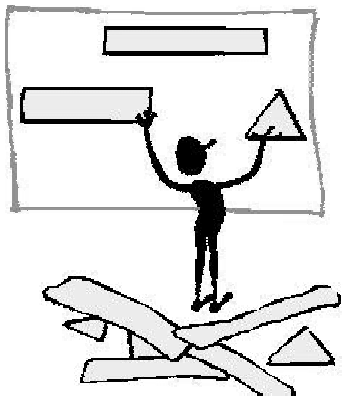


Engenharia de Software

Tema da Aula

Projeto de Software

Prof. Cristiano R R Portella
portella@widesoft.com.br

PUC CAMPINAS
Engenharia de Software

Projeto (Design) de Software

Projetar Software é o processo de aplicar várias técnicas e princípios com o propósito de se definir um dispositivo, processo ou sistema, com detalhes suficientes para permitir sua realização física. (Taylor-59).

O Projeto de software é o núcleo técnico da Engenharia de Software. É a única maneira de se traduzir "com precisão", os requisitos do usuário para um produto ou sistema acabado.

Meta:

Traduzir requisitos numa representação de software.

PUC CAMPINAS
Engenharia de Software

Projeto (Design) de Software
Princípios

Desenvolver um projeto de software é um processo que combina:

- Instituição de critérios baseados na experiência adquirida na construção de entidades similares.
- Um conjunto de princípios e/ou heurísticas que guiam o desenvolvimento do modelo.
- Um conjunto de critérios que facilitam a verificação da qualidade.
- Um processo de iteração que conduz a uma representação do projeto final

PUC CAMPINAS
Engenharia de Software

Projeto (Design) de Software
Diferentes Visões

- ✓ Projeto Procedimental: descrição da funcionalidade do software (algoritmos).
- ✓ Projeto de Dados: definição das estruturas de dados.
- ✓ Projeto das Interfaces: Layouts e mecanismos de interação homem-máquina (se necessário).
- ✓ Projeto Arquitetural: associação entre os principais elementos estruturais do software (árvore dos módulos, mensagens entre objetos, Nivelamento em Camadas).

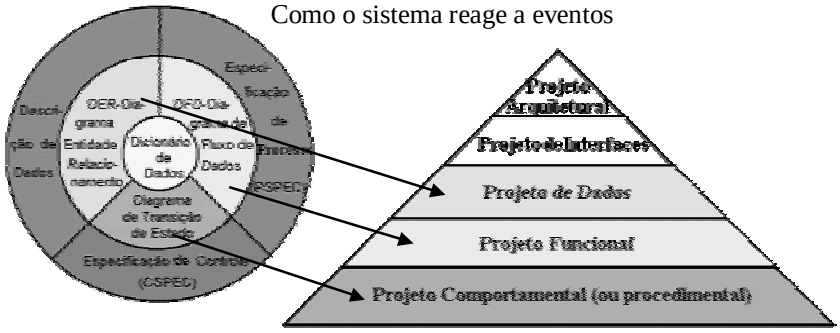
PUC CAMPINAS
Engenharia de Software

Projeto (Design) de Software
Transição Análise -> Projeto

Dados tratados pelo sistema

Como os dados são tratados

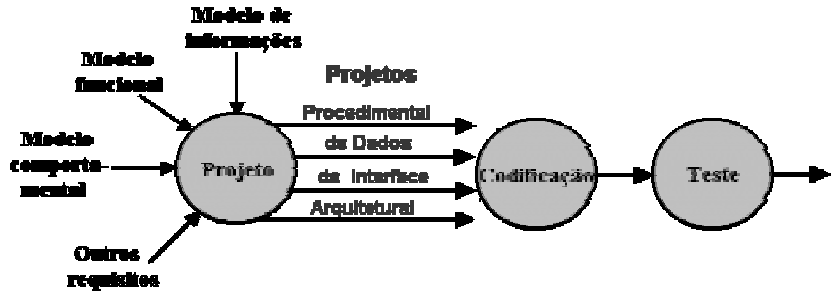
Como o sistema reage a eventos



PUC CAMPINAS
Engenharia de Software

Projeto (Design) de Software
Modelo Clássico (Cascata)

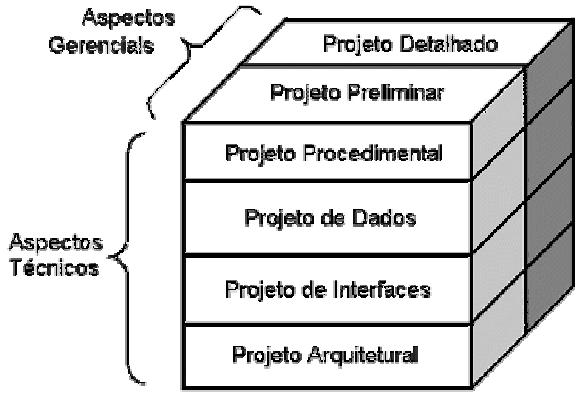
Projeto de Software:



PUC CAMPINAS
Engenharia de Software

Projeto (Design) de Software
Modelo Clássico (Cascata)

Projeto de Software:



PUC CAMPINAS
Engenharia de Software

Processo (Design) de Software

- ✓ Projeto Preliminar

  Transformação dos requisitos em modelos de arquitetura, dados e procedimentos.

- ✓ Projeto Detalhado

  Refinamento da representação da arquitetura, dos dados e dos procedimentos, gerando estruturas mais refinadas (detalhadas).

