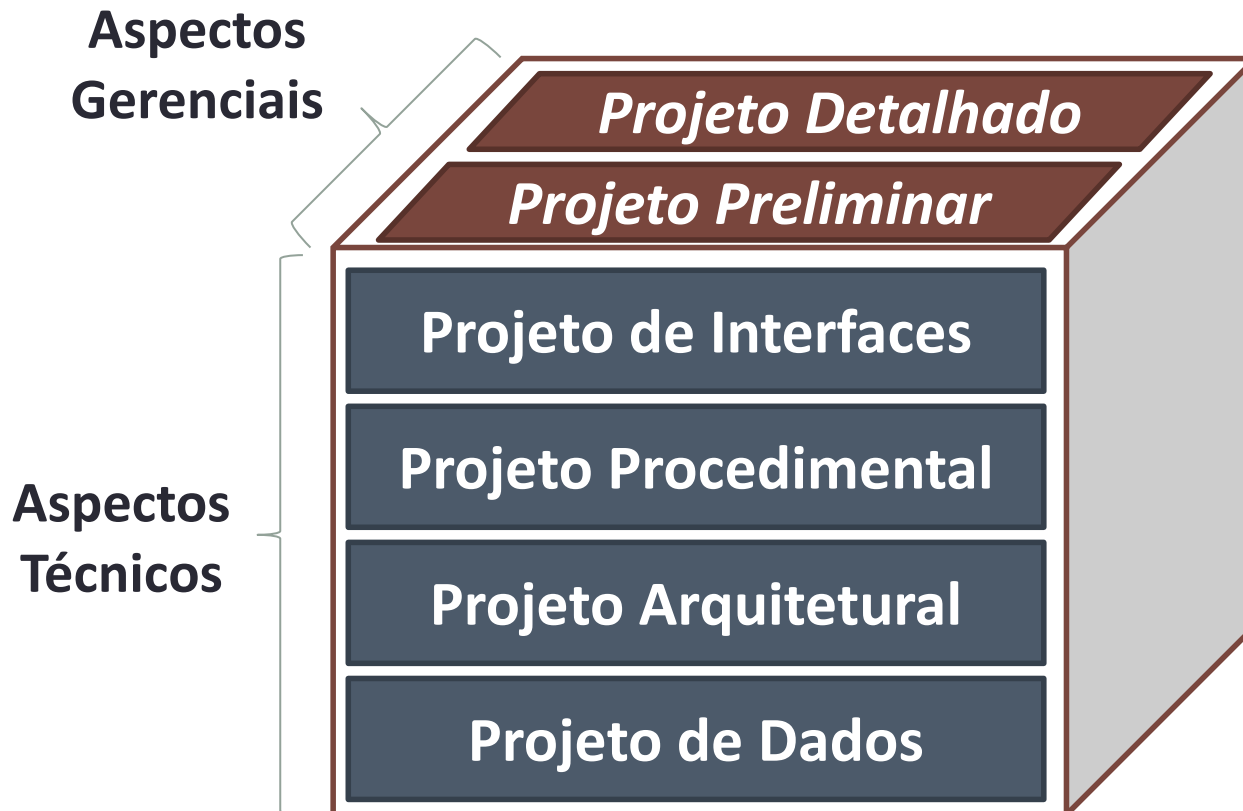


# PROJETO DE PROGRAMAS

---

Projeto de Programas – PPR0001

# Aspectos Técnicos de Projeto



# Projeto de Dados

- Durante a análise se gera o modelo de informação, e com este modelo serão definidas as estruturas de dados necessárias para a implementação do software
- O projeto é influenciado pela tecnologia de persistência escolhida para a implementação
  - Banco de dados, arquivos (xml, csv), etc...
- A base do projeto de dados é um **modelo entidade-relacionamento** (MER), que armazena os objetos de dados e as relações entre eles.

# Projeto Arquitetural

- Define quais tecnologias serão utilizadas:
  - Arquitetura de computador
  - Sistema Operacional
  - SGBD (se for o caso)
  - Linguagem de programação
  - Considerações dos requisitos não-funcionais
- Define o **estilo arquitetural** adotados:
  - Hierarquia / relação dos componentes e das estrutura de dados
  - Padrões arquiteturais: centrado nos dados, orientado a objetos, em camadas, etc...

