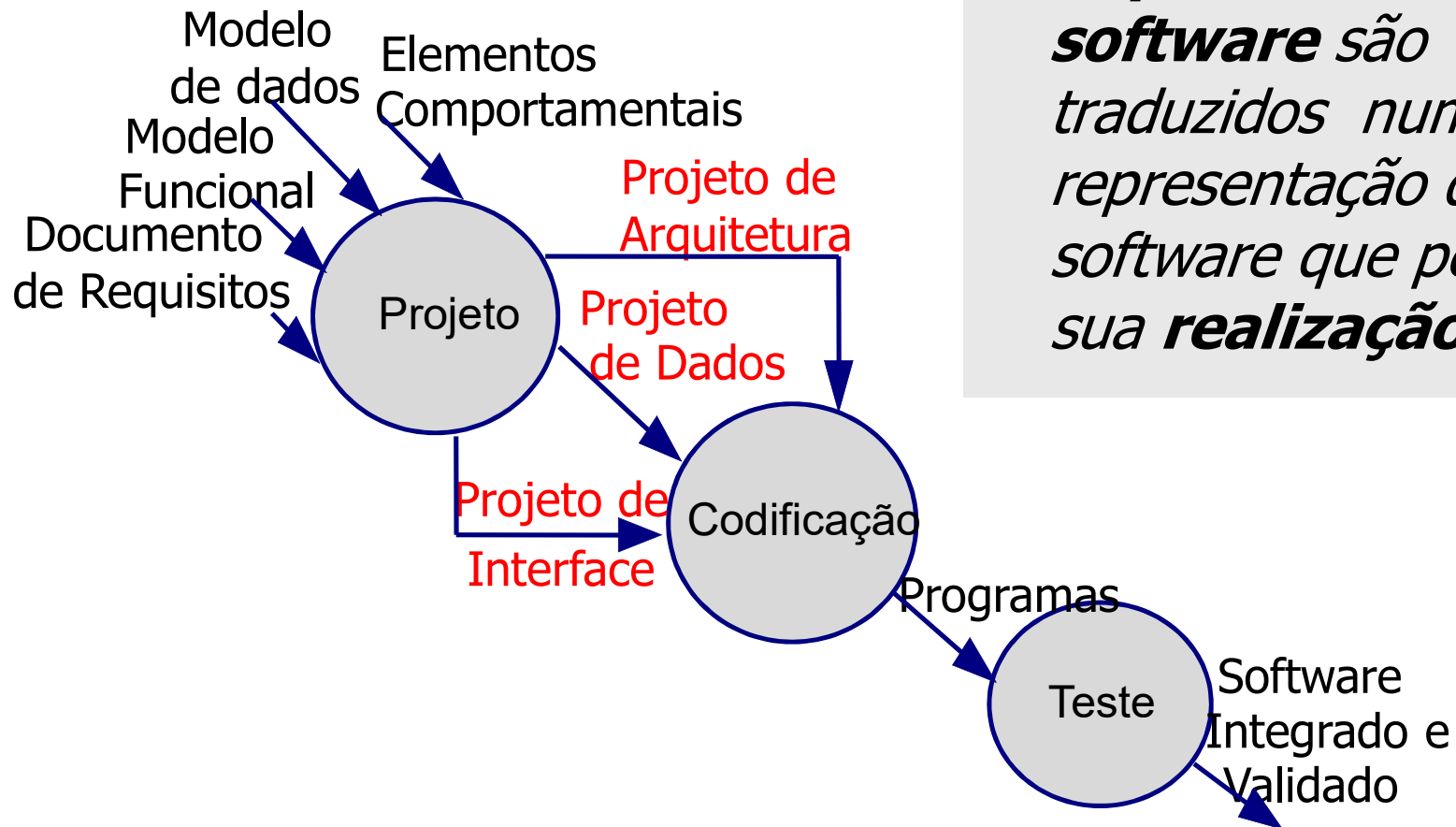


## Atividades Guarda-Chuva

- Controle de Projeto
- Revisões Técnicas Formais
- Garantia de Qualidade
- Gerenciamento de Configuração
- Gestão de Reutilização
- Medição
- Gestão de Risco

# Projeto de Software

*Processo pelo qual os requisitos do software são traduzidos numa representação do software que permite sua **realização física***





# *Projeto de Software*

---

Análise

Projeto

←

-O que

-Requisitos

-Investigação do domínio

→

-Como

-Solução lógica

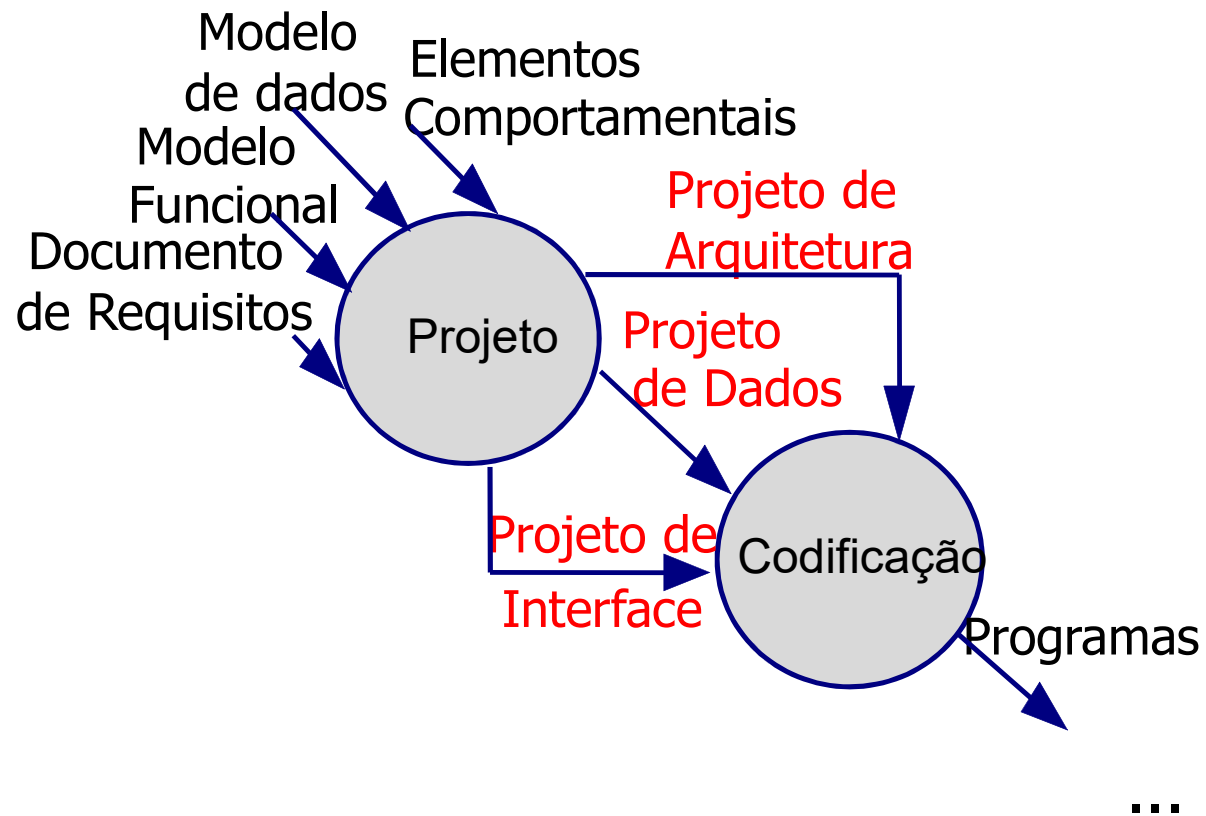


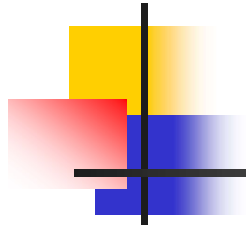
# *Projeto de Software*

---

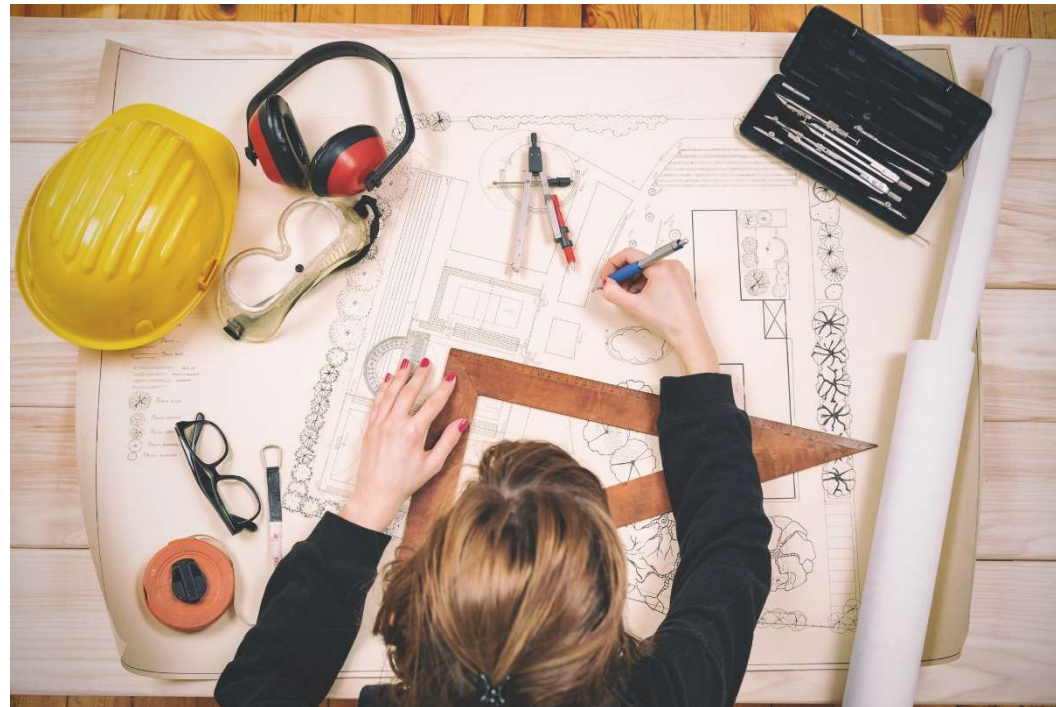
- Especificação de como será desenvolvida uma **solução** levantada na fase de análise
- Detalhes de implementação:
  - Ambientes de implementação adotado
    - linguagem, sistema operacional, banco de dados...
  - Arquitetura do software
    - módulos, comunicação entre módulos, ordem de execução
  - Detalhamento algorítmico
  - Aspectos de eficiência, desempenho e confiabilidade
    - Fundamental para sistema de tempo real, processamento paralelo, sistemas distribuídos...

