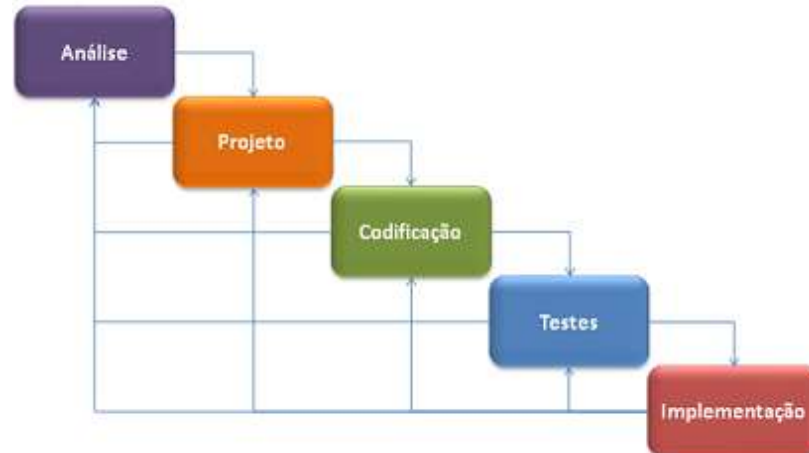


Projeto de Software

Unidade I

Princípios e Processos



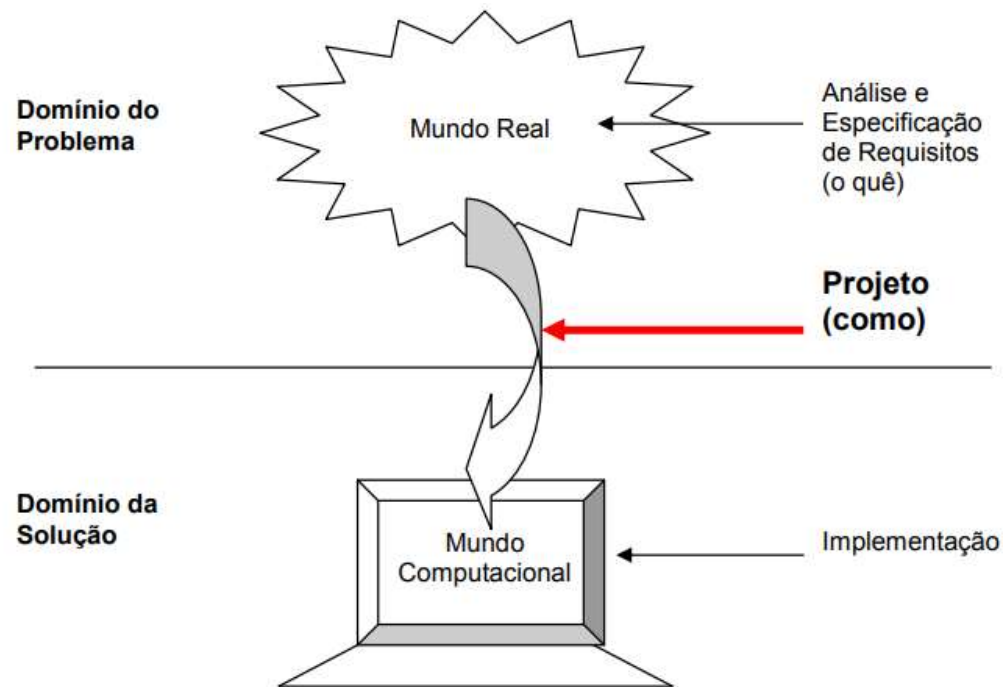
Projeto de Software

- Design
- Desenho
- Criatividade
- Arquitetura
- Modelos Gráficos



Projeto de Software

- O Lugar do Projeto de Software no Ciclo de Desenvolvimento do Software



Projeto de Software

- Roteiro
 - 1. Importância
 - 2. Competências
 - 3. Objetivos Gerais
 - 4. Objetivo Principal
 - 5. Conceitos
 - 6. Princípios Básicos de Projeto
 - 7. Princípios do Projeto Orientado a Objetos
 - 8. Atributos de Qualidade
 - 9. Produtos do Projeto
 - 10. O Processo de Desenvolvimento
 - 11. O Processo de Desenvolvimento Orientado a Objetos



Projeto de Software

- I. Importância do Projeto
 - Traduzir precisamente os requisitos;
 - Base para todas as etapas posteriores do ciclo de vida do desenvolvimento;
 - A chave da qualidade do software!



Projeto de Software

- 2. Competências do Projeto

- O projeto deve implementar todos os requisitos contidos no modelo de análise;
- O projeto deve ser um guia inteligível para os codificadores e testadores;
- O projeto deve fornecer um quadro completo do software, focalizando aspectos informacionais, funcionais e comportamentais.