PUC CAMPINAS
Engenharia de Software

Qualidade do Projeto de Software

- ✓ A Qualidade do Projeto é avaliada através de revisões técnicas formais (walkthrougs de projetos), usando-se os seguintes critérios de referência:

  1. Organização hierárquica: através do uso inteligente de controle entre os elementos do software;
  2. Modularidade: particionamento lógico em elementos que executam funções e subfunções específicas;

PUC CAMPINAS
Engenharia de Software

Qualidade do Projeto de Software

- ✓ A Qualidade do Projeto é avaliada ...

  3. Representações distintas para dados e procedimentos: mesmo que sejam posteriormente agrupados em objetos;
  4. Deve levar a Módulos ou Classes de objetos que apresentam características funcionais independentes;
  5. Deve levar à interfaces que reduzam a complexidade de conexões entre os módulos e com o ambiente externo; e

PUC CAMPINAS
Engenharia de Software

Fundamentos do Projeto de Software

Fundamentos:

1. Abstração: Concentrar-se no problema com um certo nível de generalização.

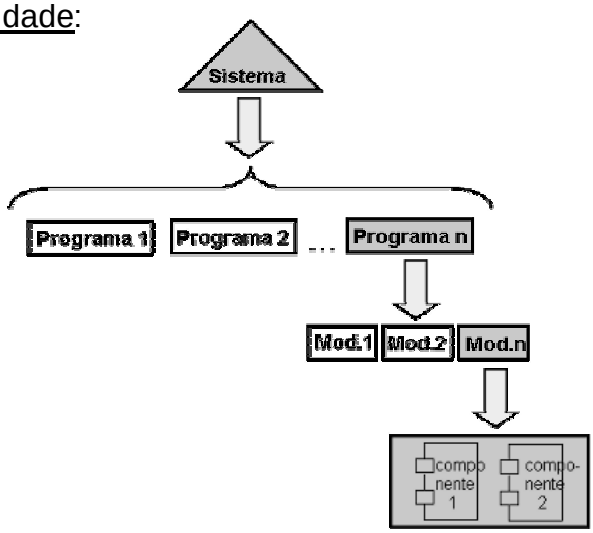
   Abstração procedimental
   Abstração de dados
   ⇨ { Abstração do Projeto e de seu ambiente

2. Refinamento sucessivo: É um processo de elaboração que parte de uma declaração de função, que será elaborada através de sucessivos refinamentos, cada um incorporando mais detalhes.

PUC CAMPINAS
Engenharia de Software

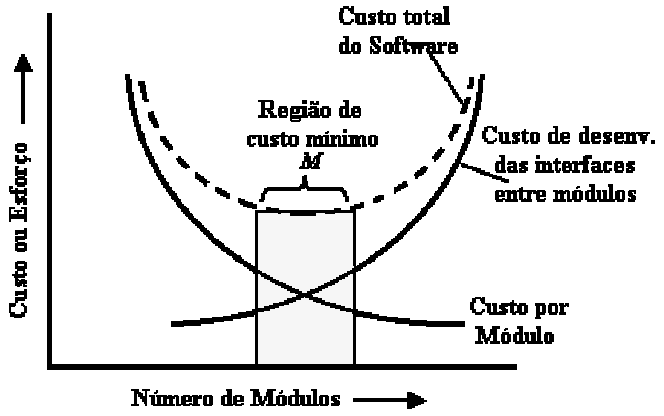
Fundamentos do Projeto de Software

3. Modularidade:



PUC CAMPINAS
Engenharia de Software

Projeto de Software



PUC CAMPINAS
Engenharia de Software

Fundamentos do Projeto de Software

4. Arquitetura de Software:
   - Estrutura hierárquica de componentes procedimentais e
   - Estrutura de dados

5. Hierarquia de Controle:

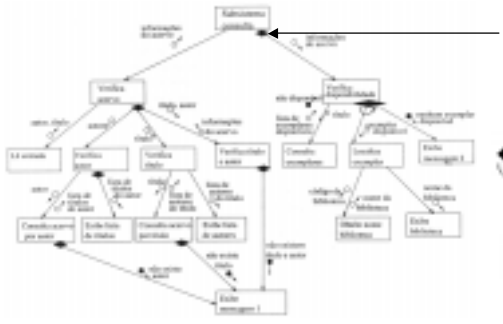
   Representa a organização de componentes de um programa e a hierarquia de controle entre seus módulos.

PUC CAMPINAS
Engenharia de Software

Projeto Procedimental de Software
Diagrama Estrutura - Hierarquia dos Módulos

PUC CAMPINAS
Engenharia de Software

Projeto Procedimental de Software
Diagrama Estrutura - Hierarquia dos Módulos



- ◇ Decisão: o módulo abaixo pode ou não receber o controle (decisão).
- ◆ Chamada repetida (iterativa)
- ○ Passagem de dados entre módulos
- ● Passagem de controle entre módulos

Projeto de Dados de Software

Projeto de Dados

PUC CAMPINAS
Engenharia de Software

Projeto de Dados de Software

Representação do relacionamento lógico entre elementos de dados individuais. Determina a organização, métodos de acesso, associações e alternativas de processamento.

- ✓ Parte dos dados levantados e representados na fase de análise:
  - Dicionário de Dados
  - Fluxo, Conteúdo e Estrutura

PUC CAMPINAS
Engenharia de Software

Projeto de Dados de Software

Atividades do Projeto de Dados:

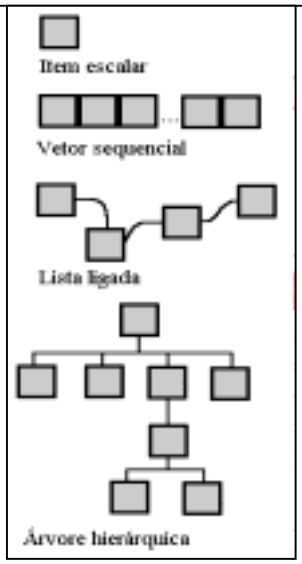
- ✓ Selecionar representações de dados;
- ✓ Estudar e escolher estruturas de dados que permitam a implementação mais adequada;
- ✓ Caracterizar a Abrangência (escopo) dos Dados
  - Local (componente), Partes do software ou Global
  - Caracterizas a Persistência dos Dados
    - Persistentes (B.Dados)
    - Não Persistentes (Dados em memória, estruturas de manipulação/intermediária)..

