



UNIVERSIDADE  
FEDERAL DO CEARÁ  
Campus Russas

# ARQUITETURA DE SOFTWARE

**Motivação e Benefícios**

Profa. Dra. Anna Beatriz Marques

Baseado em slides do Prof. Hyggo Oliveira de Almeida

# O QUE É ARQUITETURA?

- Arquitetura como um *processo*
  - A arte ou ciência de **construir edificações** de quaisquer tipos para uso humano
  - A ação ou **processo de construir**
  - O **método ou estilo** especial que direciona os detalhes da **estrutura e ornamentação** de uma construção
- Arquitetura como um *artefato*
  - Trabalho arquitetural: **estrutura, construção**
  - **Construção ou estrutura** em geral

# O QUE É ARQUITETURA DE SOFTWARE?

## Muitas definições...

- “...conjunto de elementos arquiteturais (ou de projeto) que têm uma forma particular.” **Perry and Wolf, 1992**
- “...estrutura(s) do sistema, que compreende(em) os componentes de software, as propriedades externamente visíveis desses componentes, e os relacionamentos entre eles.” **Clements et al., 1997**
- “...especificação abstrata do sistema consistindo primeiramente de componentes funcionais descritos em termos de seus comportamentos, interfaces e interconexões entre componentes.” **Hayes-Roth, 1994**
- “Arquitetura de software é o estudo da estrutura em larga escala e desempenho de sistemas de software” **Lane, 1990**
- “A arquitetura de um sistema complexo é seu estilo e método de projeto e construção” **Hayes-Roth, 1995**
- “...estrutura de suporte de um sistema” **Rechtin, 1992**
- “...estrutura de sistema que consiste de módulos ativos, um mecanismo para permitir interação entre estes módulos e um conjunto de regras que governam a interação” **Boasson, 1995**

# O QUE É ARQUITETURA DE SOFTWARE?

## Muitas definições...



- “...conjunto de elementos arquiteturais (ou de projeto) que têm uma forma particular.” **Perry and Wolf, 1992**
- “...estrutura(s) do sistema, que compreende(em) os componentes de software, as propriedades externamente visíveis desses componentes, e os relacionamentos entre eles.” **Clements et al., 1997**
- “...especificação abstrata do sistema consistindo primeiramente de componentes funcionais descritos em termos de seus comportamentos, interfaces e interconexões entre componentes.” **Hayes-Roth, 1994**
- “Arquitetura de software é o estudo da estrutura em larga escala e desempenho de sistemas de software” **Lane, 1990**
- “A arquitetura de um sistema complexo é seu estilo e método de projeto e construção” **Hayes-Roth, 1995**
- “...estrutura de suporte de um sistema” **Rechtin, 1992**
- “...estrutura de sistema que consiste de módulos ativos, um mecanismo para permitir interação entre estes módulos e um conjunto de regras que governam a interação” **Boasson, 1995**

# O QUE É ARQUITETURA DE SOFTWARE?

## Resumindo...

- Elementos em comum...
  - Descrição dos componentes principais
  - Relacionamentos e interações entre componentes
  - Omite informação sobre o conteúdo dos componentes não relacionada a suas interações
  - O comportamento dos componentes é uma parte da arquitetura enquanto possa ser discernido do ponto de vista de outro componente
  - A arquitetura define uma lógica por trás dos componentes e da estrutura