Projeto de Software

- 3. Objetivos Gerais do Projeto

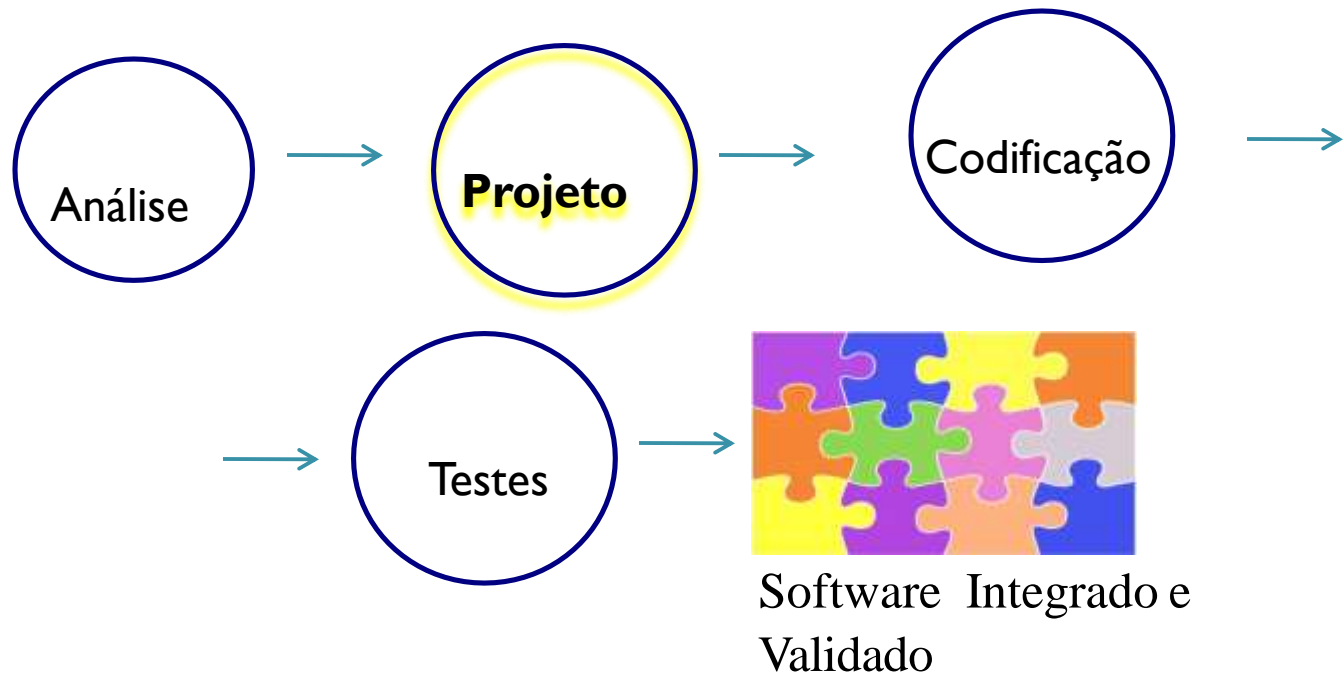
- Fomentar a qualidade durante o processo de desenvolvimento;
- Fornecer representações do software que podem ser avaliadas quanto à qualidade;
- Traduzir com precisão os requisitos de um cliente num produto de software finalizado.



Projeto de Software

- 4. Objetivo Principal do Projeto

- Transformar os requisitos de software para uma representação (modelos) de software que permita sua implementação.



Projeto de Software

- 5. Conceitos de Projeto



Projeto de Software

- 5. Conceitos de Projeto
 - **Abstração:** transformação do mundo real em modelos;
 - **Refinamento:** decompor um enunciado macroscópico de função em enunciados menores;
 - **Modularidade:** conceito de divisão do software em componentes, cada um com um objetivo, que quando integrados satisfazem os requisitos do problema;



abstrato

Que não é concreto nem real; irreal.
Que resulta do processo intelectual de abstração,
só podendo existir no pensamento (ideia).

Projeto de Software

- 5. Conceitos de Projeto

- **Ocultamento:** técnica pela qual um componente torna inacessível a outro, suas informações e comportamento;
- **Coesão:** técnica que torna um componente independente ou pouco dependente de outros, através de sua especialização;
- **Acoplamento:** medida da interconexão entre componentes numa estrutura de software;



Projeto de Software

- **6. Princípios Básicos do Projeto**



Projeto de Software

- 6. Princípios Básicos do Projeto
 - Não reinventar a roda (use padrões!);
 - Apresentar uniformidade e integração;
 - Pode ser avaliado quanto à qualidade, à medida que é desenvolvido;
 - Deve ser estruturado para permitir modificações;
 - Projeto (ainda) não é codificação!