PUC CAMPINAS
Engenharia de Software

Projeto de Dados de Software

Estruturas/Elementos de Dados Comuns:

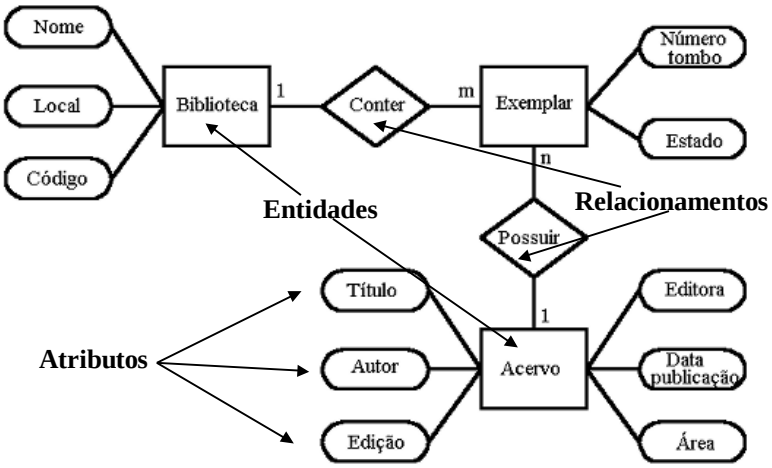
- ✓ Itens Elementar (Tipos Primitivos)
- ✓ Listas Lineares
  - Gerais
  - Pilhas
  - Filas
- ✓ Lista não lineares
  - Árvores
  - Grafo



PUC CAMPINAS
Engenharia de Software

Projeto de Dados de Software
Dados Persistentes

Modelo de Entidade-Relacionamento (MER)



PUC CAMPINAS
Engenharia de Software

Projeto de Software

Projeto Arquitetural de Software

PUC CAMPINAS
Engenharia de Software

Projeto Arquitetural de Software

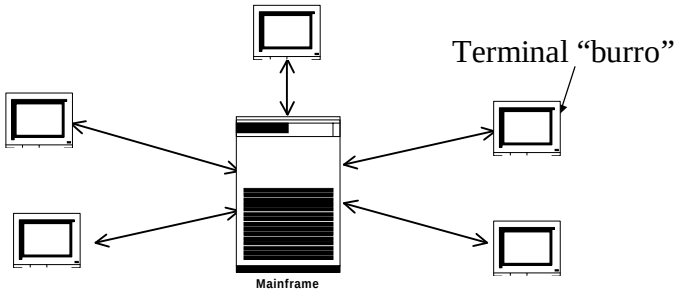
Visa modelar a estrutura completa do software e as maneiras fornecidas para manter a integridade conceitual de um sistema:

Arquiteturas Tradicionais:

- Centralizada
- Parcialmente Distribuída
- Cliente-Servidor (2 Camadas)
- Cliente-Servidor (3 Camadas)
- Distribuída (Multi-Camadas)

PUC CAMPINAS
Engenharia de Software

Projeto Arquitetural de Software
Arquitetura Centralizada

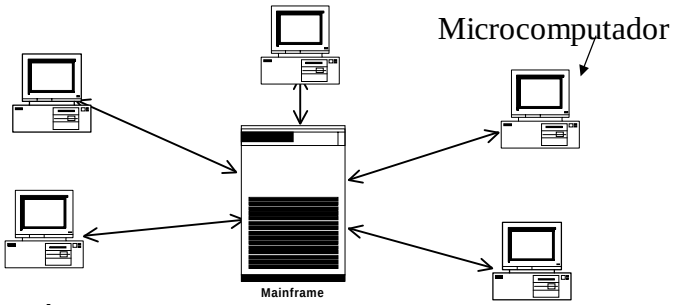


✓ Características:

- Processamento centralizado no Mainframe;
- Terminais Burros (sem processamento);
- Redes Corporativas;
- Software uso Departamental (Baixa Integração).

PUC CAMPINAS
Engenharia de Software

Projeto Arquitetural de Software
Arquitetura Parcialmente Distribuída

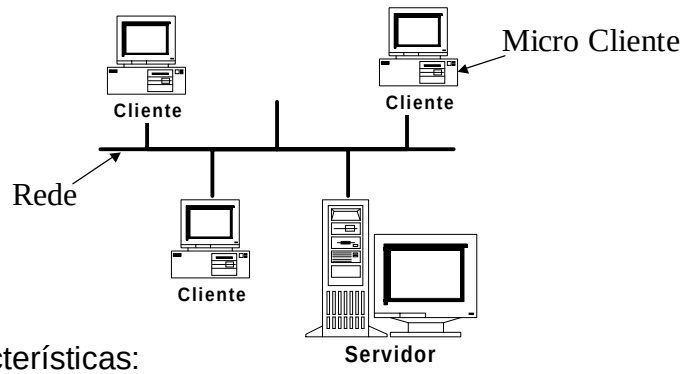


Características:

- Um pouco de processamento departamental;
- Micros com algumas aplicações locais;
- Rede Corporativa conectando micro-mainframe;
- Software Departamental com maior Integração;
- Início de Integração entre parceiros (EDI).