# Projeto de Software





# *Projeto de Arquitetura*





# *Projeto de Arquitetura*

---

- É o primeiro estágio do processo de projeto de sistema.
- Envolve a identificação dos componentes principais do sistema e suas comunicações
- A escolha da arquitetura do sistema pode afetar o desempenho, facilidade de distribuição e manutenção do sistema
  - A escolha pode ser influenciada pelos requisitos não funcionais





# *Exemplos de características de arquitetura*

---

- Desempenho
  - Localizar operações críticas e minimizar comunicações. Usar componentes de alta granularidade – menos comunicação.
- Proteção
  - Usar uma arquitetura em camadas com itens críticos nas camadas mais internas.
- Segurança
  - Definir sub-sistemas exclusivos para tratar de aspectos de segurança do sistema
- Disponibilidade
  - Incluir componentes redundantes e mecanismos para tolerância a falhas.
- Facilidade de manutenção
  - Usar componentes substituíveis e de baixa granularidade.



# *Conflitos de arquitetura*

---

- O uso de componentes de alta granularidade aprimora o desempenho mas diminui a facilidade de manutenção.
- A introdução de dados redundantes aprimora a disponibilidade, mas torna a proteção mais difícil.



# *Modelo de Arquitetura*

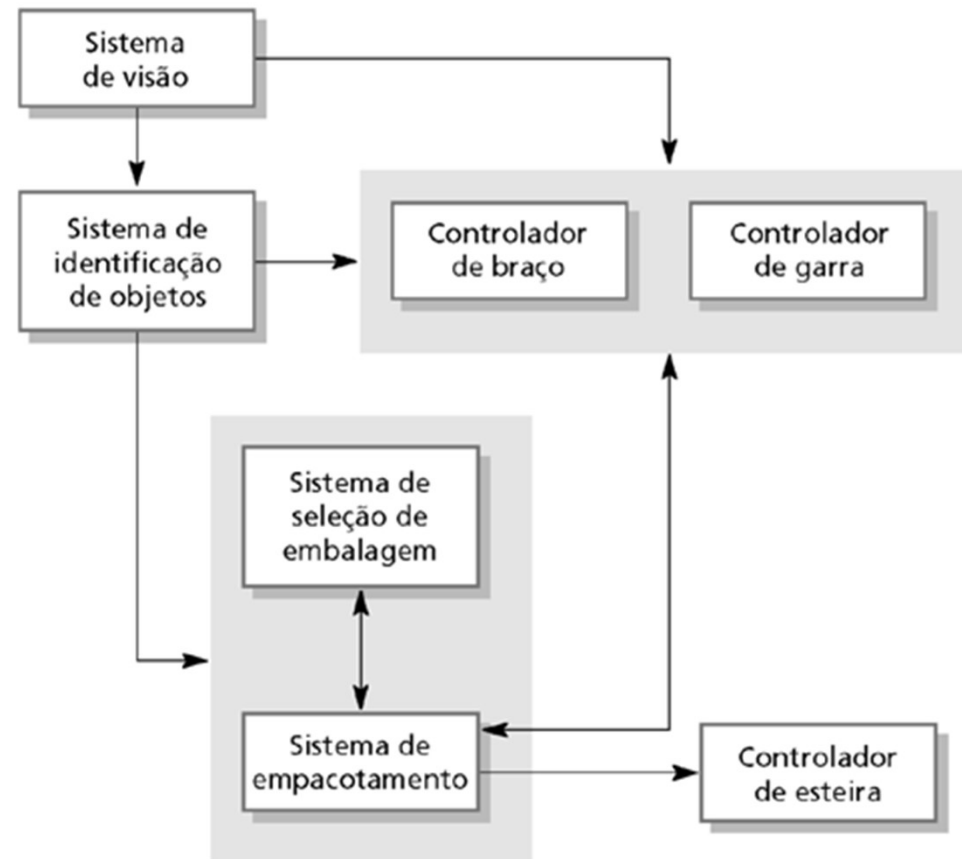
---

- Relacionado à decomposição do sistema em subsistemas que interagem.
- Expresso como um diagrama de blocos que apresentam uma visão geral da estrutura do sistema.

# Exemplo de Arquitetura

**Figura 11.1**

Diagrama de blocos de um sistema de controle robotizado de empacotamento.



# *Decisões de projeto de arquitetura*



---

- Existe uma arquitetura genérica de aplicação que possa ser usada?
- Como o sistema será distribuído?
- Quais estilos de arquitetura são apropriados?
- Qual será a abordagem fundamental usada para estruturar o sistema?
- Como o sistema será decomposto em módulos?
- Como o projeto de arquitetura será validado?
- Como a arquitetura do sistema deve ser documentada?



## *Reuso de arquitetura*

---

- Sistemas do mesmo domínio frequentemente têm arquiteturas similares que refletem os conceitos de domínio.
  - Arquitetura de referência
- As **linhas do produto** de aplicação são construídas em torno de um núcleo de arquitetura com variantes que satisfazem requisitos específicos de clientes.



# *Padrões de Arquitetura*

---

- Reflete a estratégia básica que é usada para estruturar um sistema.
- Padrões amplamente usados:
  - Arquitetura em camadas
  - Arquitetura de repositório de dados
  - Arquitetura cliente-servidor

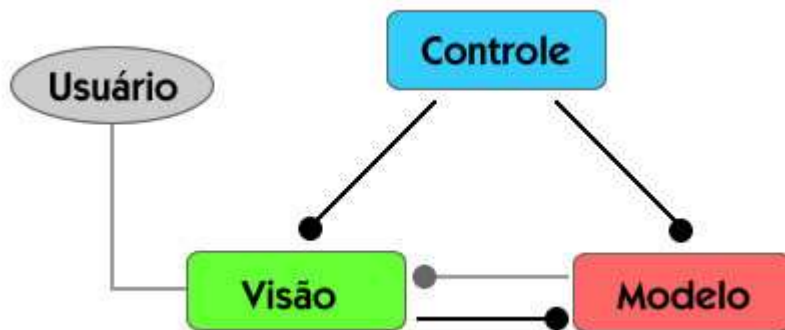
# Arquitetura em Camadas



- Usado para modelar o interfaceamento dos subsistemas.
- Organiza o sistema em um conjunto de camadas (ou máquinas abstratas), cada uma das quais fornecendo um conjunto de serviços.
- Apóia o desenvolvimento incremental dos subsistemas em camadas diferentes. Quando uma camada de interface muda, somente a camada adjacente é afetada.



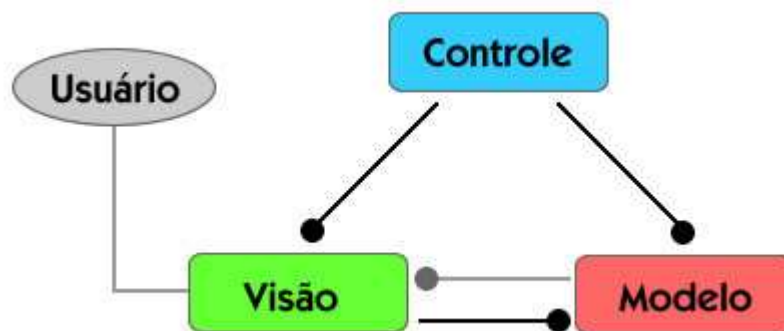
# Exemplo: Padrão MVC



Quando usar?

Quando existirem várias maneiras de visualizar e interagir com os dados

# Exemplo: Padrão MVC



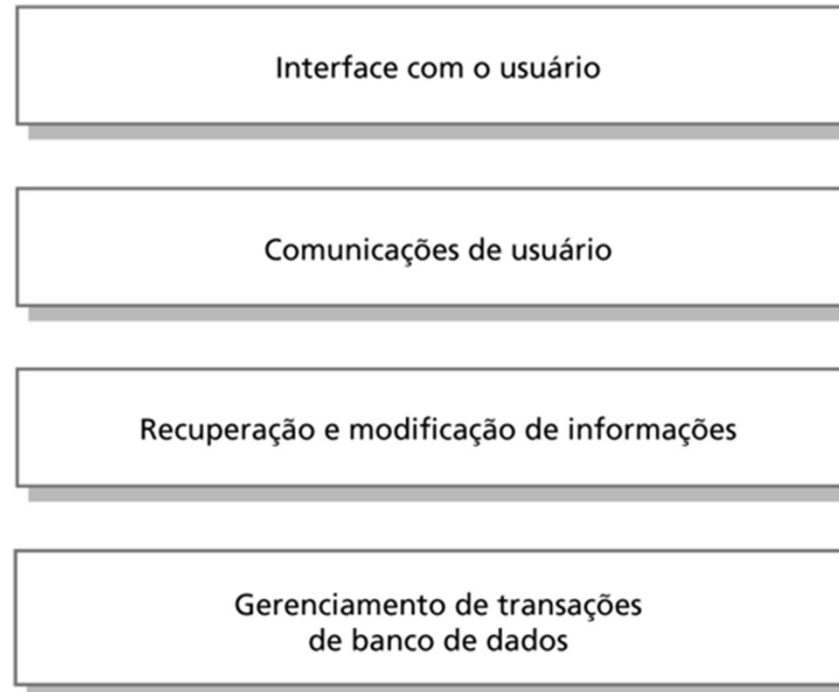
Vantagem: apoia a apresentação dos dados de maneira diferentes

Não apropriado para dados e interações simples

# Exemplo de Arquitetura em Camadas

**Figura 13.6**

Modelo em camadas de um sistema de informações.