Projeto de Software

- 6. Princípios Básicos do Projeto
 - Coesão alta e Acoplamento baixo;
 - Módulos devem ter um conjunto pequeno e bem- definido de responsabilidades;
 - Módulos devem depender um dos outros o mínimo possível.
 - facilidade de manutenção e compreensão.



Projeto de Software

- 6. Princípios Básicos do Projeto

- Dependências entre módulos não devem formar ciclos;
- Módulos de alto nível demandam os módulos de baixo nível.



Projeto de Software

- 7. Princípios do Projeto Orientado a Objetos



Projeto de Software

- 7. Princípios do Projeto Orientado a Objetos
 - Separe o que é constante do que varia;
 - Encapsule o que varia!
 - Desenvolva para uma interface e não para uma implementação;



Projeto de Software

- 7. Princípios do Projeto **Orientado a Objetos**
 - Prefira composição de objetos a herança;
 - Herança quebra o encapsulamento;
 - Maior flexibilidade;
 - Estrutura do sistema pode mudar em tempo de execução;
 - Maior coesão;
 - Classes mais focadas em um único objetivo.



Projeto de Software

- 7. Princípios do Projeto Orientado a Objetos
 - Uma Classe deve ser aberta para extensão e fechada para modificação;
 - Subclasses devem ser capazes de substituir suas superclasses.



Projeto de Software

- 8. Atributos de Qualidade do Projeto



Projeto de Software

- 8. Atributos de Qualidade do Projeto
 - Facilidade de entendimento;
 - Facilidade de implementação;
 - Facilidade de realização de testes;
 - Facilidade de modificações e de tradução corretas das especificações de requisitos.



Projeto de Software

- 8. Atributos de Qualidade do Projeto
 - Facilidade de fazer modificações;
 - Independência dos componentes (módulos);
 - Identificação e tratamento de exceções(erros);
 - Prevenção de defeitos e tolerância a defeitos;



Projeto de Software

- 9. Produtos do Projeto

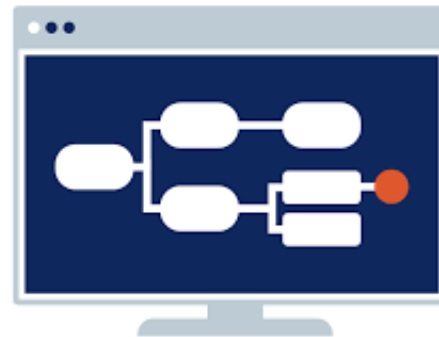


Projeto de Software

- 9. Produtos de um projeto

- **Projeto de Interface**

- Layouts e mecanismos de interação homem-máquina;
- Mockups.

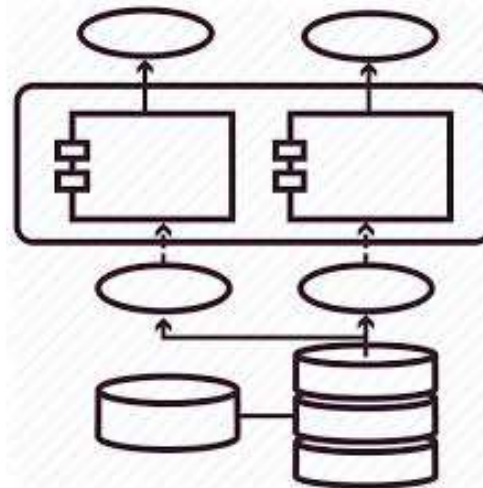


Projeto de Software

- 9. Produtos de um projeto

- **Projeto Arquitetural**

- Relacionamento/associação entre os grandes componentes estruturais do programa (módulos);
- Interconexões/comunicação entre módulos;

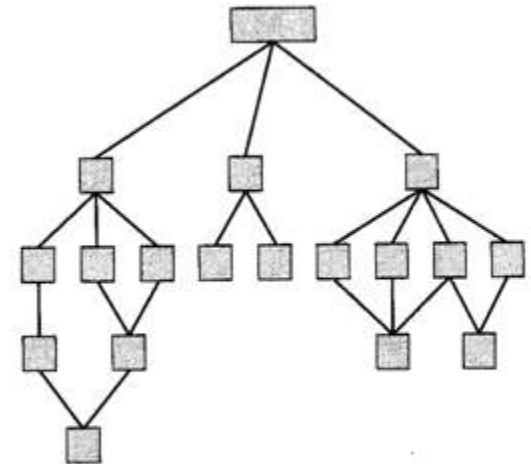
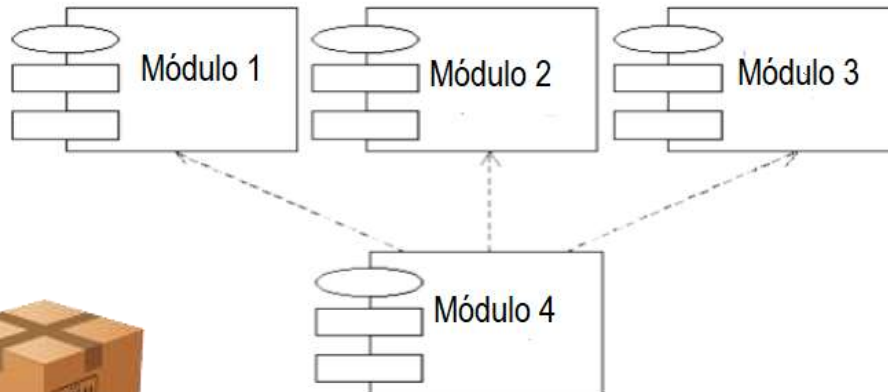


