



CURSO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
**CMP1550**

# **Desenho de aplicativos para dispositivos móveis**

Prof Fabricio Schlag  
Resumo das aulas anteriores - ES

# Como é o processo de desenvolvimento de software

- Fases:
  - Análise de Requisitos
  - Projeto
  - Codificação
  - Teste
  - Manutenção

# Análise de Requisitos

- Levantamento dos requisitos do sistema que será construído
  - Captura
  - Modelagem
  - Avaliação
  - Documentação
- Modelagem conceitual é uma atividade essencial
  - => Elabora modelos descrevendo o que o software deve fazer

# Domínio da Solução

- Com um levantamento parcial dos requisitos já é possível já visualizar a solução
- Muitas soluções são possíveis para um mesmo conjunto de requisitos

# Projeto ou Design

- Tem como função especificar uma solução a ser implementada
- Fase de tomada de decisões
  - Várias soluções possíveis
- É também um processo de refinamento

# Projeto da Arquitetura



# Projeto da Arquitetura

- Objetivo de descrever a estrutura de mais alto nível da aplicação
- Identifica os principais elementos ou componentes
  - E a relação entre eles

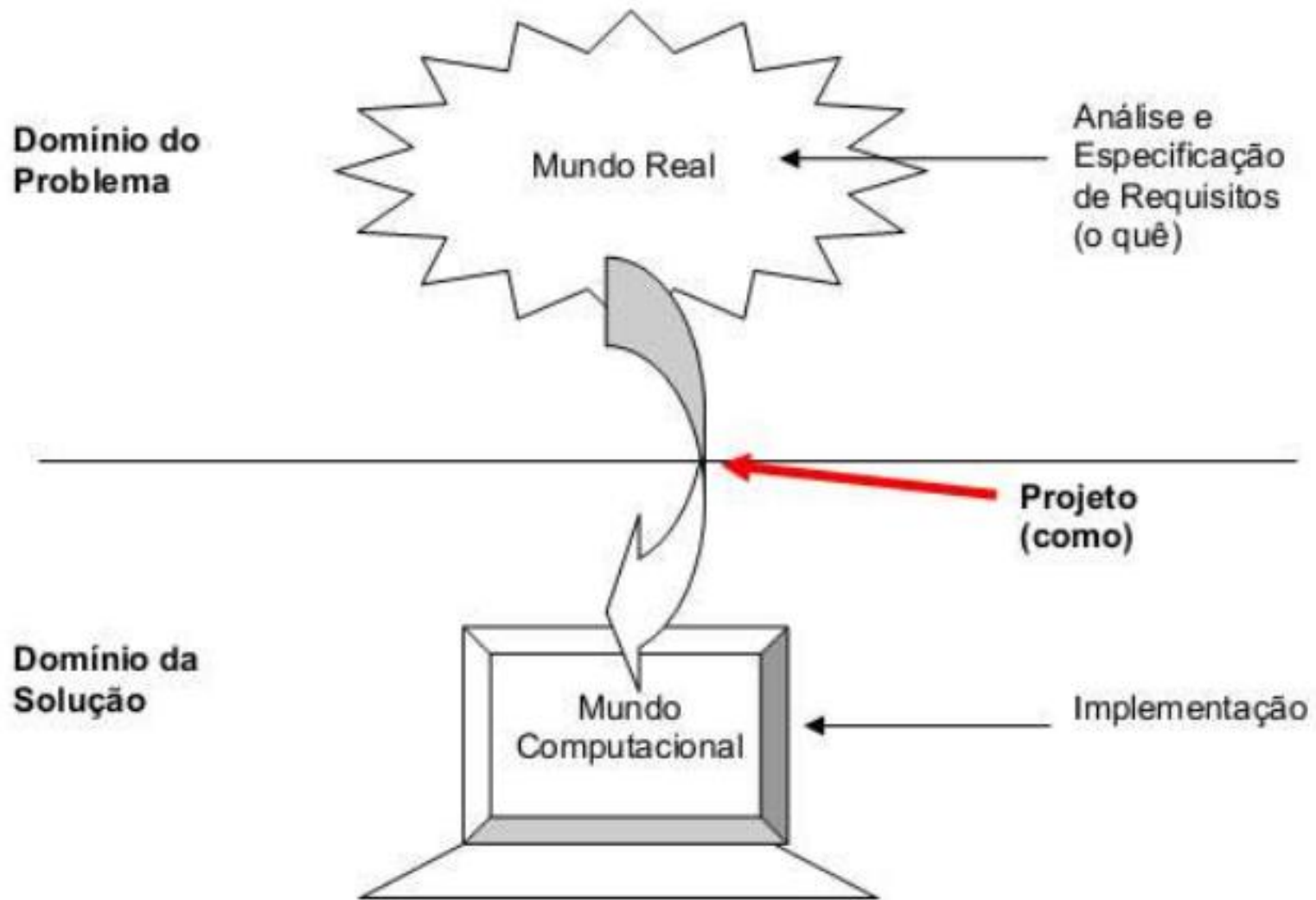
# Esforço do Projeto

- Concentra-se no detalhamento de cada um dos elementos da arquitetura
- Este elementos deve chegar ao nível de implementação

# O projetista deve

- Conhecer as tecnologias existentes
- Conhecer os ambientes de Hardware e Software onde o sistema vai ser implantado