PUC CAMPINAS
Engenharia de Software

Projeto Arquitetural de Software
Cliente-Servidor (2 camadas)

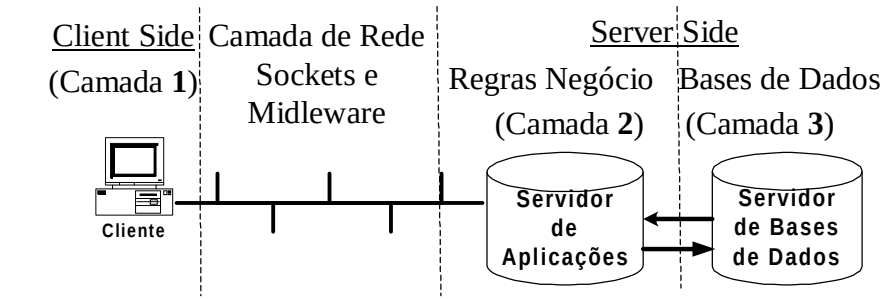


Características:

- Boa parte do processamento local
- Micros com algumas aplicações locais
- Redes LAN's ou WAN's
- Integração entre Parceiros (cliente-servidor)

PUC CAMPINAS
Engenharia de Software

Projeto Arquitetural de Software
Cliente-Servidor 3 camadas

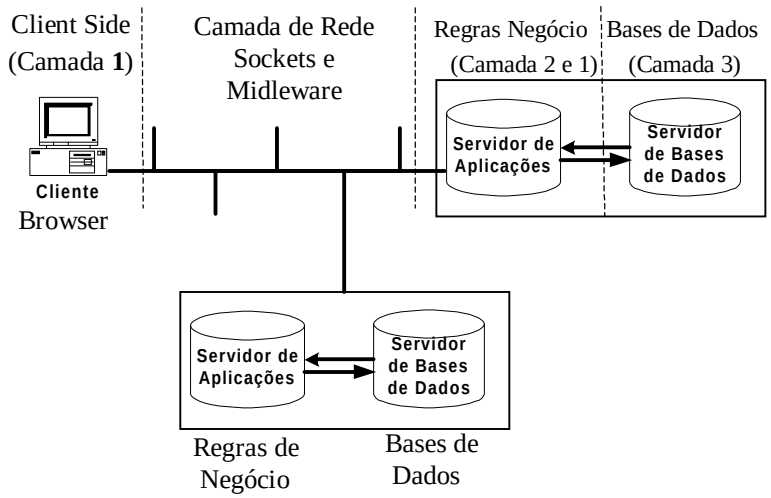


Características:

- Mais capacidade de Processamento Distribuído
- Internet/Intranet como infra-estrutura de rede
- 2 Níveis de Servidor
- Internet Aplications / Software Integrados(ERPs)

PUC CAMPINAS
Engenharia de Software

Projeto Arquitetural de Software
Arquitetura Distribuída Multi-Camadas



PUC CAMPINAS
Engenharia de Software

Projeto Arquitetural de Software
Arquitetura Distribuída Multi-Camadas

- ✓ Características:
  - Alta Capacidade de Processamento Distribuído
  - Internet/Intranet/Extranet/VPNs como infra-estrutura de rede
  - Distribuição dos Serviços em várias camadas de servidores de aplicação e dados
  - Internet Aplications de última geração, CRM (Relacionamento com Cliente), SCM (Cadeias de Produção e Distribuição Integradas), Bancos de Dados Distribuídos

PUC CAMPINAS
Engenharia de Software

Projeto de Software

Projeto Procedimental

PUC CAMPINAS
Engenharia de Software

Projeto Procedimental de Software

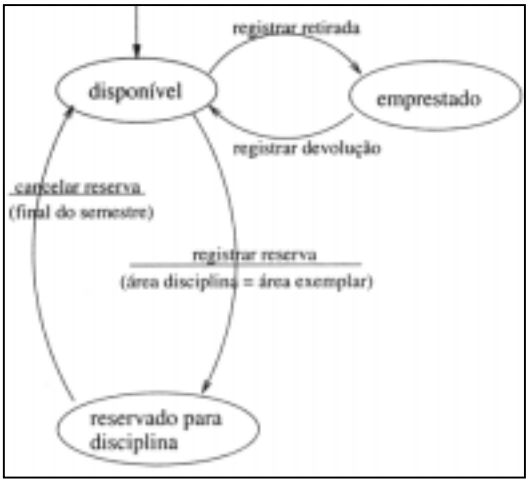
Finaliza os detalhes de processamento (procedimentos) de cada módulo. O procedimento deve oferecer uma especificação precisa do processamento, inclusive a seqüência de eventos, operações repetitivas etc.

Baseia-se na Especificação dos requisitos, na modelagem (DFD, Diagrama Classes) e no Dicionário de Dados, obtidos na análise.

PUC CAMPINAS
Engenharia de Software

Projeto Procedimental de Software
Modelo Comportamental

Diagrama de Estados dos Exemplares



PUC CAMPINAS
Engenharia de Software

Projeto Procedimental de Software

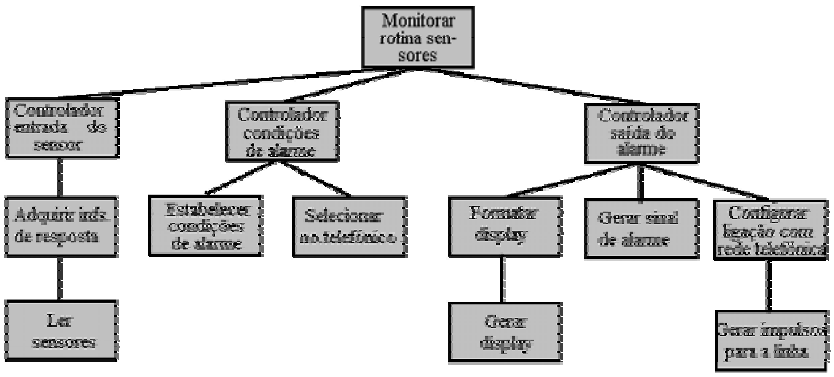
Seqüência:

1. Decomposição, Identificação e Modelagem do software através de Componentes (Módulos ou Classes)
2. Representação da estrutura de controle e interação entre os componentes
3. Revisão e Refinamento da Estrutura
4. Representação dos detalhes algorítmicos

PUC CAMPINAS
Engenharia de Software

Projeto Procedimental de Software

Estrutura do módulo "Monitorar Sensores"



PUC CAMPINAS
Engenharia de Software

Projeto Procedimental de Software

Características Importantes:

- Coesão
- Acoplamento
- Independência entre os Componentes

Técnicas de Representação:

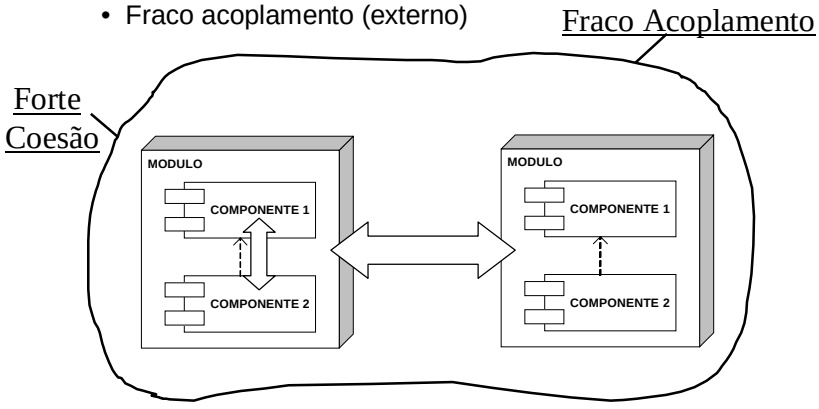
- PDL (Program Design Language) ou Português Estruturado
- Tabelas/Árvores de Decisão
- Fluxogramas
- Diagramas de quadros de Nassi e Schneiderman
- Diagrama de Chapin

PUC CAMPINAS
Engenharia de Software

Fundamentos do Projeto de Software

Relação entre componentes:

- Forte coesão (interna)
- Fraco acoplamento (externo)



Projeto de Interface

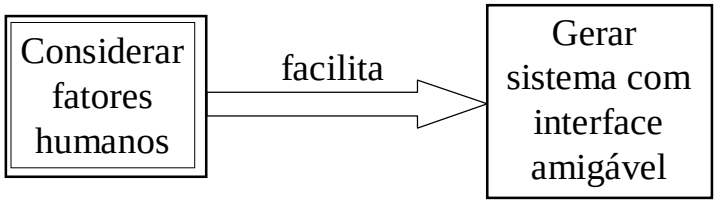
Projeto de Interface de Software

O projeto de interface com o usuário tem pontos em comum com o estudo das questões tecnológicas e pontos em comum com o estudo das pessoas (Psicologia Cognitiva):

- ✓ Percepção Visual;
- ✓ Psicologia Cognitiva de leitura;
- ✓ Memória Humana;
- ✓ Raciocínio Indutivo e Dedutivo;
- ✓ Comunicação Textual ou Pictórica (ícones)
- ✓ Nível Intelectual e/ou Habilidades do Usuário.

PUC CAMPINAS
Engenharia de Software

Projeto de Interface de Software



PUC CAMPINAS
Engenharia de Software

Projeto de Interface de Software

Podemos classificar os usuários, quanto a sua exposição ao sistema, como:

- ✓ Principiantes:
  Sem conhecimento de interação com o sistema e pouco conhecimento das funções que são executadas.
- ✓ Usuários treinados e freqüentes:
  Bom conhecimento da interação e das funções do sistema. Gosta de atalhos e modos abreviados de interação.
- ✓ Usuários treinados e intermitentes:
  Razoável conhecimento das funções porém pouca lembrança da interação com o sistema (com usar a interface).

Projeto de Interface de Software

Boas práticas em Interfaces:

- ✓ Facilidade de ajuda integrada:
  Projetada desde o início do sistema. Sensível ao contexto (Exemplo: Assistente do Office).
- ✓ Facilidade de ajuda Add-On:
  Adicionada após o produto estar construído, para atualiza-lo (Exemplo: Help do Delphi).
- ✓ Mensagens de Erro:
  Evitar mensagens lacônicas ou que apenas o remetem ao Manual ou ao Suporte.

PUC CAMPINAS
Engenharia de Software

Projeto de Interface de Software

PUC CAMPINAS
Engenharia de Software

Projeto de Interface de Software

Boas práticas em Interfaces:

- ✓ Mensagens de Erro:
  - Descrever o problema em linguagem que o usuário entenda;
  - Orientar para o procedimento de recuperação a ser executado;
  - Alertar para as conseqüências do erro;
  - Usar sinal visual (cores) e sonoro;
  - Usar linguagem neutra e profissional.
  - Manter mensagem na tela até que seja reconhecida

PUC CAMPINAS
Engenharia de Software

Projeto de Interface de Software

Boas práticas em Interfaces:

- ✓ Evitar nomes de campos com abreviaturas e siglas;
- ✓ Colocar totalização nos campos adequados;
- ✓ Evitar a necessidade de memorizar códigos;
- ✓ Evitar que o usuário precise lembrar o próximo código (use tipo auto-incremental);
- ✓ Caso o cadastro não exista, abrir automaticamente o formulário de cadastramento (evitar “passeios” pelos menus);
- ✓ Se possível, criar opção de desfazer (UNDO) as ações executadas;

PUC CAMPINAS
Engenharia de Software

Projeto de Interface de Software

Boas práticas em Interfaces:

- ✓ Colocar no sistema uma ajuda para a seqüência de rotinas de atualização (o que já foi executado – próximo passo).
- ✓ Procurar evitar o erro antes que ele aconteça, ao invés de priorizar o tratamento de erros depois que ele já ocorreu;
- ✓ Mensagens com gravidade não podem ser reconhecidas apenas teclando [ENTER].
- ✓ Criar padrões para interfaces. O usuário aprende mais rápido (intuitivo), usa o aprendizado anterior e sente-se mais seguro na utilização.