Projeto de Software

• 9. Produtos de um projeto

• **Projeto Arquitetural**

- Associação entre os principais elementos/módulos estruturais do software;
 - Árvore dos módulos;
 - Mensagens entre objetos;
 - Nivelamento em camadas.



Projeto de Software

- 9. Produtos de um Projeto Software
 - **Projeto Procedimental (comportamental)**
 - Descrição da funcionalidade do software (algoritmos);
 - Transforma os componentes estruturais em descrições procedimentais;
 - Apresenta os detalhes de processamento de cada módulo individualmente.

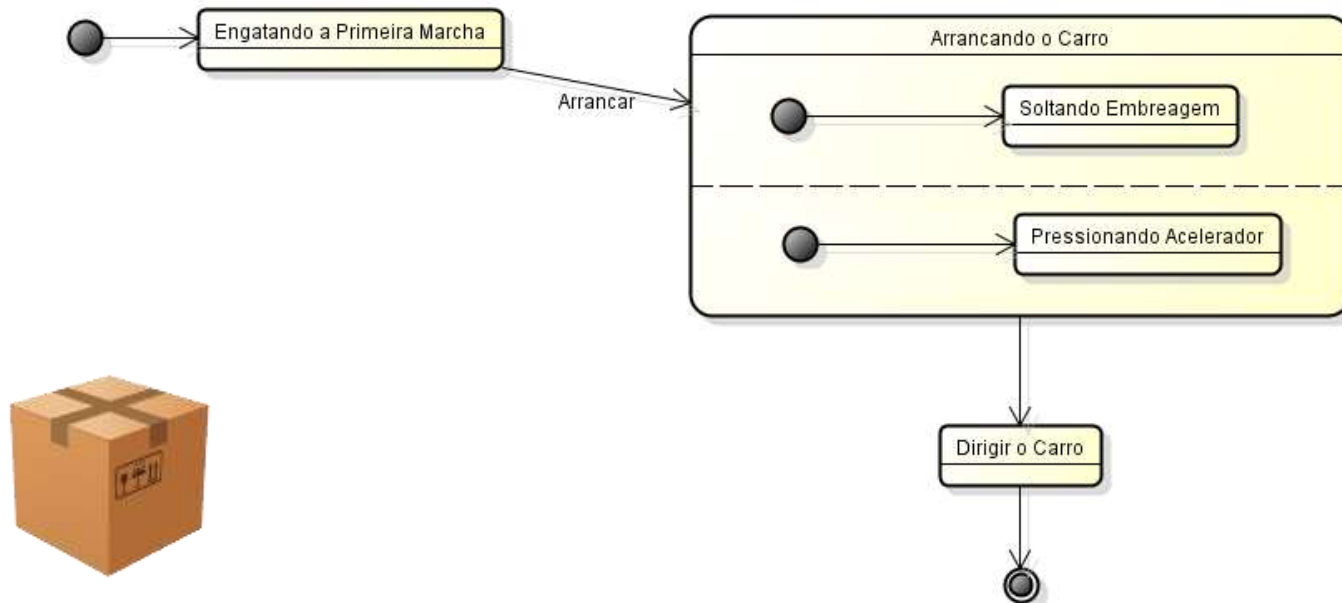


Projeto de Software

- 9. Produtos de um Projeto Software

- **Projeto Procedimental (comportamental)**

- O procedimento deve fornecer uma especificação precisa do processamento, incluindo sequência de eventos, pontos exatos de decisão, operações repetitivas e até organização e estrutura de dados;