# Exemplo de Arquitetura em Camadas



## Quando usar?

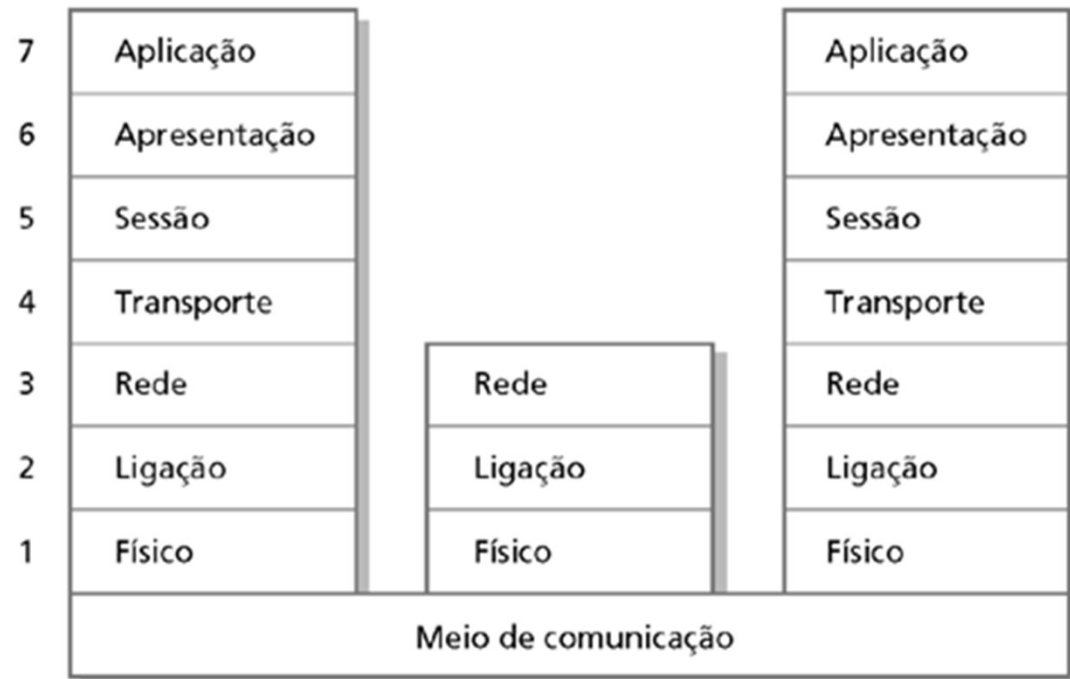
- Construção de recursos em cima de sistemas já existentes;
- Recursos espalhados em várias equipes;
- Cada equipe responsável por uma camada;
- Necessidade de proteção multinível

# Exemplo de Arquitetura em Camadas

**Figura 11.11**

Arquitetura de modelo de referência OSI.

Clara separação das camadas





# *Padrões de Arquitetura*

---

- Reflete a estratégia básica que é usada para estruturar um sistema.
- Padrões amplamente usados:
  - Arquitetura em camadas
  - Arquitetura de repositório de dados
  - Arquitetura cliente-servidor

# Arquitectura de repositório de dados

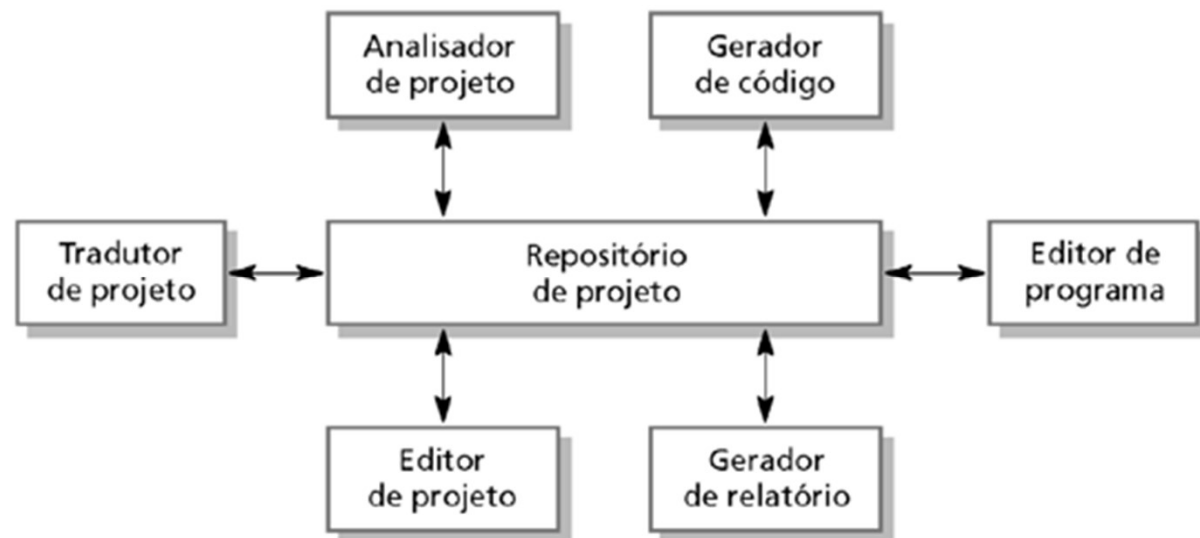
- Os dados de um Sistema são gerenciados em um repositório central
  - Os componentes do Sistema não interage entre si, somente por meio do repositório



# Exemplo: Arquitetura de repositório de dados

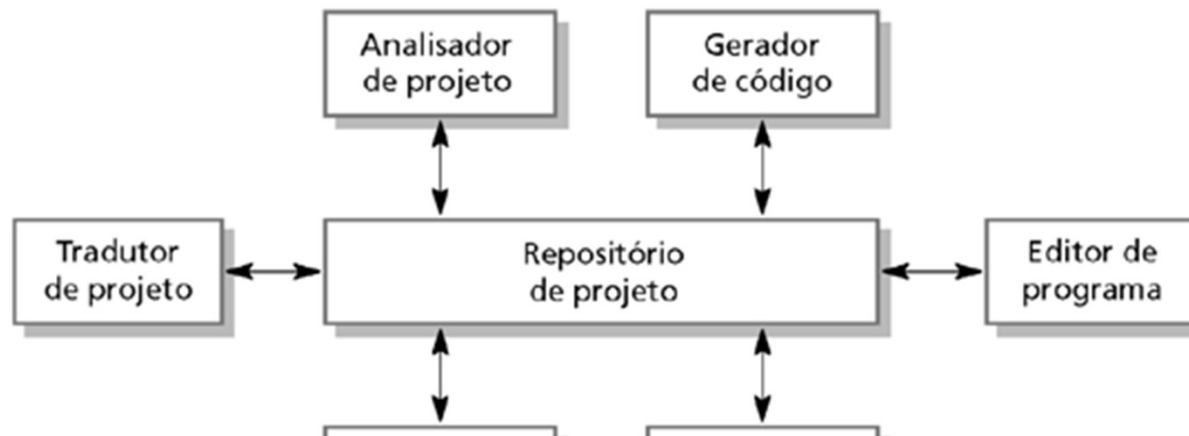
Figura 11.2

Arquitetura de um conjunto de ferramentas CASE integradas.





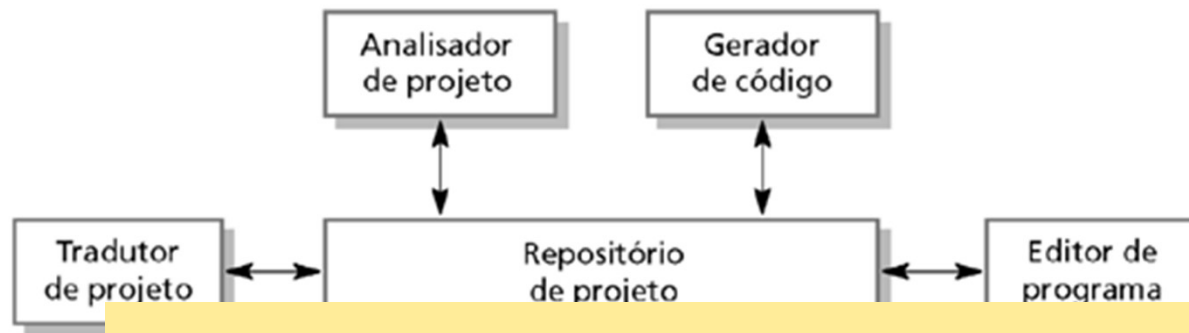
# Exemplo: Arquitetura de repositório de dados



## Quando usar?

Sistemas em que grande volume de informações são gerados e precisam ser armazenados por longos períodos  
Em sistemas dirigidos a dados (dados disparam ações no sistema)

# *Exemplo: Arquitetura de repositório de dados*



## ■ Desvantagens:

- Problemas no repositório afetam todo o sistema
- Os subsistemas devem estar de acordo com o modelo de repositório
- Problemas de ineficiência



# *Padrões de Arquitetura*

---

- Reflete a estratégia básica que é usada para estruturar um sistema.
- Padrões amplamente usados:
  - Arquitetura em camadas
  - Arquitetura de repositório de dados
  - Arquitetura cliente-servidor



# *Arquitetura cliente-servidor*

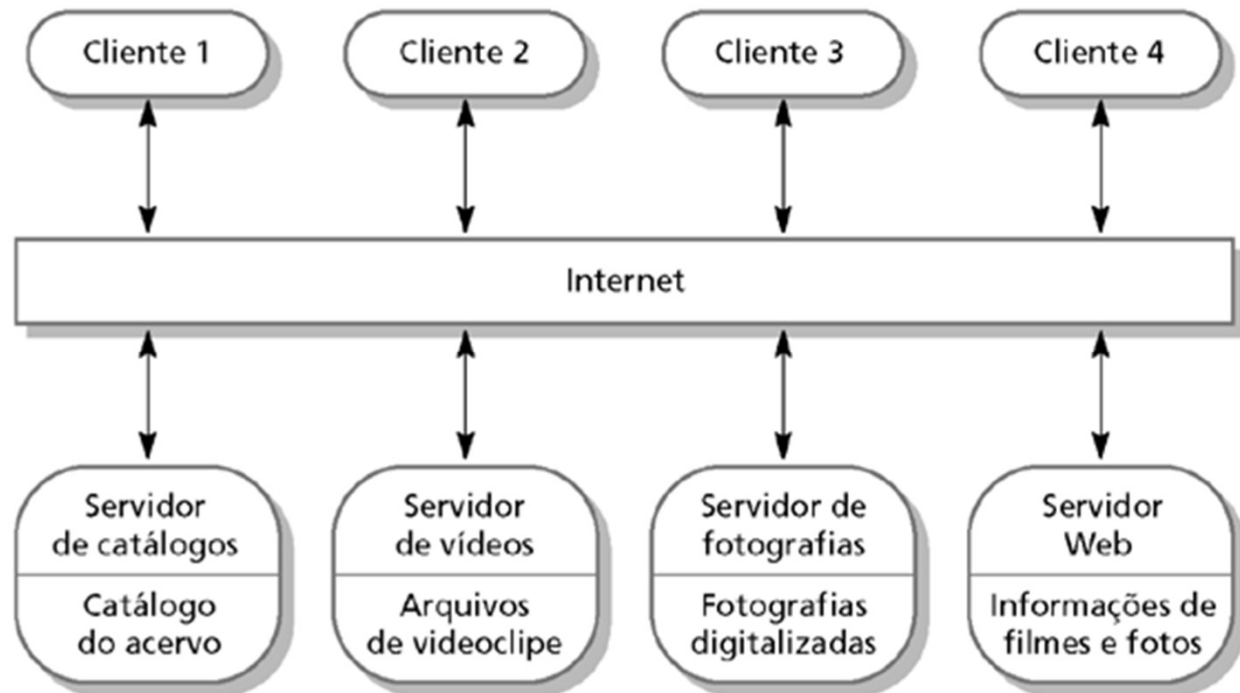
---

- A funcionalidade do Sistema está organizada em serviços
  - Cada serviço prestado por um servidor
- Ideal para sistemas distribuídos, pois mostra como dado e processamento são distribuídos por uma variedade de componentes.