# O QUE É ARQUITETURA DE SOFTWARE?

## Rational Software

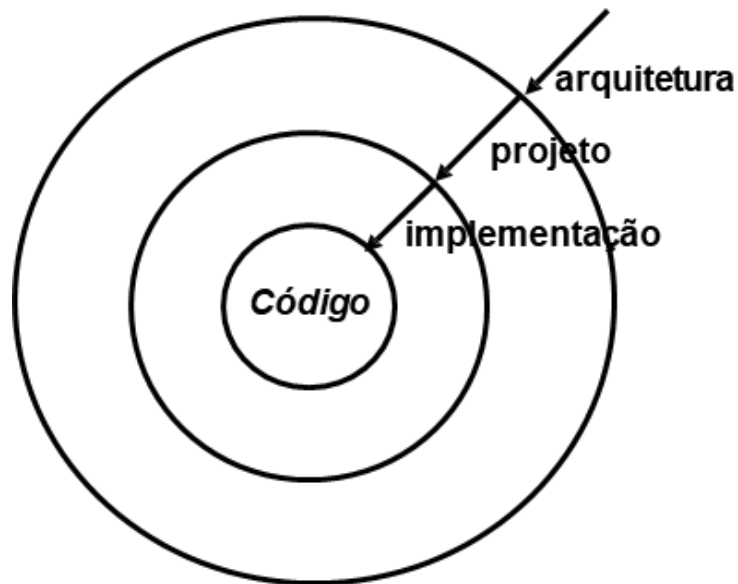
- Arquitetura de Software também envolve
  - Funcionalidade
  - Usabilidade
  - Desempenho
  - Reúso
  - Facilidade de compreensão
  - Restrições e equilíbrio de fatores econômicos e tecnológicos



# O QUE É ARQUITETURA DE SOFTWARE?

## Restringindo Projeto e Implementação

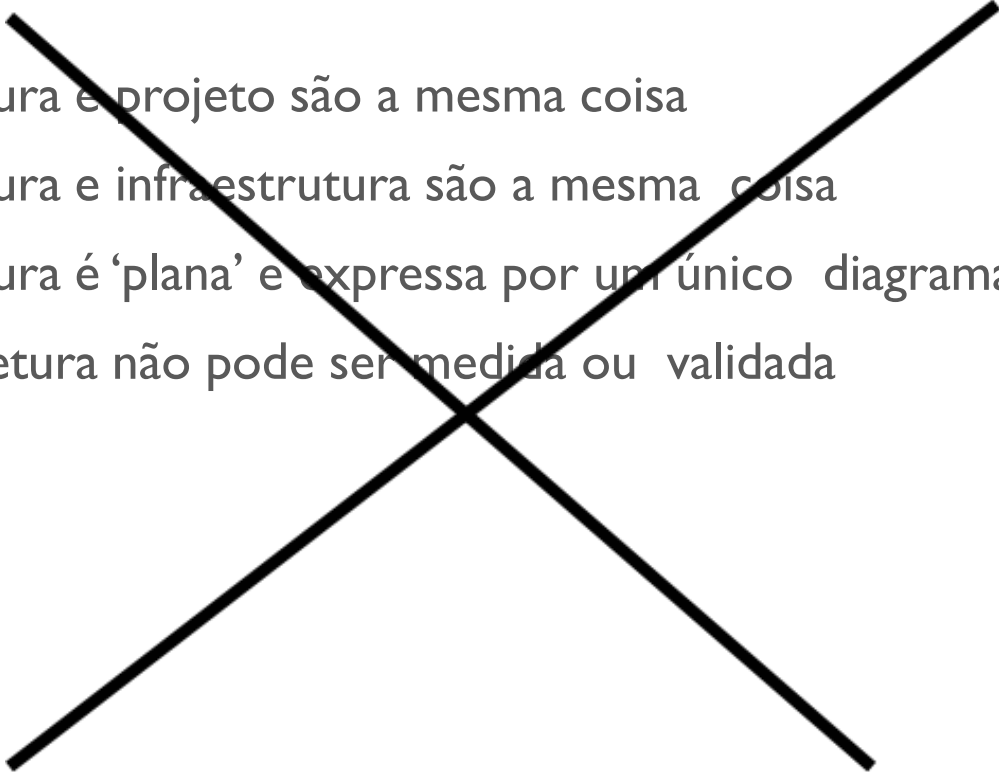
- A arquitetura envolve um **conjunto de decisões de design**, **regras ou padrões** que restringem o **projeto e a implementação**



- *As decisões de arquitetura são as **mais fundamentais** e **alterá-las provoca efeitos colaterais** significativos.*

# O QUE É ARQUITETURA DE SOFTWARE?

## Concepções errôneas

- 
- Arquitetura e projeto são a mesma coisa
  - Arquitetura e infraestrutura são a mesma coisa
  - Arquitetura é 'plana' e expressa por um único diagrama
  - A arquitetura não pode ser medida ou validada