# Projeto Arquitetural

- Define as interações internas (consigo mesmo):

- Assinaturas de métodos

- Define como interage com outros sistemas:

- E.g.: Troca de Informações entre sistemas bancários

Como outro sistema bancário deve acessar meus dados:

```
string clientData (int conta, int agencia, string codSeg);
```

Como meu sistema deve acessar o sistema bancário X:

```
string getClient (int conta, string codSeg);
```

- Diagrama utilizado:

- Diagrama de classes

# Projeto Procedimental

- Especifica detalhes de processamento de cada módulo
  - Transforma os elementos estruturais da arquitetura em uma descrição procedimental dos componentes
  - Sequência de eventos realizados: Funcionalidades, Algoritmos
- Diagramas utilizados:
  - Diagrama de Fluxo de Dados (DFD)
  - Diagrama de Estados
  - Diagrama de Sequência (UML)

# Projeto de Interfaces

- Define aspectos da interface do sistema:
  - Qual tecnologia será utilizada?
  - Como será a interface gráfica com o usuário?
  - O que deve ser feito para usar a funcionalidade X?
  - Questões envolvendo a 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

# QUALIDADE DO PROJETO

---



# Qualidade do Projeto de Software

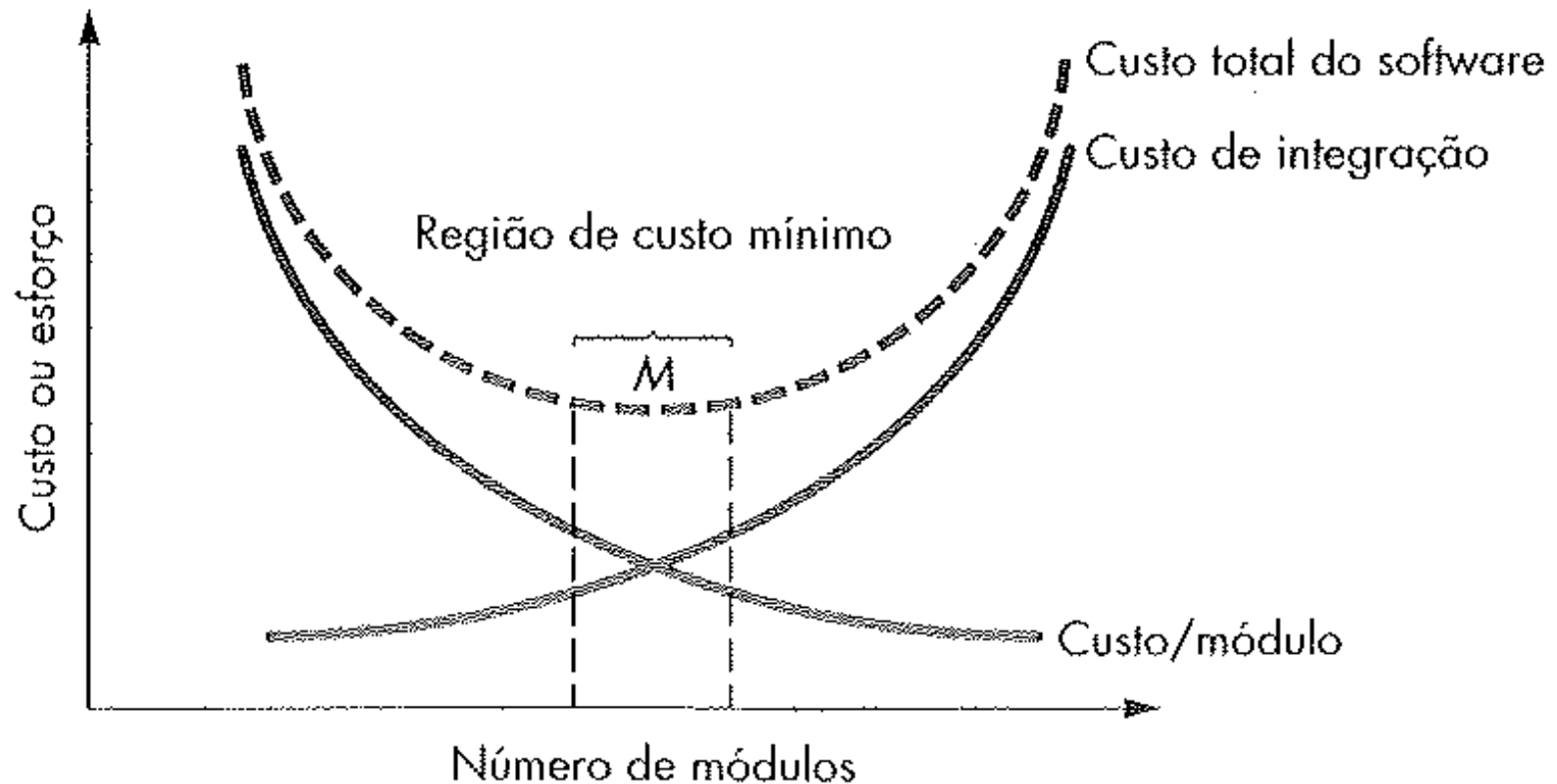
**Modularidade**: gerar particionamento em elementos que executam funções específicas;