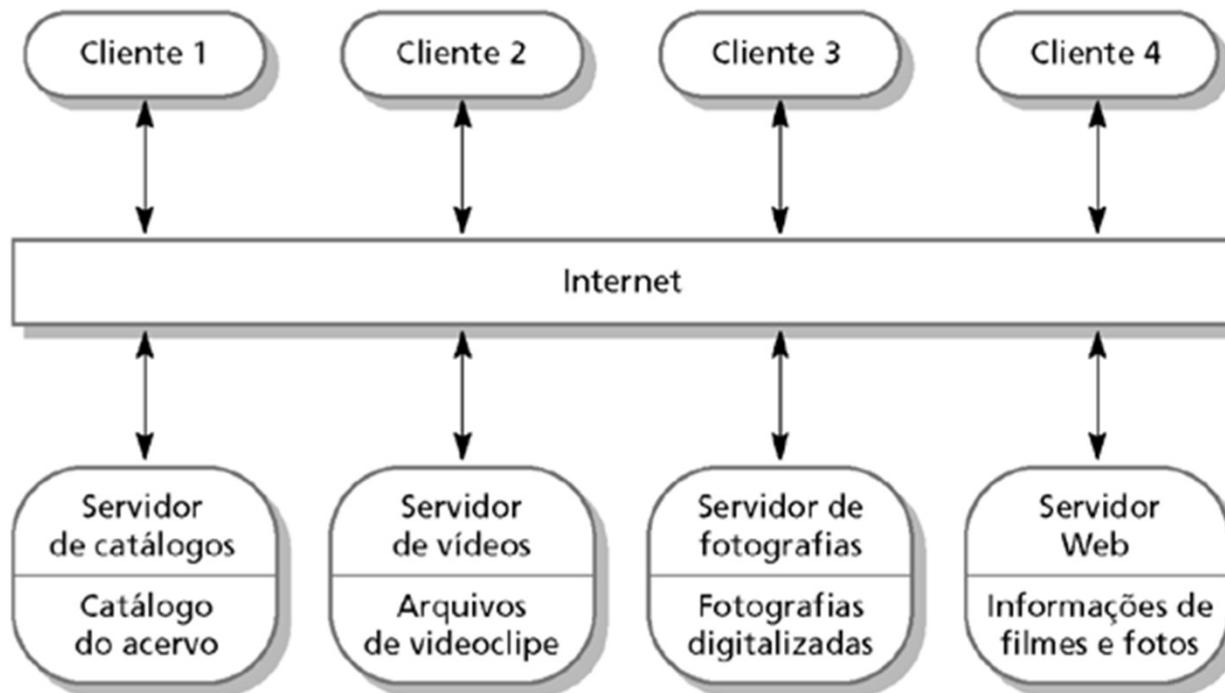
# Modelo cliente-servidor

Figura 11.3

Arquitetura de um sistema de acervo de filmes e fotografias.



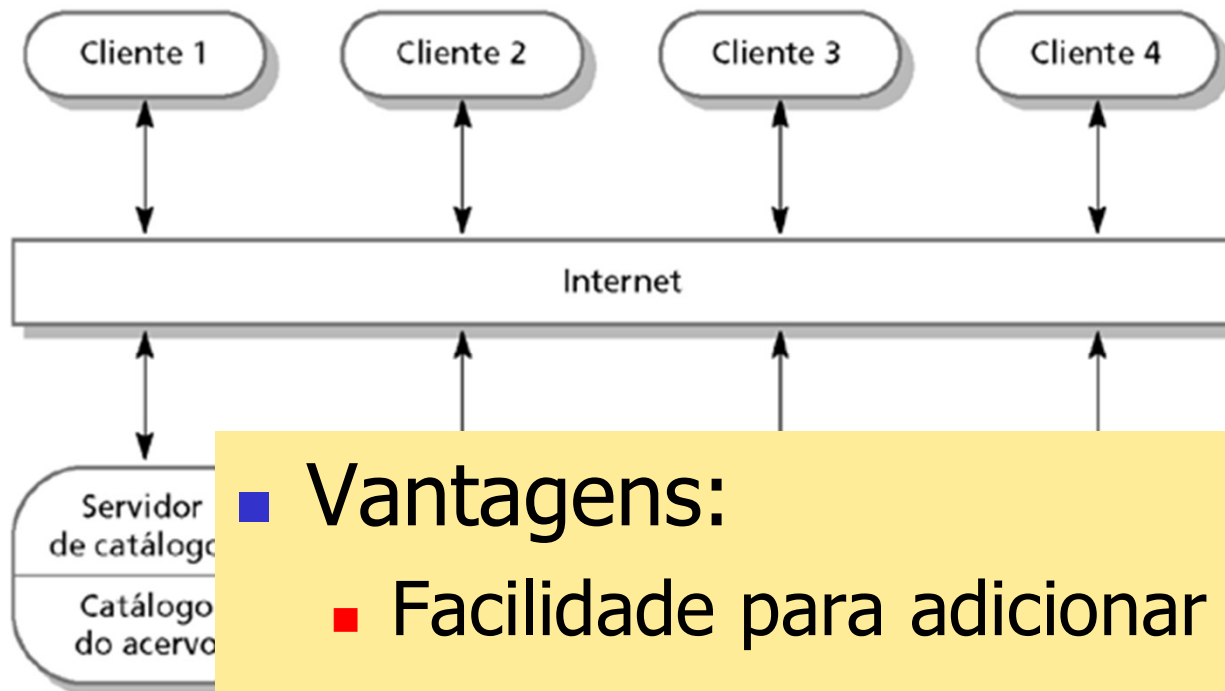
# Modelo cliente-servidor



## Quando usar?

Quando os dados em um BD compartilhado precisam ser acessados de vários locais

# Modelo cliente-servidor



- **Vantagens:**

- Facilidade para adicionar novos serviços.

- **Desvantagem:**

- Problemas com desempenho da rede.

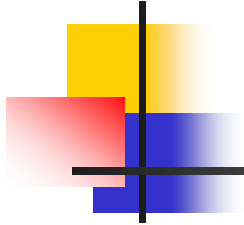


# *Qual padrão de arquitetura utilizar?*

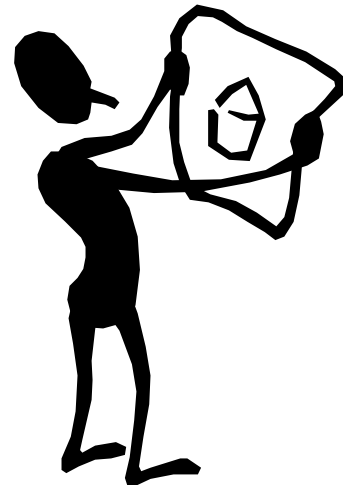
---

1. Sistema de emissão de bilhetes usado por passageiros em uma estação ferroviária
2. Sistema de videoconferência que permite que áudio, vídeo e dados sejam visíveis a todos os participantes ao mesmo tempo
3. Um robô limpador de chão, que se destina a limpar espaços livres. O robô deve perceber obstáculos e paredes.





# ***Projeto de Dados***





# *Projeto de Dados*

---

- Retrata as **estruturas de dados** gerais e sucessivamente acrescenta-se detalhes sobre os **elementos de dados** e seus **relacionamentos**
- A partir de **modelos da análise** (ou **DR**), o projeto concentra-se nos elementos que devem ser manipulados pelo sistema.