# O QUE É ARQUITETURA DE SOFTWARE?

## Concepções errôneas

- **Arquitetura e projeto são a mesma coisa**
- A arquitetura **é um aspecto do projeto**, focada em
  - elementos que são importantes estruturalmente
  - elementos que têm impacto significativo em desempenho, confiabilidade, custo, adaptabilidade, etc.
- A arquitetura **não diz respeito ao projeto detalhado** de componentes individuais

# O QUE É ARQUITETURA DE SOFTWARE?

## Concepções errôneas

- **Arquitetura = infraestrutura**
- Arquitetura aborda...
  - Aspectos dinâmicos
  - Argumentação lógica
  - Adequação ao contexto
    - Contexto de negócio
    - Contexto de desenvolvimento

# O QUE É ARQUITETURA DE SOFTWARE?

## Concepções errôneas

- Arquitetura é ‘plana’ e expressa por um único diagrama
- A arquitetura é plana apenas em casos muito triviais
- A arquitetura tem muitas dimensões, que representam **múltiplas questões de múltiplos stakeholders**
- Usar um único diagrama para representar todas as dimensões da arquitetura leva à **sobrecarga semântica** (leia-se “confusão”)
- Arquitetura requer **múltiplas visões**

# O QUE É ARQUITETURA DE SOFTWARE?

## Concepções errôneas

- **A arquitetura não pode ser medida ou validada**
- Arquiteturas podem ser sistematicamente avaliadas contra **requisitos de qualidade e funcionais**, riscos e atributos chave do sistema
  - **Revisando** os artefatos da arquitetura
  - **Testando** os protótipos da arquitetura

# MOTIVAÇÃO E BENEFÍCIOS

## Por Que Arquitetar?

- Integridade e qualidade do sistema
- Controle da complexidade
- Previsibilidade
- Testabilidade
- Reúso
- Comunicação
- Organização e gerência de projetos

# MOTIVAÇÃO E BENEFÍCIOS

## Integridade e qualidade do sistema

- A arquitetura propicia **integridade conceitual e qualidade ao sistema**, ao mesmo tempo em que permite mudanças
  - Funcionalidade requerida
  - Atributos de qualidade
  - Novos requisitos
  - Requisitos modificados
  - Tendências tecnológicas
- A arquitetura propicia **longevidade ao sistema**
  - Facilidade de evolução
  - Escalabilidade

# MOTIVAÇÃO E BENEFÍCIOS

## Controle da complexidade

- **Dividir para conquistar**
  - Decomposição em componentes
  - Detalhes de implementação ocultos
- **Separação de áreas de interesse**
  - Componentes (e camadas) encapsulam detalhes
  - Componentes distintos podem ser implementados por pessoas com especialidades diferentes

# MOTIVAÇÃO E BENEFÍCIOS

## Previsibilidade

- **Previsibilidade do processo**
  - Protótipos da arquitetura permitem coletar métricas
    - Métricas de custo do desenvolvimento
    - Métricas de cronograma
- **Previsibilidade comportamental**
  - Iterações da arquitetura removem maiores riscos



# MOTIVAÇÃO E BENEFÍCIOS

## Testabilidade

- Sistemas adequadamente **componentizados** suportam maior e melhor
  - Diagnóstico de problemas
  - Rastreabilidade
  - Detecção de erros

# MOTIVAÇÃO E BENEFÍCIOS

## Reúso

- A arquitetura define **regras de substituição**
- **Interfaces entre componentes** definem limites de substituição
- A arquitetura propicia **diferentes granularidades** de reúso
  - Reúso em pequena escala, no nível de componentes
  - Reúso em larga escala
    - Subsistemas
    - Produtos
    - Frameworks

# MOTIVAÇÃO E BENEFÍCIOS

## Comunicação

- A arquitetura auxilia a comunicação entre stakeholders
- Visões distintas endereçam questões de stakeholders distintos
- A arquitetura comunica decisões chave de projeto

# MOTIVAÇÃO E BENEFÍCIOS

## Organização e gerência de projetos

- Estrutura organizacional alinhada com a arquitetura
  - Equipe(s) de desenvolvimento
  - Subcontratados
- Componentes / subsistemas utilizados como unidades para
  - Desenvolvimento
  - Gerência de configuração
  - Teste
  - Implantação e evolução
- Componentes / subsistemas desenvolvidos concorrentemente