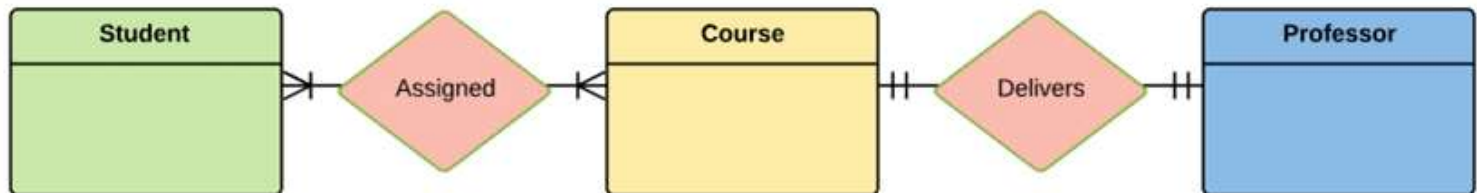


Projeto de Software

- 9. Produtos de um Projeto Software

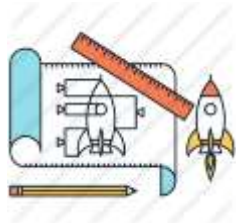
- **Projeto de Dados**

- Modelo do domínio da informação transformado em estruturas de dados;
- Definição das estruturas de dados.



Projeto de Software

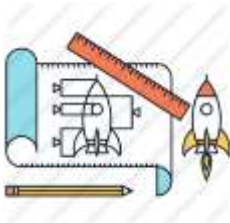
- 10. O Processo de Desenvolvimento do Projeto



O Abstrato se
transforma no
Concreto na execução
do planejamento.

Projeto de Software

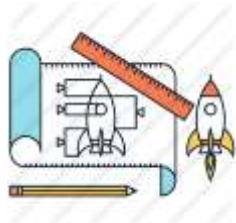
- 10. O Processo de Desenvolvimento
 - Atividade do ciclo de vida na qual os requisitos de software são analisados para produzir uma descrição da estrutura interna do software que servirá de base para a sua construção.



Projeto de Software

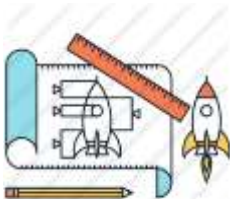
- 10. O Processo de Desenvolvimento

- É a transformação de requisitos de software
 - estabelecidos em termos relevantes ao domínio do problema;
- Em uma descrição
 - que explica como solucionar os aspectos do problema relacionados com software.



Projeto de Software

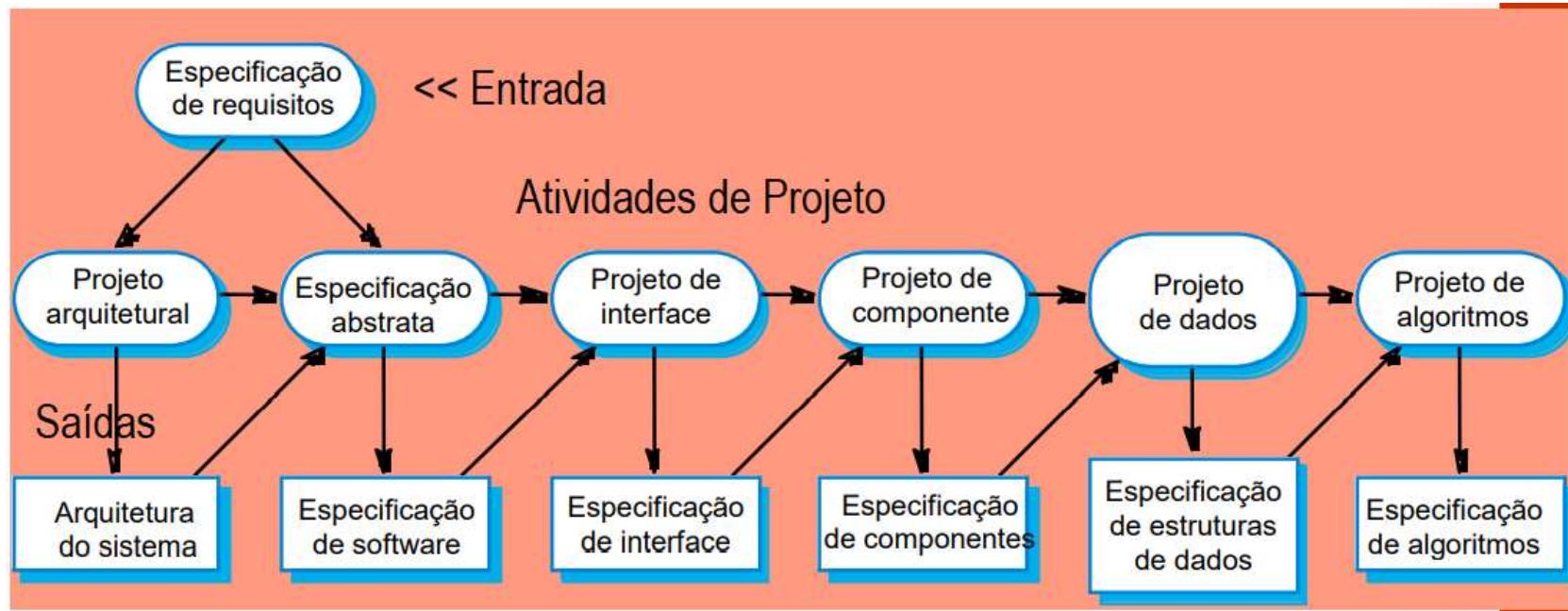
- 10. O Processo de Desenvolvimento
 - **O Resultado do processo**
 - Arquitetura do software;
 - Sistema decomposto e organizado em módulos e as interfaces entre esses módulos;
 - Refinamento dos módulos até um nível de detalhamento que permita a sua implementação.