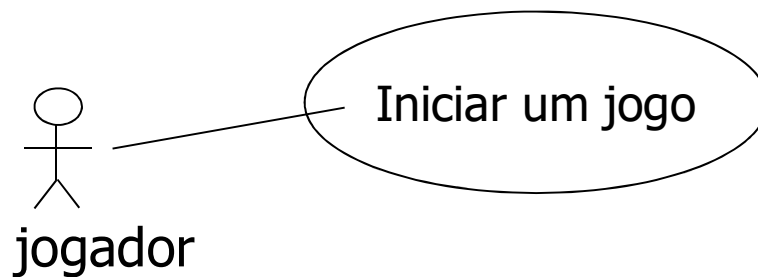
# *Projeto de Dados*

---

- Pode empregar modelos de BD (DER)
  - A partir dos dados extraídos, estabelece-se as funções que irão manipular esses dados.
- Pode empregar modelos OO (Diagrama de Classes)
  - Identifica classes de objetos e seus relacionamentos
  - Apresenta os atributos de cada objeto, ações e relacionamentos

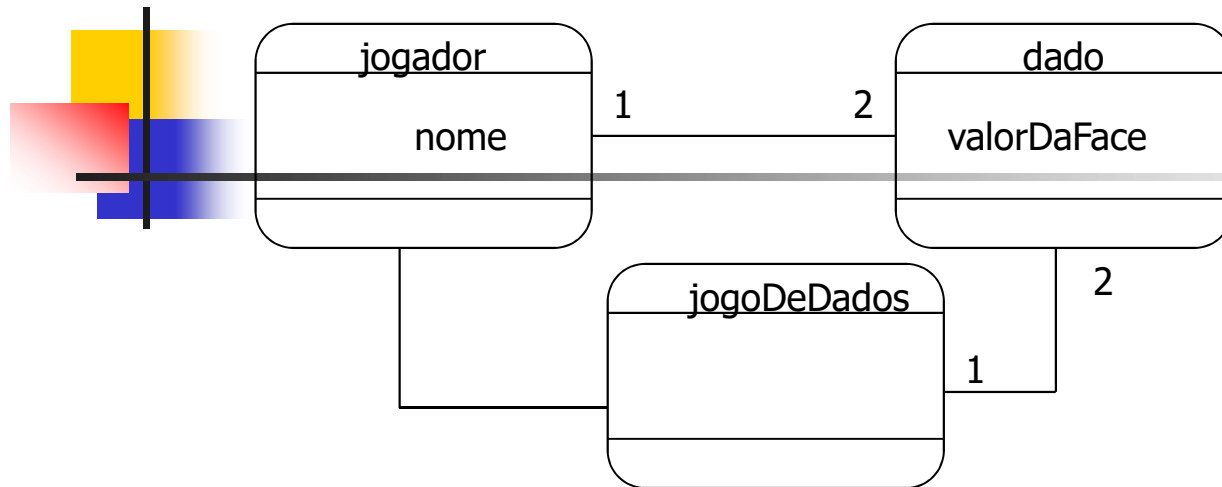
# Projeto OO

- Exemplo:



Descrição: este caso de uso começa quando o jogador lança os dados. Se o total dos dados é sete, os dados vencem; caso contrário o jogador ganha.

# Projeto OO



Modelo conceitual  
(domínio do problema)

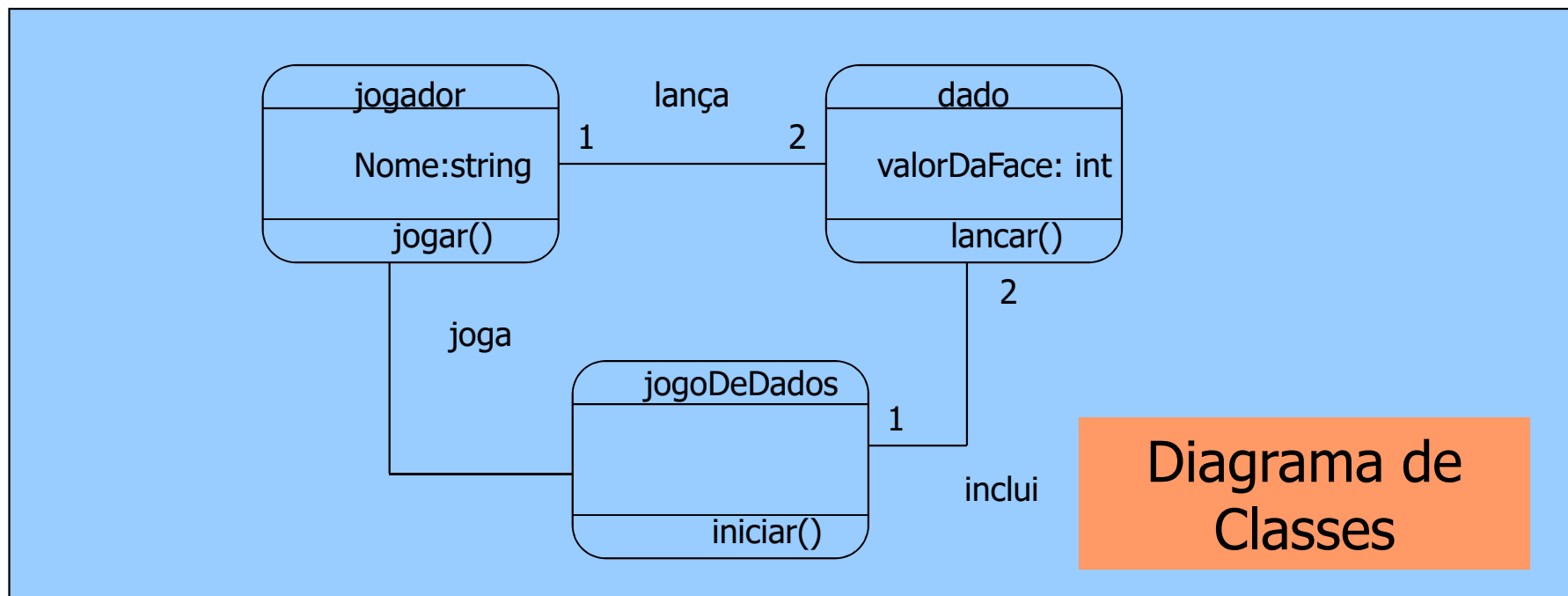


Diagrama de  
Classes



# *Projeto de Dados*

---

- Gera o Diagrama de Classes com informações relevantes para implementação
  - Definição dos atributos
  - Identificação de métodos
  - Identificação de novos relacionamentos
  - Eventualmente, pode destacar relacionamentos e conceitos do modelo conceitual
- O projeto OO engloba outros modelos:
  - diagramas de interações (seqüência e colaboração), modelagem do comportamento, pacotes, modelo de implementação ...