# MOTIVAÇÃO E BENEFÍCIOS

## Evitar ou reduzir custos

- Minimizar o custo de (re)trabalho
- Economias com o reúso
  - Reúso de componentes
  - Desenvolvimento de componentes reutilizáveis
- Economias com a utilização efetiva de recursos do projeto
- Economias com estimativas de custo / cronograma mais precisas
- Economias com manutenção aprimoradas

# MOTIVAÇÃO E BENEFÍCIOS

## Evitar ou reduzir custos

- O investimento em arquitetura ocorre nas fases iniciais do desenvolvimento
  - Concepção
  - Elaboração
- Modelos de estimativa de custo são imprecisos nas fases iniciais
- Os benefícios da arquitetura ocorrem durante a implementação e manutenção

# MOTIVAÇÃO E BENEFÍCIOS

## Evitar ou reduzir custos

- A gerência sempre é muito relutante em bancar custos nas fases iniciais do projeto
- Arquiteturas robustas são mais críticas se a complexidade do projeto, criticalidade e tamanho são grandes
- Boa arquitetura ajuda a estimar e a controlar os custos do projeto

*A longo prazo, não arquitetar o sistema pode ser muito caro*

# CONCEITOS RELACIONADOS À ARQUITETURA

- Acoplamento
- Coesão
- Interface
- Componentes
- Conectores
- Padrão Arquitetural
- Estilo arquitetural
- Visão arquitetural



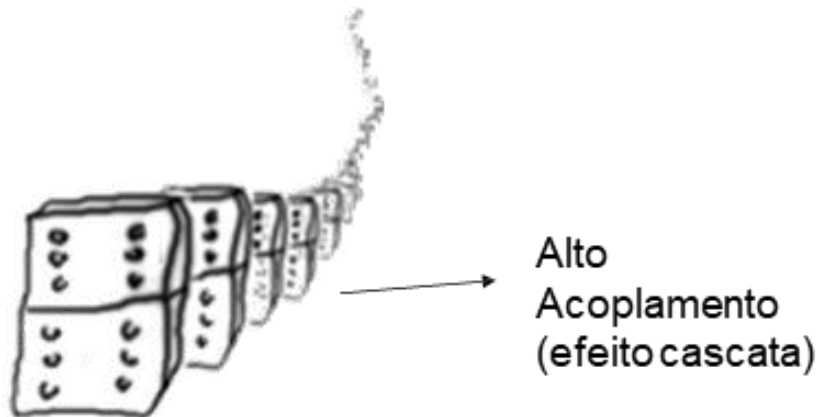
# CONCEITOS RELACIONADOS À ARQUITETURA

- **Acoplamento**
- Coesão
- Interface
- Componentes
- Conectores
- Padrão Arquitetural
- Estilo arquitetural
- Visão arquitetural

# CONCEITOS

## ACOPLAMENTO

- Grau de interconexão entre diferentes pedaços de um sistema
- Pedaços menos acoplados são mais fáceis de entender, testar, reusar e manter
- Baixo acoplamento também promove o paralelismo de implementação



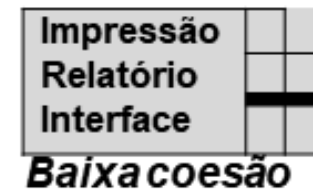
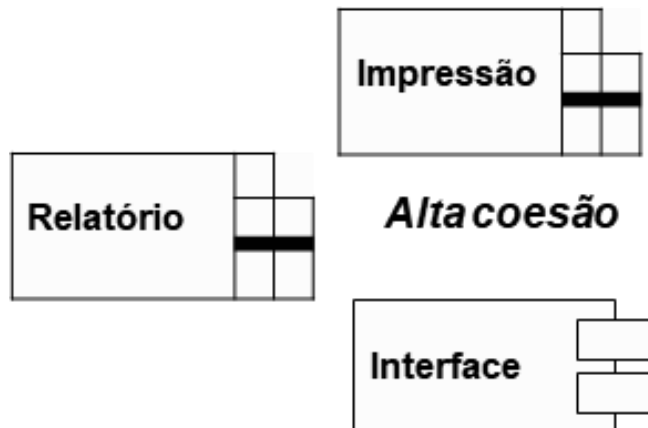
# CONCEITOS RELACIONADOS À ARQUITETURA