- Características:
  - **Coesão**: medida da identidade funcional de um módulo;  
(uma funcionalidade somente? Mais de uma?)
  - **Acoplamento**: Mede o grau em que um módulo está “conectado” a outros elementos;
- Desafios:
  - Qual o número “correto” de módulos para um projeto?
  - Qual o tamanho “correto” dos módulos?



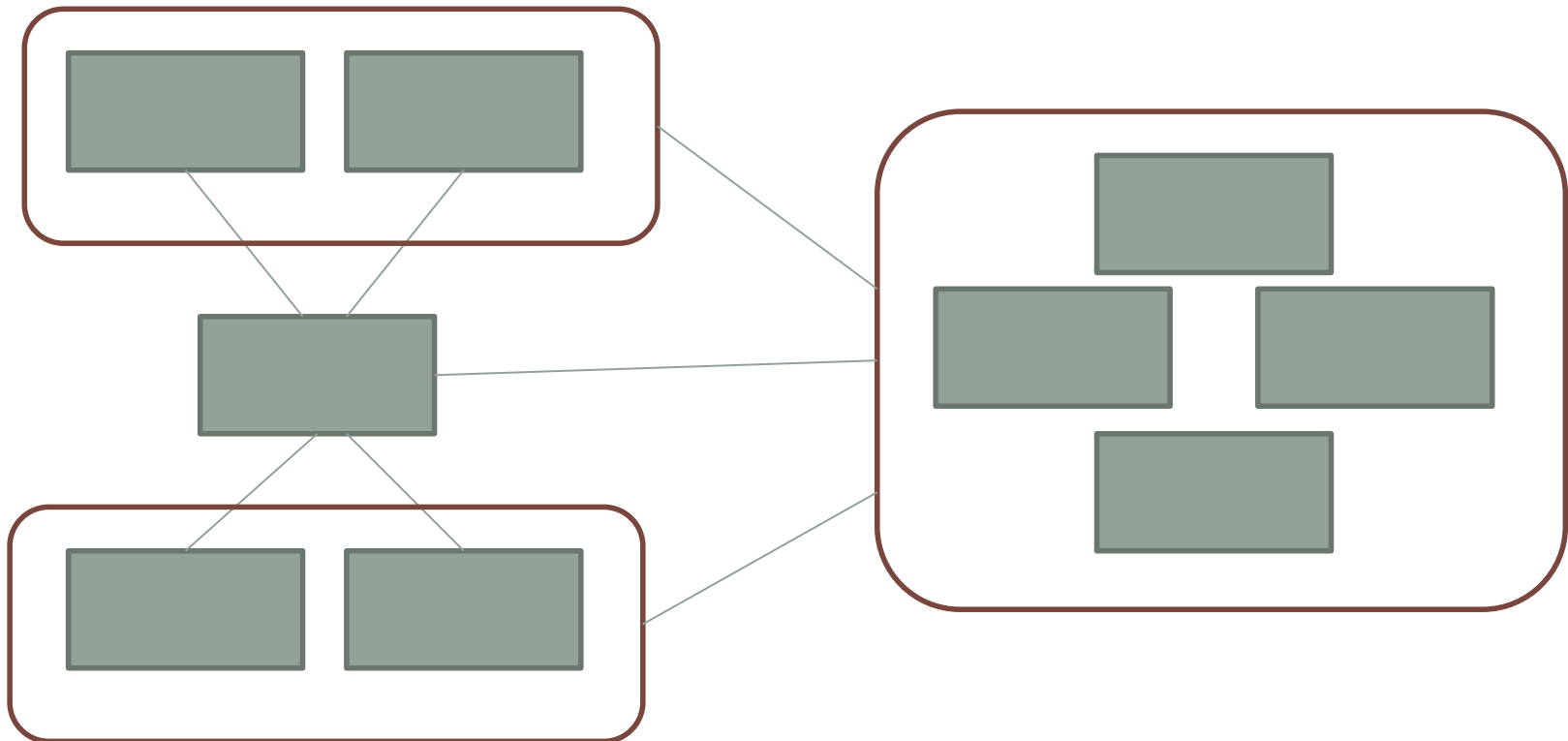
# Qualidade do Projeto de Software

## Modularidade:



# Qualidade do Projeto de Software

**Organização Hierárquica**: manter uma boa organização entre os elementos / módulos de software;



# Qualidade do Projeto de Software

**Encapsulamento**: As informações de um módulo não devem ser acessíveis aos módulos que não precisam delas

- Reduz a ocorrência de efeitos colaterais
  - Erros não se propagam pelo software

( **public, private, package, protected** )

# Fundamentos de Projeto de Software

## Abstração

- Uma abstração é uma representação das características e do comportamento relevantes de um conceito do mundo real para um determinado problema.
- O que faz parte do escopo de projeto?
- Pode ser definida em dois níveis:
  - Dados: informações relacionados a uma entidade
  - Procedimental: Formas de interação/ação com o elemento

# Princípios de Projeto

- O projeto deve se relacionar com o modelo de análise
  - Indicar como os requisitos são satisfeitos no projeto
- Não reinventar a roda - Utilize padrões
- Deve minimizar a distância intelectual entre o software e o problema do mundo real
  - Evitar associações abstratas quando possível
- Deve exibir uniformemente a integração
  - Regras de estilo e projeto deve ser definidas para a equipe do projeto;