PUC CAMPINAS
Engenharia de Software

Projeto de Interface de Software

Boas práticas em Interfaces:

- ✓ Na medida do possível, flexibilizar as interfaces, pensando que cada usuário tem um perfil cognitivo pessoal, um nível de conhecimento do sistema e uma necessidade operacional.
- ✓ O sistema deve fazer “tudo” que ele pode fazer, deixando para o usuário apenas as tarefas que o sistema não pode fazer.

PUC CAMPINAS
Engenharia de Software

Projeto de Interface de Software

As tarefas a serem realizadas pela interface, são classificadas como:

- ✓ Tarefas de Comunicação:
  Transmitir informações.
- ✓ Tarefas de Diálogo:
  Usuário interage com o sistema.
- ✓ Tarefas Cognitivas:
  Atividades associadas a funções do sistema e que são executadas assim que as informações são obtidas.
- ✓ Tarefas de Controle:
  Permite que o usuário controle as informações e a funcionalidade do sistema.

PUC CAMPINAS
Engenharia de Software

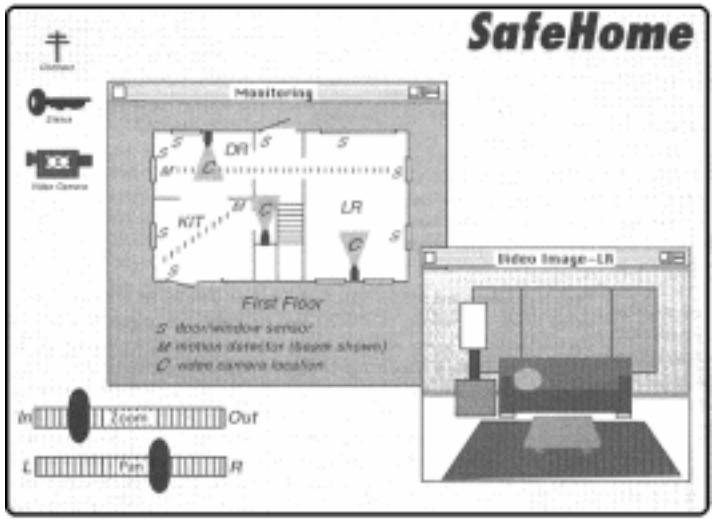
Projeto de Interface de Software

- ✓ Estilos de Interface Humano-Computador
  - Painéis e Botões. Linhas de Comando. Menus.
  - Interface “Point-and-Pick” (aponte e “pegue”)
    - Janelas
    - Menus Pop-UP e PULL-Down
    - Ícones
    - Quadros, Barras de Rolagem, etc...
  - Reconhecimento
    - Comandos de voz, Reconhecimento Gráfico, leitura de íris, leitura de digital etc...
  - Realidade Virtual

Projeto de Interface de Software

PUC CAMPINAS
Engenharia de Software

Projeto de Interface de Software



PUC CAMPINAS
Engenharia de Software

Etapas de um Projeto de Software

Seqüência:

- ✓ Projetar Dados e Procedimentos
  - Orientado a Fluxo de Dados
  - Orientado a Objetos
  - Orientado a Estrutura de Dados
- ✓ Definir as características gerais da Interface do Software, em função do tipo de usuário, da arquitetura que foi projetada, etc..
- ✓ Definir e Projetar a Arquitetura do Software
- ✓ Revisão Geral dos Projetos

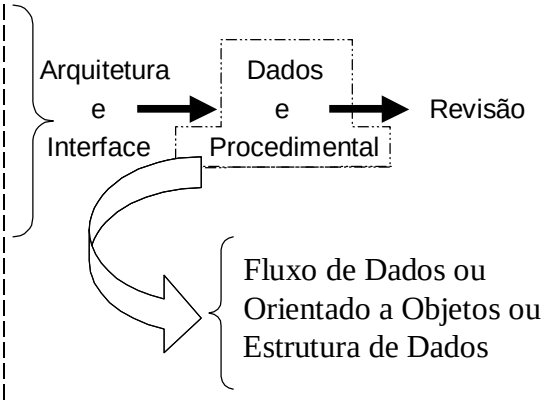
PUC CAMPINAS
Engenharia de Software

Transição da Análise para o Projeto

Análise

- ✓ Especificação de Requisitos
- ✓ Modelos Lógicos
- ✓ Dicionário de Dados (REVISADOS)

Projeto



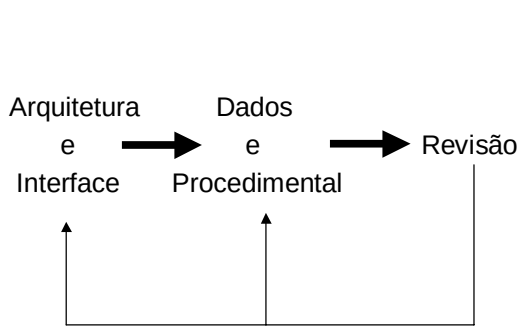
Projeto Orientado a Fluxo de Dados

Projeto Orientado a Fluxo de Dados

Análise

- ✓ Especificação de Requisitos
- ✓ Diagrama de Fluxo de Dados (DFD)
- ✓ Dicionário de Dados