- Acoplamento
- **Coesão**
- Interface
- Componentes
- Conectores
- Padrão Arquitetural
- Estilo arquitetural
- Visão arquitetural

# CONCEITOS

## COESÃO

- Quão proximamente são relacionadas as atividades dentro de um único pedaço (componente) ou entre um grupo de pedaços?
  - Componentes altamente coesos = relacionados a apenas UMA funcionalidade

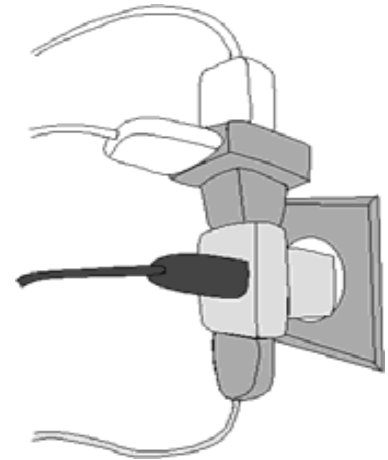


# CONCEITOS RELACIONADOS À ARQUITETURA

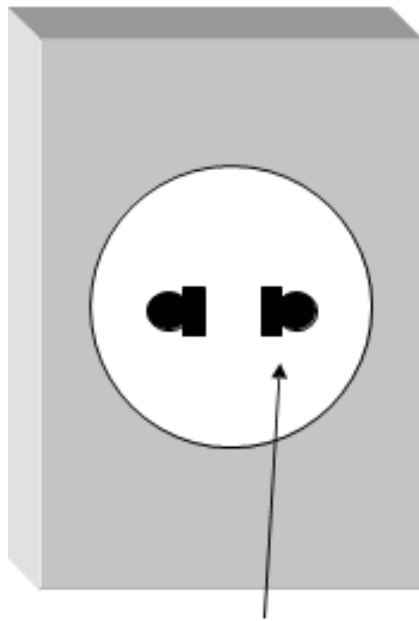
- Acoplamento
- Coesão
- **Interface**
- Componentes
- Conectores
- Padrão Arquitetural
- Estilo arquitetural
- Visão arquitetural

# CONCEITOS INTERFACE

- As maneiras em que subsistemas dentro de um projeto maior interagem são claramente definidas.
- Idealmente, interações são especificadas de um modo que possam se manter relativamente estáveis ao longo do ciclo de vida do sistema.
- Um modo de alcançar isso é através de abstrações sobre a implementação concreta.



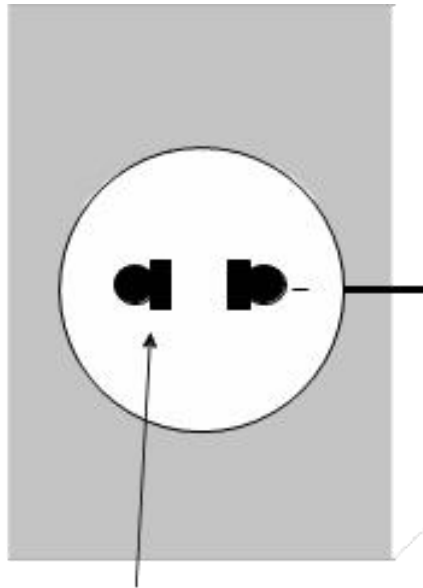
# CONCEITOS INTERFACE



Interface bem definida



# CONCEITOS INTERFACE



Abstração sobre...



... a implementação concreta



# CONCEITOS RELACIONADOS À ARQUITETURA

- Acoplamento
- Coesão
- Interface
- **Componentes**
- Conectores
- Padrão Arquitetural
- Estilo arquitetural
- Visão arquitetural

# CONCEITOS

## COMPONENTES

- Entidade da arquitetura que:
  - Encapsula um subconjunto de funcionalidades ou dados do sistema.
  - Restringe o acesso a esse subconjunto por meio de uma interface.
  - Possui um contexto necessário para sua execução.
- Pode ser simples ou complexo – depende das decisões da arquitetura.
- Contexto necessário de um componente inclui:
  - Interface necessária para fornecer serviços a outros componentes
  - Disponibilidade de recursos, como dados.
  - Softwares necessários – linguagem de programação, protocolos, drivers, etc.
  - Configurações de hardware.