  - Um projeto é integrado se houver cuidado na definição de suas interfaces

# Princípios de Projeto

- Projeto deve ser estruturado para suportar modificações
- Projeto **não** é codificação;
- A qualidade do projeto deve ser avaliada enquanto ele é criado
- O projeto deve ser revisto para evitar erros conceituais
  - Garantir que não há omissões, ambiguidades e inconsistências antes de prosseguir para próximas etapas

# PROJETO DE DADOS

---



# Atividades Envolvidas

- Realizar a abstração de dados (visando o escopo do problema);
- Estudar e escolher as estruturas de dados que permitam a implementação mais adequada;
- Caracterizar a Abrangência dos dados:
  - De componente (local) ou parte do software (global);
  - Persistência de dados: uso de Banco de Dados , arquivos ou memória;

# Diagrama de Classes Simplificado

- Por enquanto vamos trabalhar com um modelo simplificado;
- Escreva as classes para o seu sistema e identifique os seus atributos (não pense por enquanto nos tipos de dados)

Nome da Classe
Atributos / Variáveis
Métodos / Funções

# Bibliografia

- **Básica:**

BEZERRA, E. Princípios de Análise e Projetos de Sistemas com UML. Rio de Janeiro: Campus, 2003.

PRESSMAN, R.S. Engenharia de Software. São Paulo: Makron Books, 2002.

SOMMERVILLE, I. Engenharia de Software. São Paulo: Addison Wesley, 2003.

- **Complementar:**

WARNIER, J. Lógica de Construção de Programas. Rio de Janeiro: Campus, 1984.

JACKSON, M. Princípios de Projeto de Programas. Rio de Janeiro: Campus, 1988.

PAGE-JONES, M. Projeto Estruturado de Sistemas. São Paulo: McGraw-Hill, 1988.