Projeto



PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Fluxo de Dados

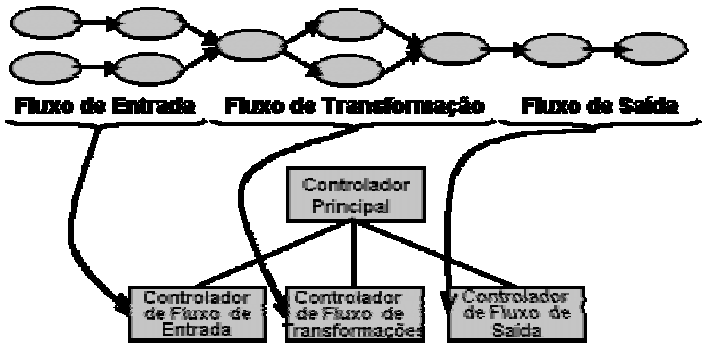
Seqüência:

1. Revisar os modelos de Análise (DFDs, Diag.Dados).
2. Identificar Funções que agrupadas formam subsistemas ou partes do sistema maior.
3. Identificar as Funções que podem ser decompostas, fatoradas (ou explodidas).
3. Efetuar as Fatorações das Funções.
4. Obter da fatoração a primeira versão do Diagrama Hierárquico de Estrutura, contendo os dados que são passados de um módulo para outro.

PUC CAMPINAS
Engenharia de Software

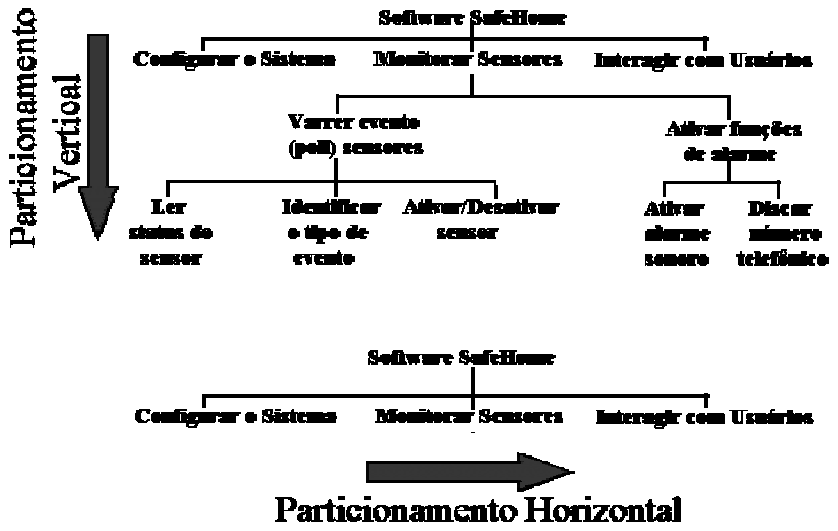
Projeto Orientado a Fluxo de Dados

Fatoração das Funções:



PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Fluxo de Dados



PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Fluxo de Dados

Seqüência (cont):

5. Obter a primeira versão do modelo de Dados do Sistema sejam eles persistentes (MER) ou não (Estruturas de Apoio)
6. Refinar ambos os projetos
7. Especificar e Documentar cada Módulo (PDL, Tabelas de Decisão) e cada estrutura de dados (Listas, tabelas do bancos dados)
8. Projetar Interface para cada módulo de E/S
9. Revisar Projetos de Dados e Procedimental

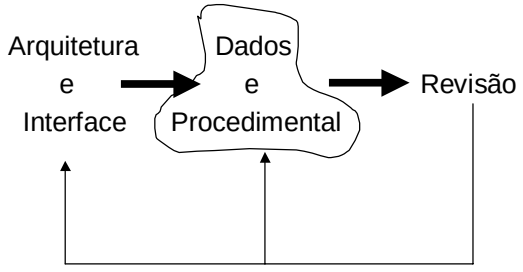
Projeto Orientado a Objetos

Projeto Orientado a Objetos

Análise

- ✓ Especificação de Requisitos
- ✓ Diagrama de Casos de Uso
- ✓ Descrição dos Casos de Uso
- ✓ Dicionário de Dados

Projeto



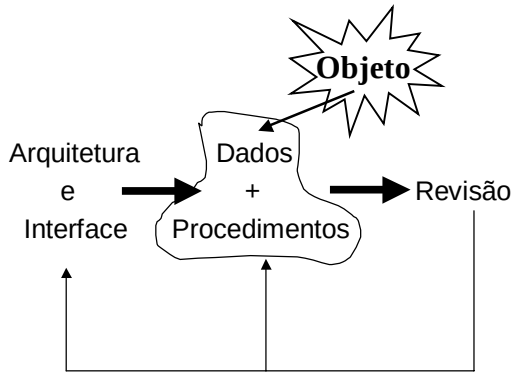
PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Objetos

Análise

- ✓ Especificação de Requisitos
- ✓ Diagrama de Casos de Uso
- ✓ Descrição dos Casos de Uso
- ✓ Dicionário de Dados

Projeto



PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Objetos

Seqüência:

1. Revisar os modelos de Análise: Casos de uso (funcionalidades) e Dicionário Dados
2. Obter Diagrama de Classes Conceitual -Classes do Domínio do Problema, seus atributos, operações, associações, especializações e generalizações (herança)
3. Obter a Percepção de como cada Caso de Uso será implementado no sistema, através dos Diagrama de Interação
4. Obter os Modelos de Comportamento do Sistema - Estados dos Objetos

PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Objetos

Seqüência (cont.):

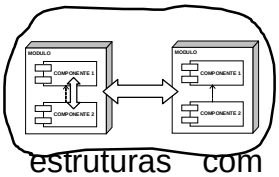
5. Identificar e projetar as classes de Interface (E/S) obtendo Diagrama de Classes Detalhado
6. Refinar os modelos de classes, interação e Estados (Projetar aspectos de reutilização)
7. Especificar e Documentar cada Classe:
   Operações, Atributos, Domínio, Persistência, etc.
8. Projetar Diagramas de Componentes e Implementação
9. Revisar o Projeto Orientado a Objetos

PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Objetos
Heurística de Projeto

Heurística de Projeto:

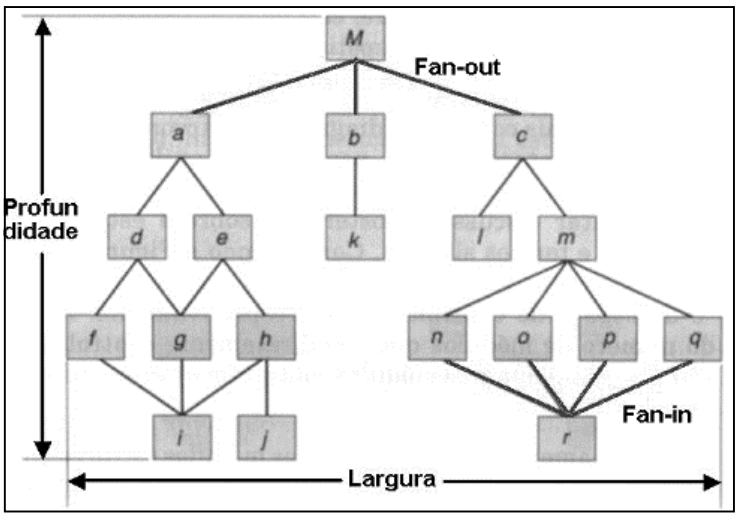
1. Avalie a estrutura do programa para reduzir acoplamentos e melhorar a coesão;



2. Tente minimizar estruturas com elevado FAN-OUT. Esforce-se para ter FAN-IN à medida que a profundidade do projeto aumentar;

PUC CAMPINAS
Engenharia de Software

Projeto de Software
Heurística de Projeto

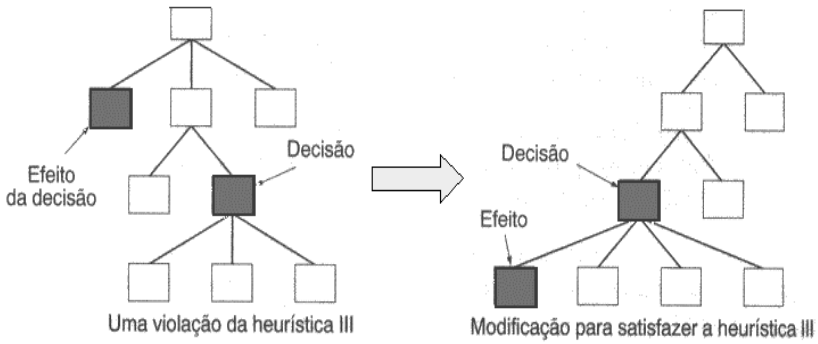


PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Objetos
Heurística de Projeto

Heurística de Projeto:

3. Mantenha o alcance dos efeitos de um módulo dentro do alcance de controle desse módulo;



PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Objetos
Heurística de Projeto

Heurística de Projeto:

4. Avalie interfaces para reduzir a complexidade e melhorar a consistência e entendimento;

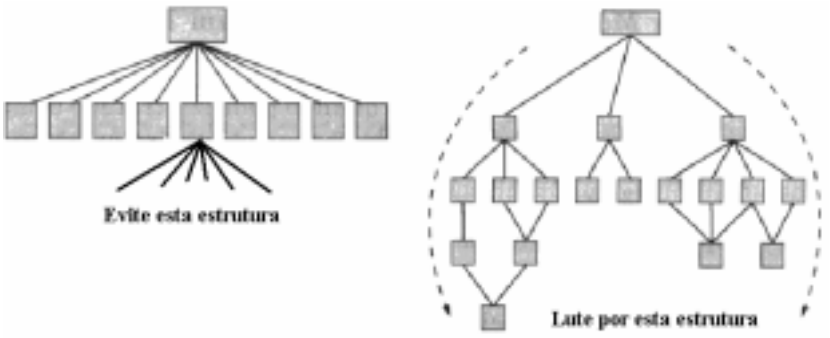
5. Defina módulos cujas funções sejam previsíveis mas não crie módulos exageradamente restritos;

PUC CAMPINAS
Engenharia de Software

Projeto de Software
Heurística de Projeto

Modularidade efetiva:

1. Independência funcional; e
2. Módulos com um só propósito.



PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Objetos
Heurística de Projeto

Heurística de Projeto:

6. Lute por módulos com uma única entrada e uma única saída (sem acoplamentos patológicos);

7. Empacote o software tendo como base os requisitos de portabilidade e as restrições de projeto (por exemplo a plataforma do ambiente de operação).

PUC CAMPINAS
Engenharia de Software

Projeto Orientado a Objetos
Exercício

1. Projete a interface de pesquisa ao acervo do Sistema de Informatização da Biblioteca.
2. Faça o Dicionário de Dados de todas as informações utilizadas na interface projetada para pesquisa ao acervo.
3. Como os conceitos de “acoplamento” e “portabilidade” de software se relacionam? Apresente um exemplo.
4. Ao escrever código fonte, nós estamos projetando software ? Explique.