# A fase do Projeto



# Fase de Projeto

- Projeto é um processo de refinamento
  - Inicialmente é representado num alto nível de abstração
  - Após a definição da arquitetura refinamentos levam a representações de menores níveis de abstração
  - A intenção é chegar ao projeto de algoritmos e estrutura de dados

# Atividades do Projeto

- Projeto de arquitetura
  - visa definir os elementos estruturais do software e seus relacionamentos
- Projeto dos elementos da arquitetura
  - decomposição de módulos em outros módulos menores
- Projeto Detalhado
  - Refinar e detalhar os elementos mais básicos da arquitetura:
  - Interfaces
    - internas – entre elementos
    - externas – outros sistemas
  - Procedimentos (algoritmos)
  - Estruturas de Dados

# Outros elementos de Arquitetura

- Lógica de Negócio
  - elemento da arquitetura que trata da lógica de negócio apoiada pelo sistema
- Interface com o Usuário
  - Interação Homem-computador
  - Elementos de Interface - Ex: botões, telas, campos de texto, etc.
  - Controle de Interação - habilitando e desabilitando botões , abrindo e fechando janelas, etc.
    - Ex: Se sexo for masculino habilitar campo Número de Reservista do Exército
- Persistência
  - Armazenamento e recuperação dos dados da aplicação

# Projeto de Software

- Transforma especificação de um problema (requisitos) em uma especificação de uma solução
- Muitas soluções são possíveis
- Várias decisões devem ser tomadas
  - Para acomodar os requisitos funcionais e não funcionais

# Como é a construção de uma casa?

- Análise das necessidades do dono da casa
  - Numero de cômodos (quartos, salas, banheiros, etc.)
  - Tipos de serviços a serem providos (central de ar condicionado, sistema de aquecedor solar)
- Restrições a serem levantadas
  - Custo
  - Prazos
  - Área disponível para construção
  - Acessibilidade
  - Restrições legais

# A casa deve (projeto)

- Prover uma solução para os problemas colocados pelo cliente
- Muitas soluções são possíveis
- Arquitetos diferentes darão soluções diferentes
- O cliente escolherá a melhor solução
  - Solução que atende os requisitos especificados

# O projeto da casa deve (projeto)

- considerar abordagens alternativas com base nos requisitos do problema, restrições e conceitos de projeto;
- ser rastreável à sua especificação;
- não “reinventar a roda”, isto é, reutilizar soluções
- exibir uniformidade (estilo) e integração (interfaces bem definidas entre componentes da coisa a ser construída);

# Projeto da casa deve...

- ser estruturado para acomodar mudanças;
- ser passível de avaliação da qualidade;
- ser revisado para minimizar erros.

# Projeto da casa deve...

- prover uma visão da totalidade da coisa a ser construída;
- decompor o todo em partes e prover diferentes visões da coisa;
  - Plantas baixas e maquetes, sistema hidráulico, sistema elétrico, etc.
- refinar e descrever com mais detalhes cada parte ou visão da coisa, de modo a prover orientação para a construção de cada detalhe;

# Projeto de Software X Projeto de uma Casa

- As características citadas para um projeto de uma casa valem para projeto de software

# Projeto de Software deve

- Considerar abordagens alternativas com base nos requisitos (funcionais e não funcionais) e conceitos de projeto de software;
- Estar relacionado aos modelos de análise e à especificação de requisitos e deve ser a eles rastreado;
- Não “reinventar a roda”
  - reutilizar componentes
  - Frameworks
  - padrões
  - outras soluções que se mostraram eficazes em outros projetos, sobretudo aqueles similares ao sistema em desenvolvimento;

# Projeto de Software deve

- exibir uniformidade (estilo) e integração (interfaces entre componentes);
- ser estruturado para acomodar mudanças (alterabilidade);
- ser passível de avaliação da qualidade;
- ser revisado para minimizar erros.

# Projeto de Software deve

- Minimizar a distância conceitual e semântica entre o software e o mundo real
  - Modelos de projeto devem ser facilmente compreensíveis
- Apresentar nível de abstração superior ao código fonte, afinal, projeto não é codificação

# Princípios Gerais da Engenharia de Software

- Um sistema de software existe para fornecer valor aos clientes e usuários.
  - Todas as decisões, inclusive as de projeto, devem ser tomadas tendo isso em mente.
- Todo projeto de software deve ser tão simples quanto possível

# Princípios Gerais da Engenharia de Software

- O comprometimento com a visão arquitetural do sistema é essencial para o sucesso do projeto de software.
- Os modelos elaborados na fase de projeto serão usados posteriormente por:
  - desenvolvedores
  - responsáveis pela implementação
  - testes e manutenção do sistema

# Princípios Gerais da Engenharia de Software

- Um sistema com um longo tempo de vida tem mais valor
  - sistema deve ser projetado para estar pronto para acomodar mudanças
- A reutilização pode ajudar a poupar tempo e esforço
  - aumenta a qualidade do sistema em desenvolvimento

# Princípios Gerais da Engenharia de Software

- Raciocinar clara e completamente antes de realizar uma ação quase sempre produz melhores resultados
- Aprender com os erros também é importante
- Raciocinar sobre uma decisão de projeto
  - soluções anteriores devem ser pesquisadas.

01

Dúvidas?

02

....

N