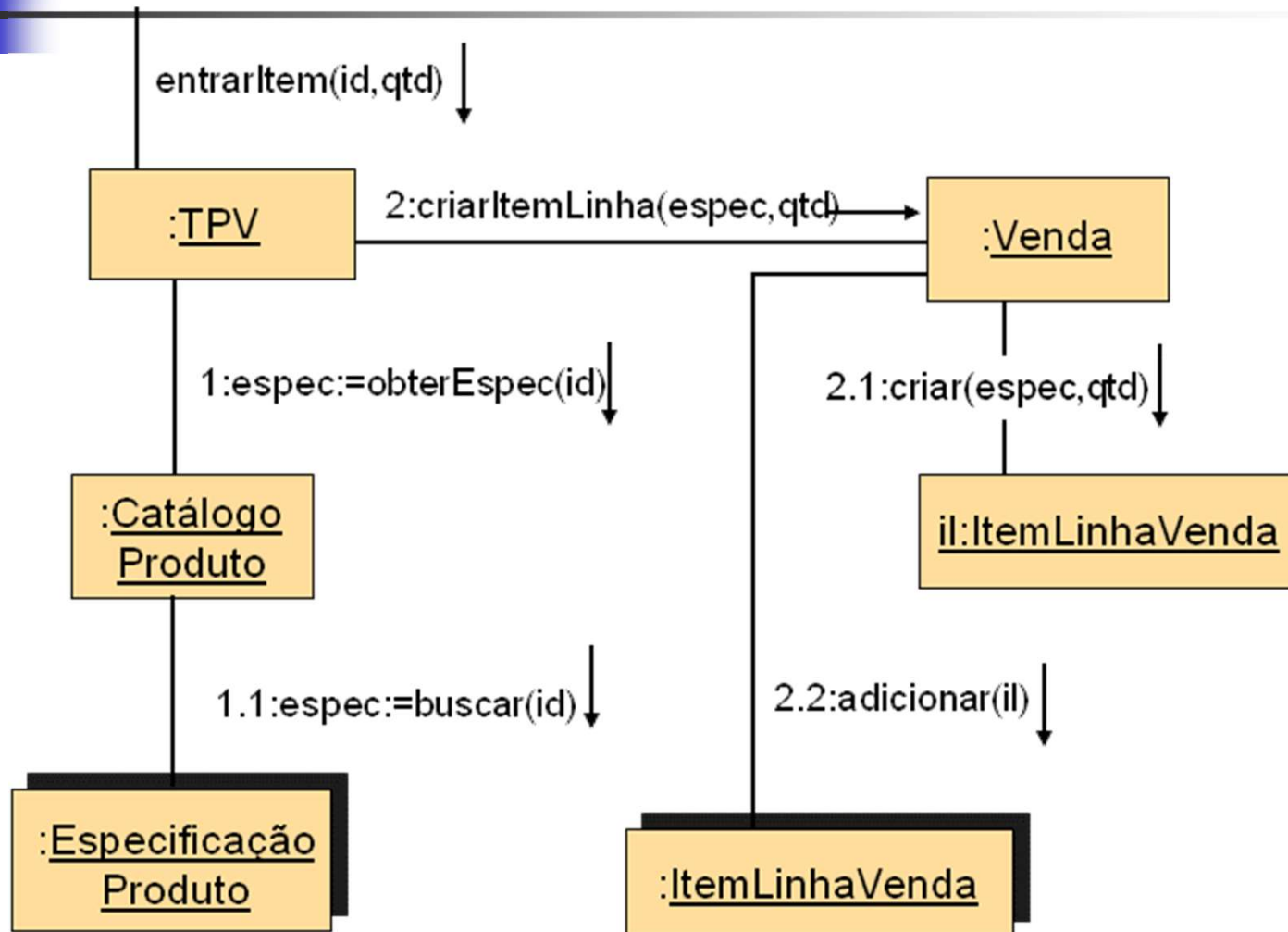


# *Projeto de Dados*

---

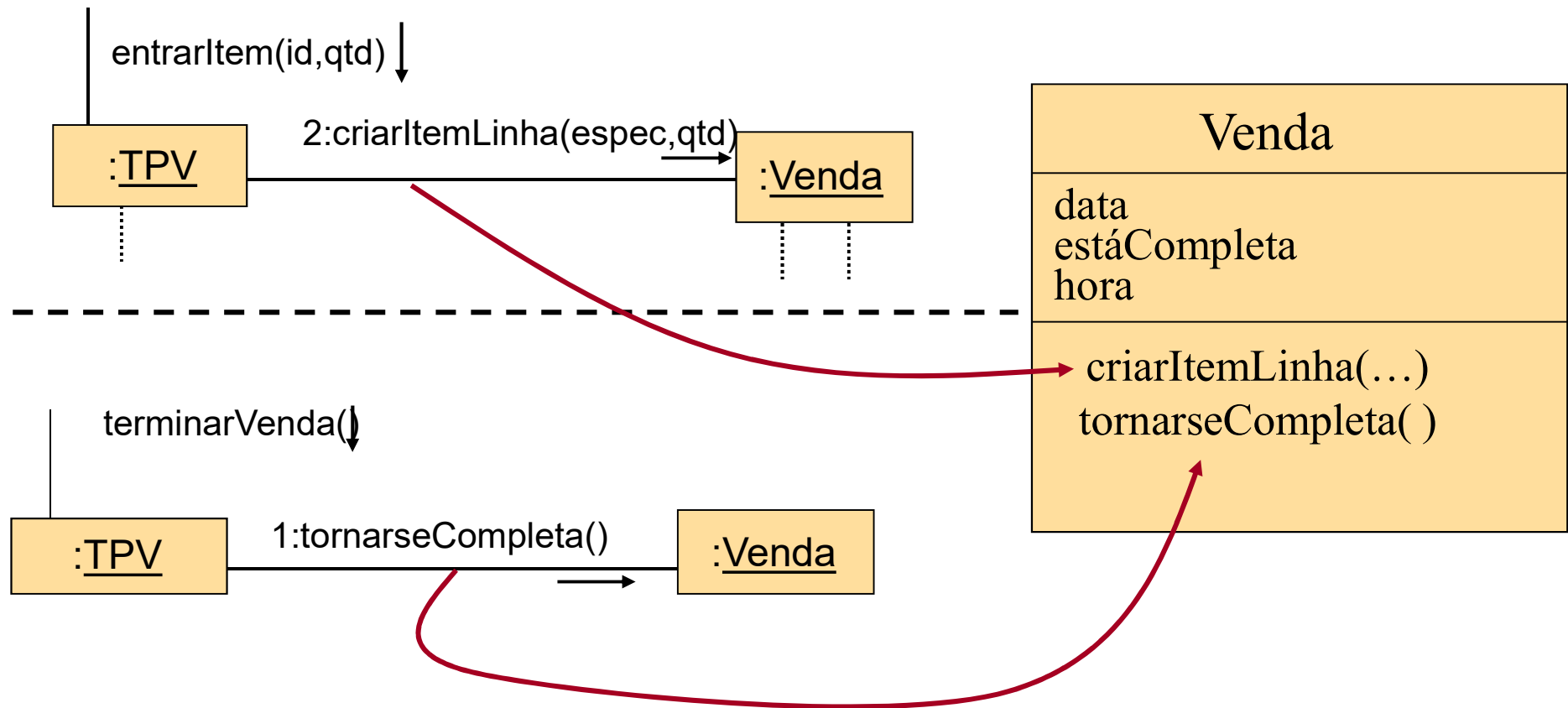
- Uma estratégia é usar informações de diagramas de seqüência e colaboração como base para definição de métodos, atributos e relacionamentos

# Exemplo Diagrama de Colaboração





# Definição de métodos





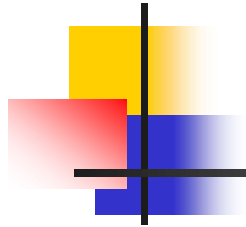
# ***Exemplo de Classe***

---

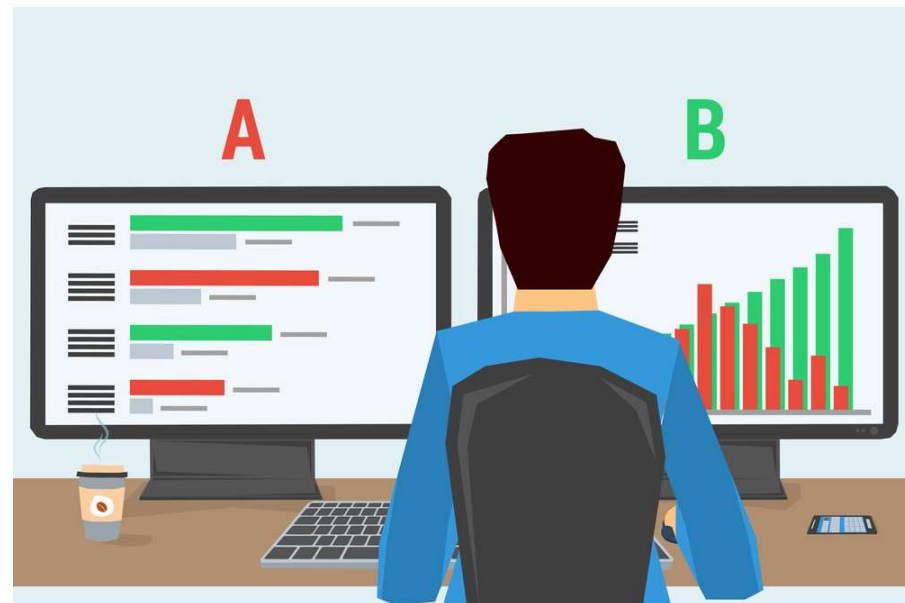
## Venda

data: **Data**  
estáCompleta: **Booleano**  
hora: **Hora**

criarItemLinha(espec:**EspecificacaoProduto**, qtd: **Inteiro**)  
tornarseCompleta( )  
obterTotal( ): **Moeda**  
fazerPagamento(quantia:**Moeda**)  
calcularTroco( ): **Moeda**



# ***PROJETO DE INTERFACE***





# *A Interface com o usuário*

---

- Projetadas para atender às habilidades, experiências e expectativas dos usuários previstos.
- Os usuários de sistemas freqüentemente julgam um sistema pela sua interface ao invés de sua funcionalidade.
- Uma interface fracamente projetada pode levar um usuário a cometer erros catastróficos.
- Projeto fraco de interface com usuário é a razão pela qual muitos sistemas de software nunca são usados.

# Fatores Humanos



- **Memória limitada de curto prazo**
  - Podemos lembrar instantaneamente de sete itens de informação.
- **As pessoas cometem erros**
  - alarmes e mensagens inapropriados podem aumentar o estresse e a probabilidade de mais erros.
- **As pessoas são diferentes**
  - capacidades físicas variadas: os projetos não podem ser desenvolvidos para as capacidades do desenvolvedor.
- **As pessoas têm preferências diferentes de interação**
  - Alguns gostam de figuras, outros, de textos





# *Princípios de Projeto de Interface de Usuário*

---

## 1. Familiaridade de usuário

- a interface deve usar termos e conceitos obtidos da experiência de pessoas que farão mais uso do sistema

## 2. Consistência

- operações comparáveis devem ser ativadas da mesma maneira

## 3. Surpresa mínima

- nunca surpreender o usuário com o funcionamento do sistema



# *Princípios de Projeto de Interface de Usuário*

---

## 4. Facilidade de recuperação

- mecanismos para recuperação de erros

## 5. Guia de usuário

- fornecer *feedback* significativo quando ocorrerem erros
- fornecer recursos sensíveis ao contexto para ajudar o usuário.

## 6. Diversidade de usuário

- interação adequada para tipos diferentes de usuários

# Projeto de IU

- *Como as informações do usuário devem ser fornecidas para o sistema de computador? (inputs)*
- *Como as informações do sistema de computador devem ser apresentadas para o usuário? (outputs)*







# ***Estilos de Interação (inputs)***

<b>Estilo de Interação</b>	<b>Vantagens</b>	<b>Desvantagens</b>	<b>Exemplos Aplicação</b>
<b>Manipulação Direta</b>	Interação rápida e intuitiva Facilidade de aprendizado	Pode ser difícil de implementar Adequado somente quando existe uma metáfora visual	Jogos de vídeo, interface de SO
<b>Seleção de Menu</b>	Evita erros do usuário Necessária pouca digitação	Lento para usuários experientes Pode tornar-se complexo se houver muitas opções	A maioria dos sistemas de propósito geral
<b>Preenchimento de formulários</b>	Entrada de dados simples Verificável Facilidade de aprendizado	Causa problemas quando as opções do usuário não combinam com os campos	Controle de estoque, Cadastros em sites na internet
<b>Linguagem de comando</b>	Poderosa e flexível	Difícil Pode levar a erros	SO, sistemas de comando e controle
<b>Linguagem natural</b>	Acessível a usuários específicos	Mais complexos para desenvolver	Consulta a informações e baseados em conhecimento, voz

# ***Apresentação de Informações (outputs)***



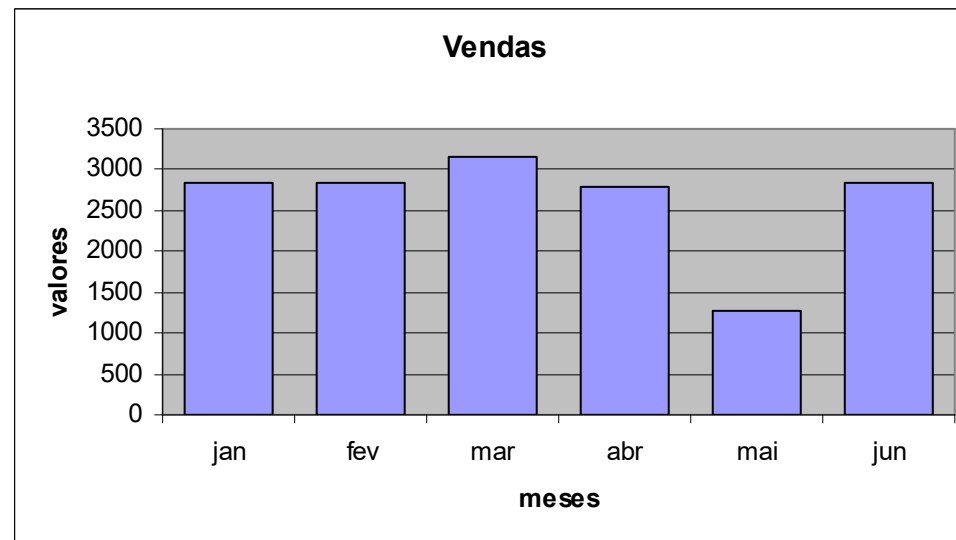
---

- diretamente
  - texto
  
- transformadas
  - tabelas, gráficos, formulários...

# Apresentação de Informações (outputs)

- O usuário está interessado em informações precisas ou nos relacionamentos de dados?

jan	fev	mar	abr	mai	jun
2.842	2.851	3.164	2.789	1.273	2.835



# Apresentação de Informações (outputs)

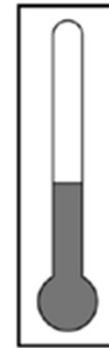
- Com que rapidez mudam os valores de informações? A mudança deve ser indicada imediatamente?