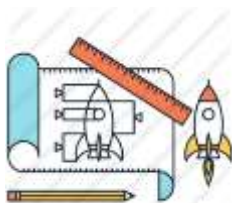
Projeto de Software

- 10. O Processo de desenvolvimento

- **Modelo Genérico de Processo**

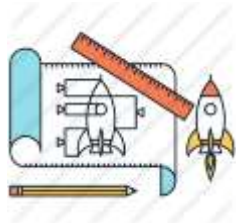


• Fonte: Sommerville



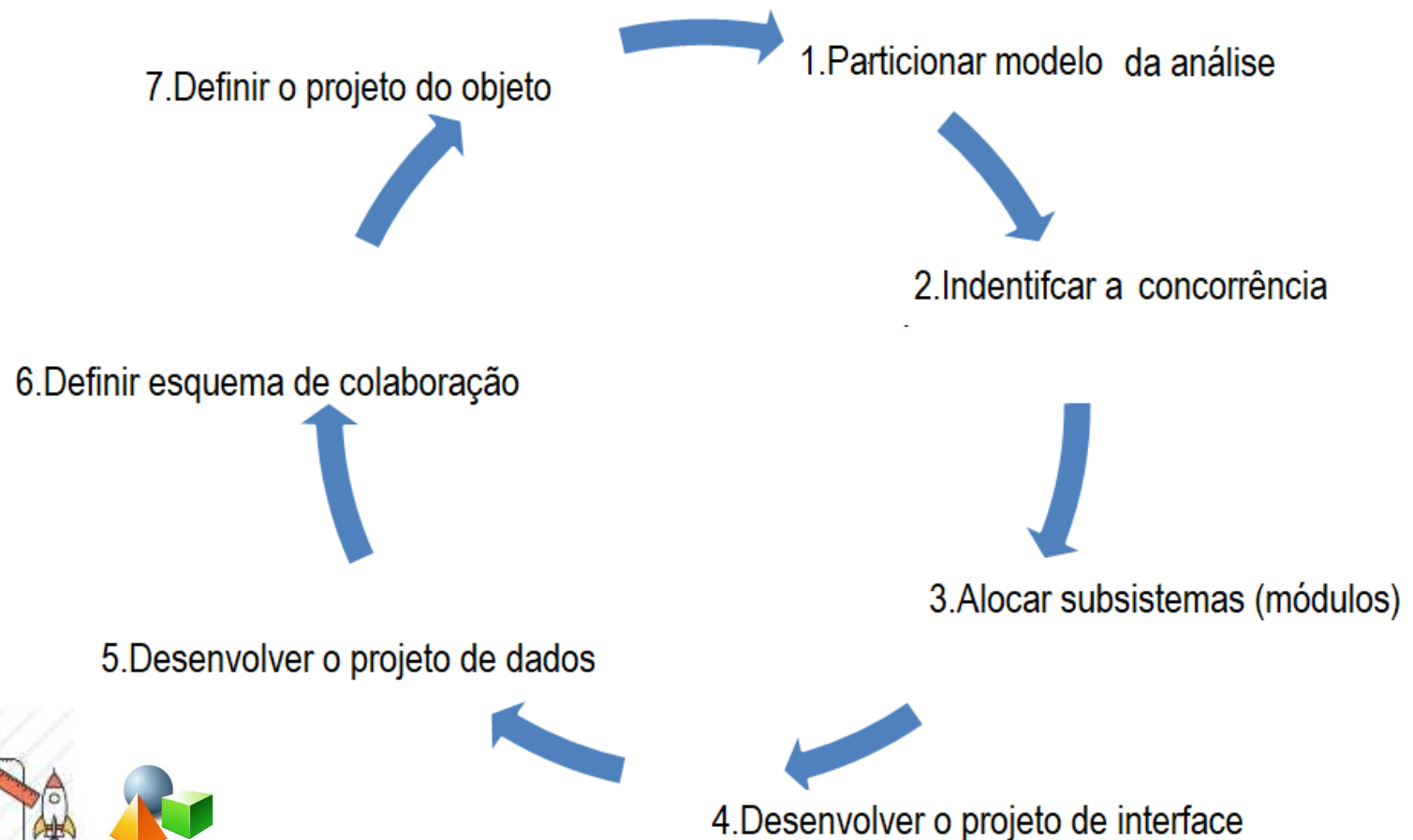
Projeto de Software

- II. O Processo de Desenvolvimento Orientado a Objetos



Projeto de Software

- II. O Processo de Desenvolvimento Orientado a Objetos

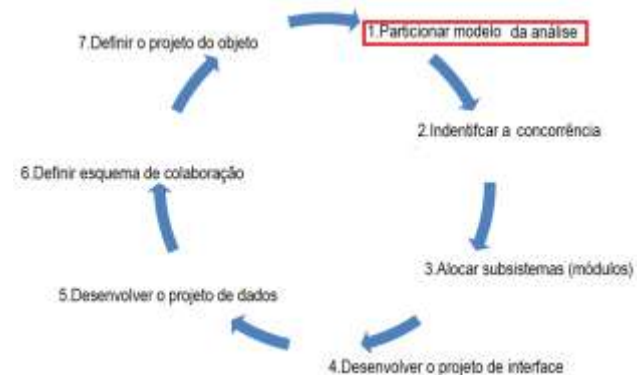
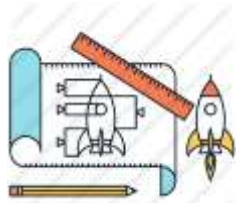


Projeto de Software

- II. O Processo de Desenvolvimento Orientado a Objetos

- **Particionar o Modelo de Análise**

- Definir coleções de classes, relacionamentos e comportamentos coesivos, a serem empacotados como um Módulo (subsistema).

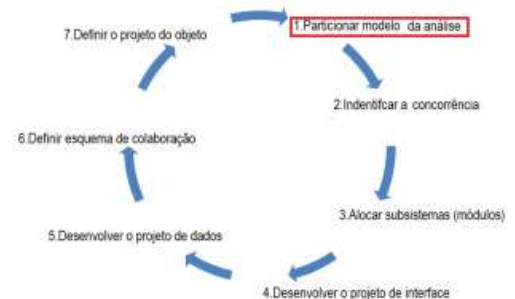
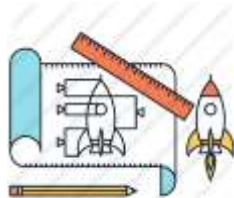


Projeto de Software

- II. O Processo de desenvolvimento
Orientado a Objetos

- **Módulo (subsistema)**

- Se necessário, um módulo deve ser particionado para reduzir a complexidade;
- Deve possuir uma única responsabilidade;
- Deve ter uma interface simples e bem definida;
- Módulos deve ter única entrada e única saída;
- Classes de um módulo devem colaborar apenas entre si.



Projeto de Software

- II. O Processo de Desenvolvimento Orientado a Objetos

- **Identificar a Concorrência**

- Classes que precisam agir sobre eventos de modo assíncrono e simultaneamente, são vistas como concorrentes.