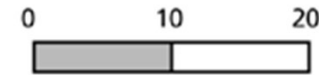
Mostrador com ponteiro



Gráfico de setores



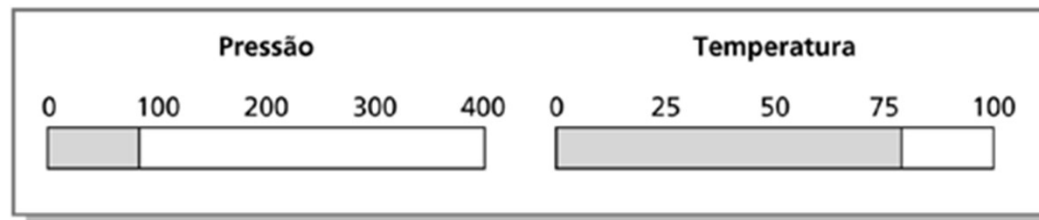
Termômetro



Barra horizontal

# Apresentação de Informações (*outputs*)

- As informações são textuais ou numéricas? Os valores relativos são importantes?





# *Visualização de Dados*

---

- Apresentação de grandes quantidades de informações.
  - A visualização pode revelar o relacionamento entre entidades e tendências nos dados.
- Exemplos:
  - **Informações meteorológicas** - mapa meteorológico;
  - **Rede telefônica** - conjunto de nós ligados e um centro de gerenciamento de redes;
  - **Indústria química** - visualizado ao mostrar pressões e temperaturas em um conjunto de tanques e tubulações ligadas;
  - O modelo de uma **molécula** é exibido em três dimensões;



# *Estilo de Apresentação de Informações - Cores*

---

- A cor adiciona uma dimensão extra à uma interface, e pode auxiliar o usuário na compreensão das estruturas complexas de informações.
- Uma cor pode ser usada para destacar eventos excepcionais.



# *Estilo de Apresentação de Informações - Cores*

---

- Diretrizes:
  - Limitar o número de cores usadas e ser conservativo quanto ao modo de serem utilizadas.
  - Usar a mudança de cores para mostrar uma mudança no status de sistema.
  - Usar codificação de cores para apoiar a tarefa que os usuários estão tentando executar.
  - Usar codificação de cores de uma maneira cuidadosa e consistente.
  - Ser cuidadoso quanto aos pares de cores.



# *Estilo de Apresentação de Informações - Cores*



- O olho humano não consegue visualizar apropriadamente algumas combinações de cores!!!
- O olho humano não consegue visualizar apropriadamente algumas combinações de cores!!!

# Estilo de Apresentação de Informações

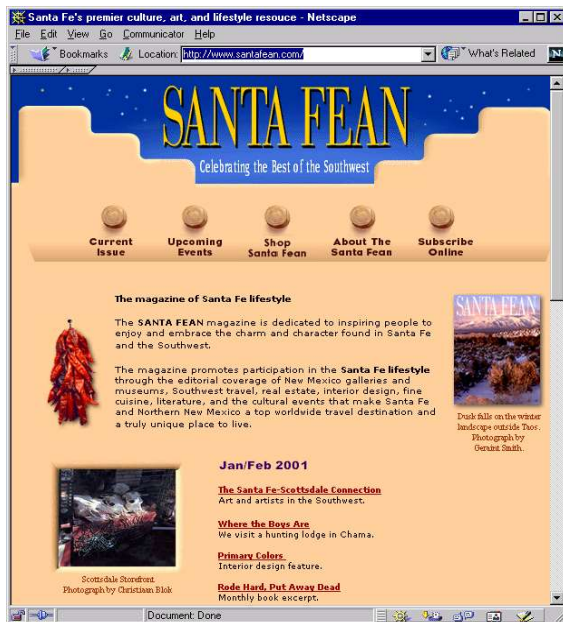
- Clareza é fundamental!



The screenshot shows the Apple website interface in a browser window. The address bar displays "http://www.apple.com/". The navigation menu includes links for Store, Switch, .Mac, QuickTime, Support, and Mac OS X, with sub-links for Hot News, Hardware, Software, Made4Mac, Education, Creative, SmallBiz, Developer, and Where to Buy. The main content area features a testimonial from Gautam Godse, a software projects manager, who states, "Nobody does anything like iPhoto in the PC world. That's why I switched to Apple." Below the testimonial is a "Hot News Headlines" section with the headline "AppleScript Studio makes life easier," says veteran developer." The product highlights section includes iCal (Organize your life. Download now.), Jaguar (Mac OS X Version 10.2 Now available.), Power Mac G4 (All dual processors starting at just \$1,699. Dual 1.25GHz now shipping.), and iPod (5GB, 10GB, 20GB). The footer contains a search bar, a search button, and links for Site Map, Search Tips, and Job Opportunities.

# Estilo de Apresentação de Informações

- Consistência remete a controle, qualidade e confiança



Home page



Content page 1



Content page 2

# *Estilo de Apresentação de Informações*

- Distribuição das informações

Name   
Addr1   
Addr2   
City   
State   
Phone   
Fax

Name	<input type="text"/>
Addr1	<input type="text"/>
Addr2	<input type="text"/>
City	<input type="text"/>
State	<input type="text"/>
Phone	<input type="text"/>
Fax	<input type="text"/>

Name   
Addr1   
Addr2   
City   
State   
Phone   
Fax

# Estilo de Apresentação de Informações

Employee First name - Netscape 6

File Edit View Search Go Bookmarks Tasks Help

Home My Netscape Net2Phone Instant Message WebMail Radio

\* First name:

\* Last name:

\* Start date: 2001 / Mar / 06

Gender:  Male  Female

Birth date: 1970 / Jan / 01

Home phone:

Address:

Street:

City:

Zip code:

State: CA

\* = Required field

Employee First name - Netscape 6

File Edit View Search Go Bookmarks Tasks Help

Home My Netscape Net2Phone Instant Message WebMail Radio

\* First name:

\* Last name:

\* Start date: 2001 / Mar / 06

Gender:  Male  Female

Birth date: 1970 / Jan / 01

Home phone:  e.g. (905) 555-1212

Mailing Address:

Street:

City:

Zip code:  e.g. 90210-1701

State: CA

\* = Required field



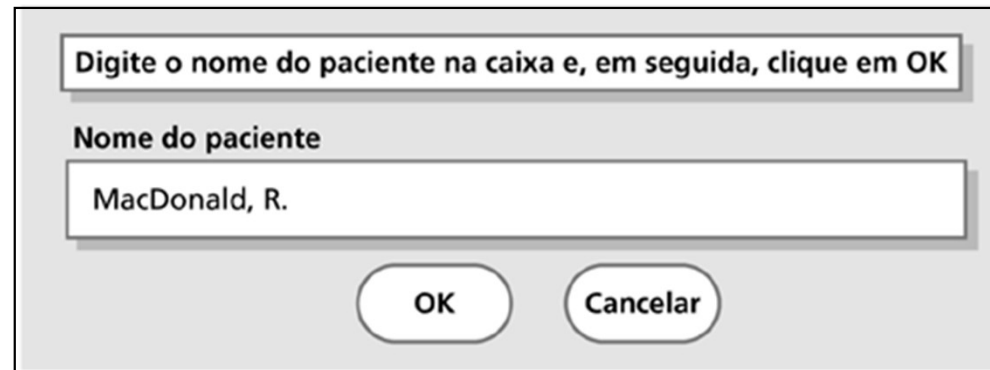
## ***Estilo de Apresentação de Informações - mensagens de erro***

---

- Mensagens devem ser gentis, concisas, consistentes e construtivas.
- O contexto e a experiência dos usuários devem ser os fatores determinantes para os projetos de mensagens.

# Estilo de Apresentação de Informações - mensagens de erro

Caixa de entrada de texto usada por uma enfermeira.



Digite o nome do paciente na caixa e, em seguida, clique em OK

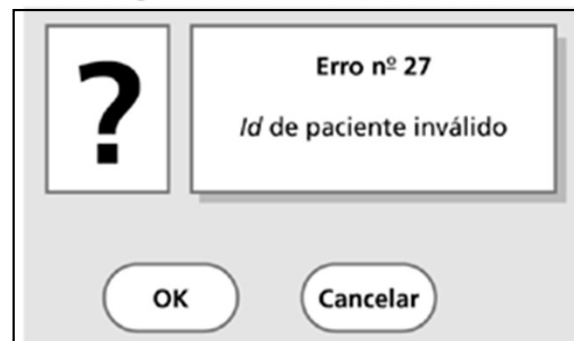
Nome do paciente

MacDonald, R.

OK Cancelar

Mensagens de erro orientada a sistemas e orientada a usuários.