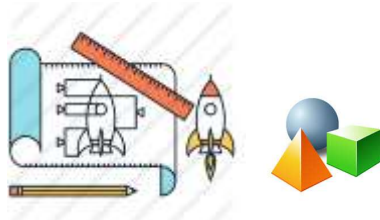


Projeto de Software

- II. O Processo de Desenvolvimento Orientado a Objetos

- **Tratamento da Concorrência**

- Aloca-se cada módulo a um processador independente ou
- Aloca-se os módulos ao mesmo processador e adota-se suporte de concorrência.

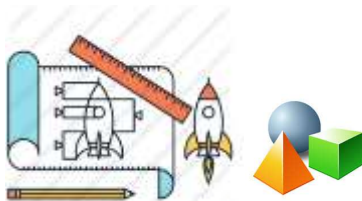


Projeto de Software

- II. O Processo de Desenvolvimento Orientado s Objetos

- **Desenvolver o Projeto de Interface**

- Usa como entrada o diálogo entre sistema e ator descrito nos casos de uso;
- Fácil de aprender: tempo e esforço mínimo para bom desempenho no uso do sistema;
- Fácil de usar: rapidez de execução de tarefas e redução de erros no uso do sistema.

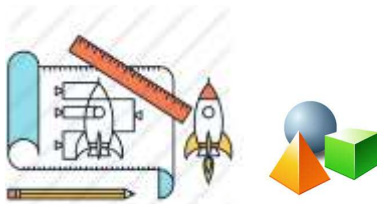


Projeto de Software

- II. O Processo de Desenvolvimento Orientado a Objetos

- **Desenvolver o Projeto de Dados**

- Definir estratégias para gestão de dados;
- Desenvolvimento do projeto dos atributos e das operações necessárias para gerir e persistir os objetos;



Projeto de Software

- **II. O Processo de Desenvolvimento Orientado a Objetos**

- **Desenvolver o Projeto de Dados**

- Construir o modelo de classes;
- Construir o modelo conceitual de dados;
- Construir o modelo lógico de dados;
- Construir o modelo físico de dados;
- Utilizar um padrão* para gerir as operações primitivas de acesso ao banco de dados.