Mensagem de erro orientada a sistemas



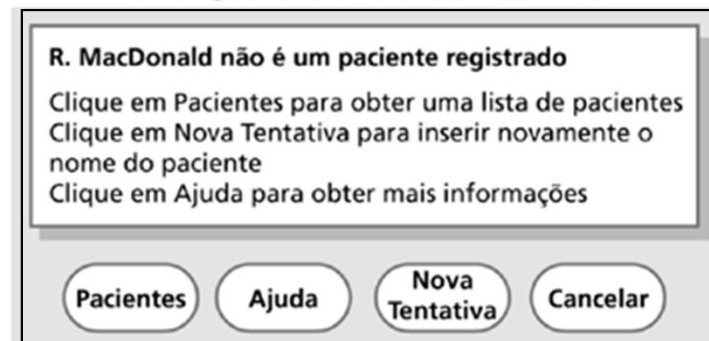
**?**

**Erro nº 27**

*Id de paciente inválido*

OK Cancelar

Mensagem de erro orientada a usuários

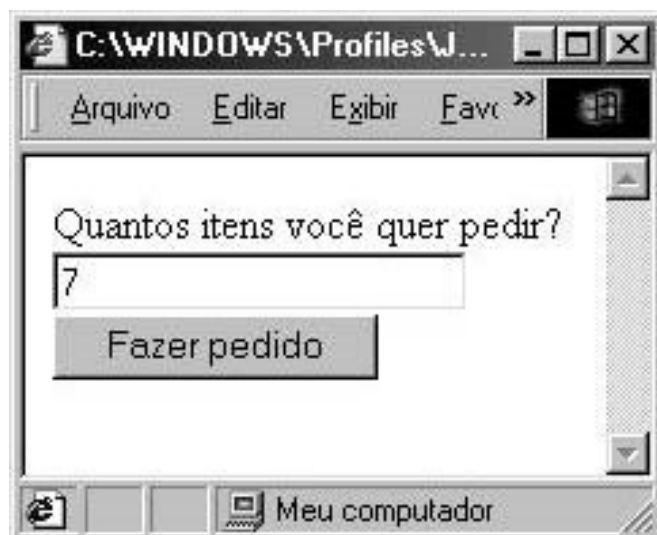
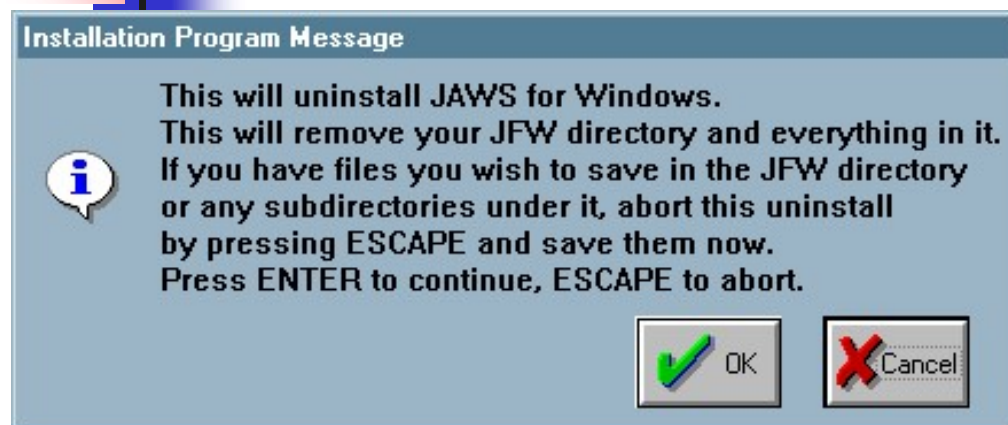


**R. MacDonald não é um paciente registrado**

Clique em Pacientes para obter uma lista de pacientes  
Clique em Nova Tentativa para inserir novamente o nome do paciente  
Clique em Ajuda para obter mais informações

Pacientes Ajuda Nova Tentativa Cancelar

# Estilo de Apresentação de Informações - mensagens de erro







## *Estilo de Apresentação de Informações - mensagens de erro*

---

- Nem sempre a mensagem é totalmente lida...
  - não apresenta em qualquer ordem as letras de uma palavra e, portanto, a única característica é que a primeira e última letras estejam no lugar certo. O resto pode ser uma qualquer palavra que você pode ainda ler sem problema. Isso é porque nós não temos cada letra isolada, mas a palavra como um todo. Vdaerde!

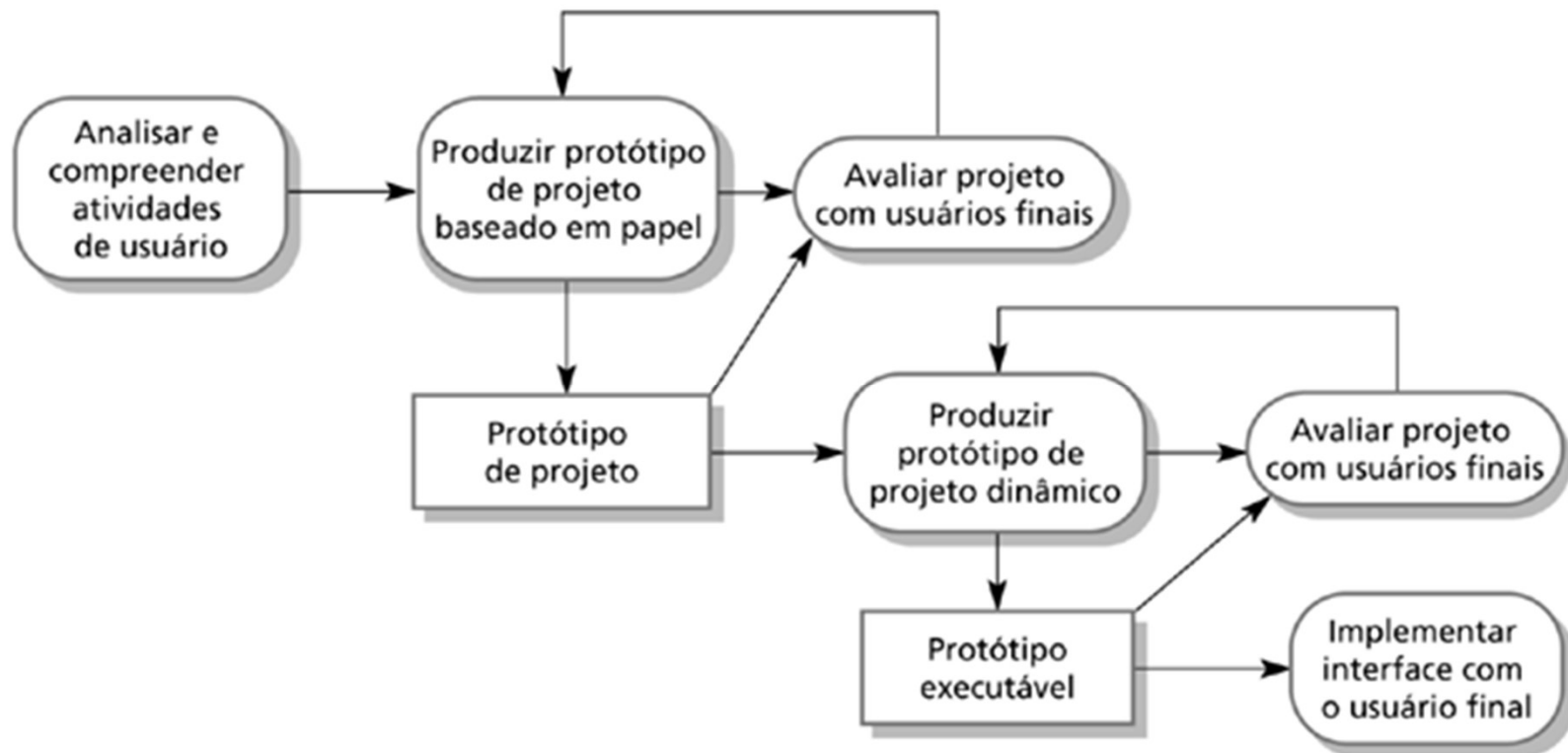


# *Processo de Projeto de IU*

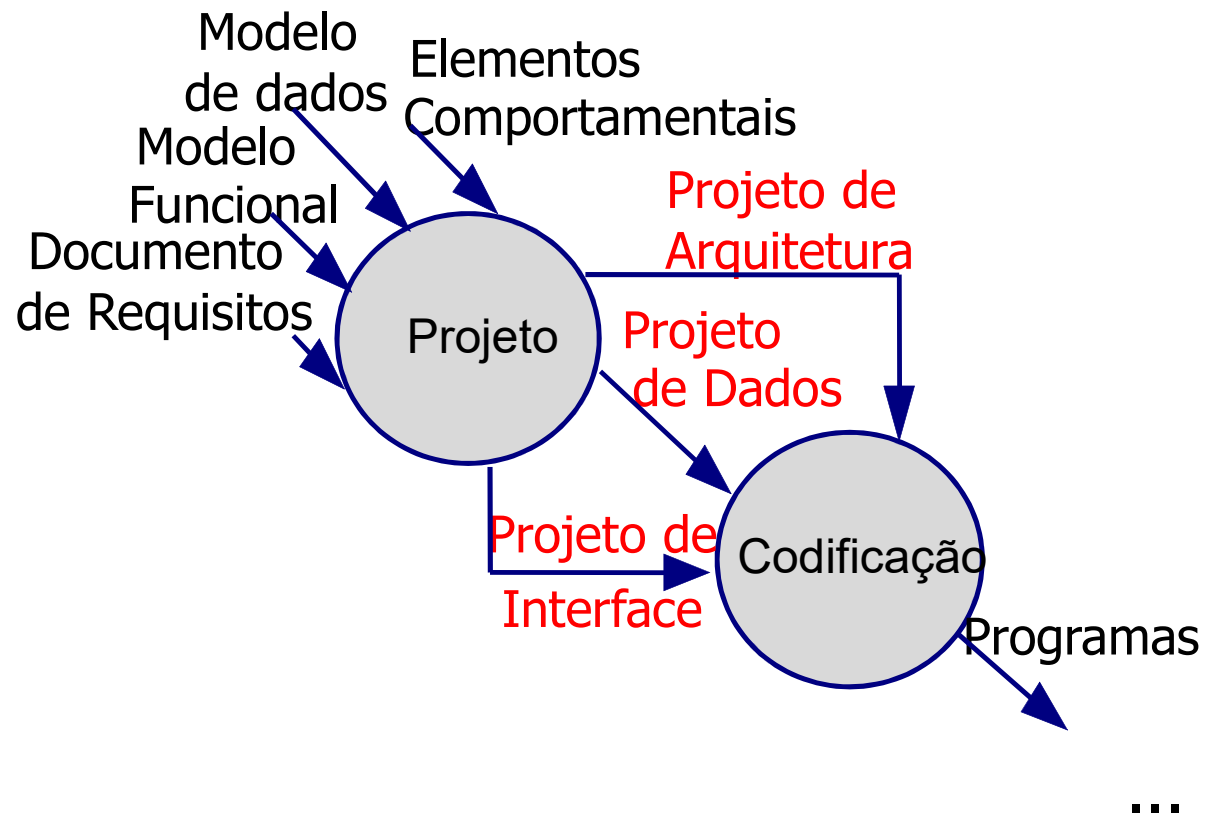
---

- O projeto de IU é um processo iterativo que envolve estreita cooperação entre usuários e projetistas.
- As três atividades centrais nesse processo são:
  - **Análise de usuário.** Compreender o que os usuários farão com o sistema;
  - **Prototipação de sistema.** Desenvolver uma série de protótipos para experimento;
  - **Avaliação de interface.** Experimentar esses protótipos com usuários.

# Processo de Projeto de IU



# Projeto de Software



# Projeto de Software

Engenharia de Software

*Simone do Rocio Senger de Souza*  
*srocio@icmc.usp.br*