*DAO (data access object) ou AR (active record)

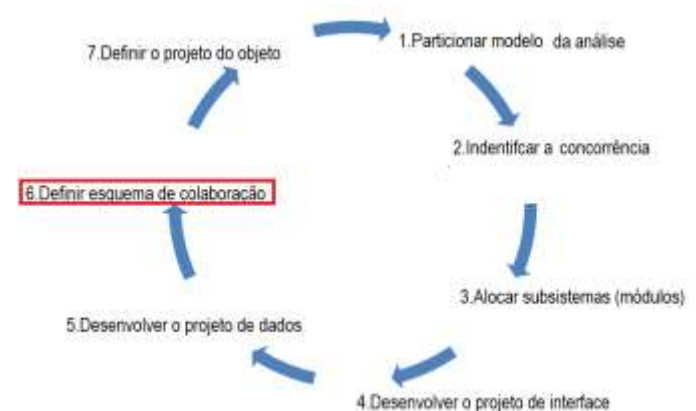
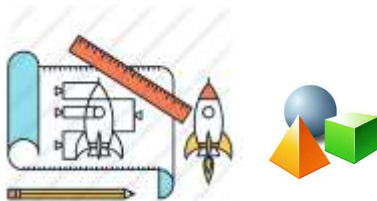


Projeto de Software

- II. O Processo de Desenvolvimento Orientado a Objetos

- Definir Esquema de Colaboração

- Colaboração é um meio de comunicação entre módulos (subsistemas), endereçado por um contrato.

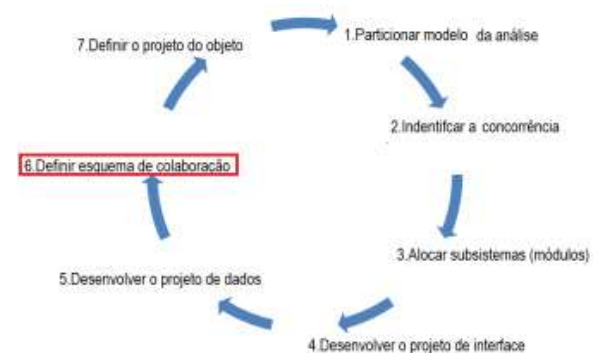
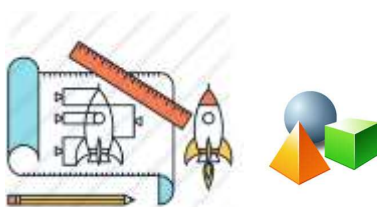


Projeto de Software

- **II. O Processo de Desenvolvimento Orientado a Objetos**

- **Definir Esquema de Colaboração**

- O contrato indica o modo que o módulo deve interagir;
- O contrato é manifestado por mensagens que se movem entre os objetos dentro dos módulos;
- O contrato deve conter as classes, operações e formato das mensagens que implementam as interações entre os módulos.

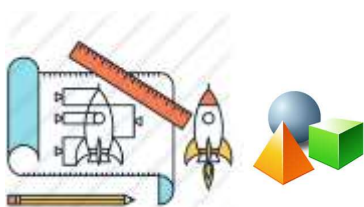


Projeto de Software

- **II. O Processo de Desenvolvimento Orientado a Objetos**

- **Definir o Projeto de Objetos**

- Especificar os objetos, suas informações, seu comportamento e sua interface.
 - Especificação de Classe;
 - Especificação de Atributos;
 - Especificação de Operações/Métodos.



Projeto de Software

• Bibliografia

- Braude, Eric. Projeto de Software - da programação à arquitetura: uma abordagem baseada em Java. Porto Alegre: Bookman, 2005.
- Sommerville, Ian. Engenharia de Software. Editora Pearson, 2011
- Masiero, Paulo C.
https://edisciplinas.usp.br/pluginfile.php/24117/mod_resource/content/1/Aula_5-ProjetoSoftware.pdf
- Zaparoli, Wagner.
<https://pt.slideshare.net/WagnerZaparoli/projeto-de-software-51586755>

Projeto de Software

Obrigado pela Atenção!



Fique atento e focado! Vem ai a lista de atividades dessa unidade!

Projeto de Software

- Tópicos para seminários

tarefa de casa



Projeto de Software

• Sugestões de Tópicos para Seminários

- UML: Linguagem de Modelagem;
- Metodologias
 - Ágeis
- Planos de Projetos;
- Gestão de Projetos;
 - Ferramentas
- Estilos Arquitetônicos;
 - Arquitetura de Software
- Padrões de Projetos (Design Patterns);
 - Soluções Reutilizáveis de Software