# CONCEITOS RELACIONADOS À ARQUITETURA

- Acoplamento
- Coesão
- Interface
- Componentes
- **Conectores**
- Padrão Arquitetural
- Estilo arquitetural
- Visão arquitetural

# CONCEITOS

# CONECTORES

- Veículos de comunicação entre componentes
  - Descrição das interações entre componentes
- Exemplos de conectores:
  - Chamadas de procedimento
  - Anúncios de eventos
  - Protocolos de comunicação
  - Invocações de serviços

# CONCEITOS RELACIONADOS À ARQUITETURA

- Acoplamento
- Coesão
- Interface
- Componentes
- Conectores
- **Padrão Arquitetural**
- **Estilo arquitetural**
- Visão arquitetural

# CONCEITOS

## PADRÃO ARQUITETURAL X ESTILO ARQUITETURAL

- Um padrão de arquitetura é um conjunto de decisões de arquitetura que são aplicáveis em um **problema de design recorrente**.
  - Pode ser aplicada a softwares em diferentes contextos nos quais o problema ocorre.
  - Exemplo: sistema 3-camadas para operações de processamento, armazenamento e busca.
- Um estilo de arquitetura é uma **coleção de decisões de arquitetura** que:
  - São aplicadas em um **determinado contexto**
  - Restringe decisões de projeto da arquitetura que são específicas a um determinado software.
  - Exemplo: separar camada cliente e camada servidor.

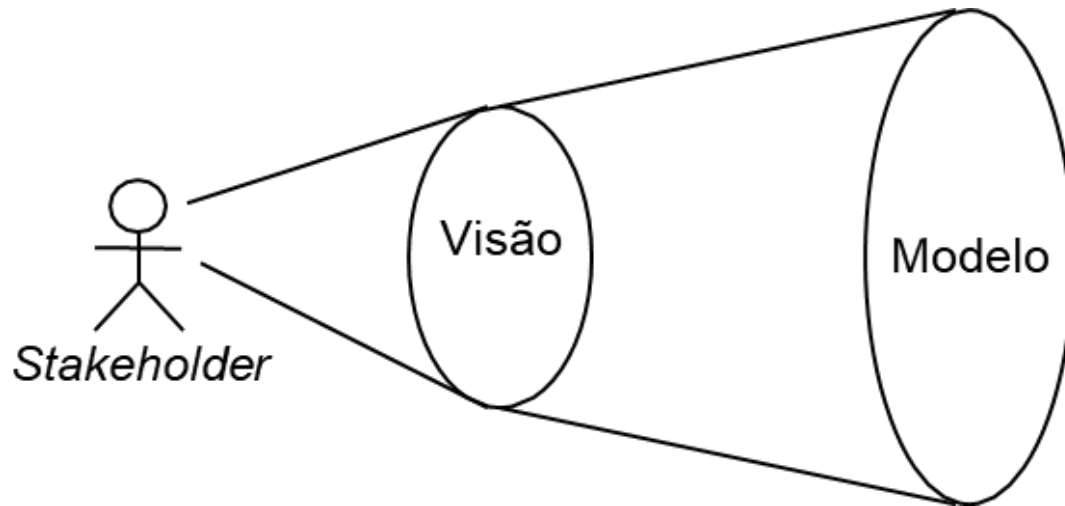
# CONCEITOS RELACIONADOS À ARQUITETURA

- Acoplamento
- Coesão
- Interface
- Componentes
- Conectores
- Padrão Arquitetural
- Estilo arquitetural
- **Visão arquitetural**

# CONCEITOS

## VISÃO ARQUITETETURAL

- Projeção de um modelo da arquitetura sob determinadas perspectivas
  - Diferentes visões para diferentes stakeholders





# CONCEITOS

## VISÃO ARQUITETURAL

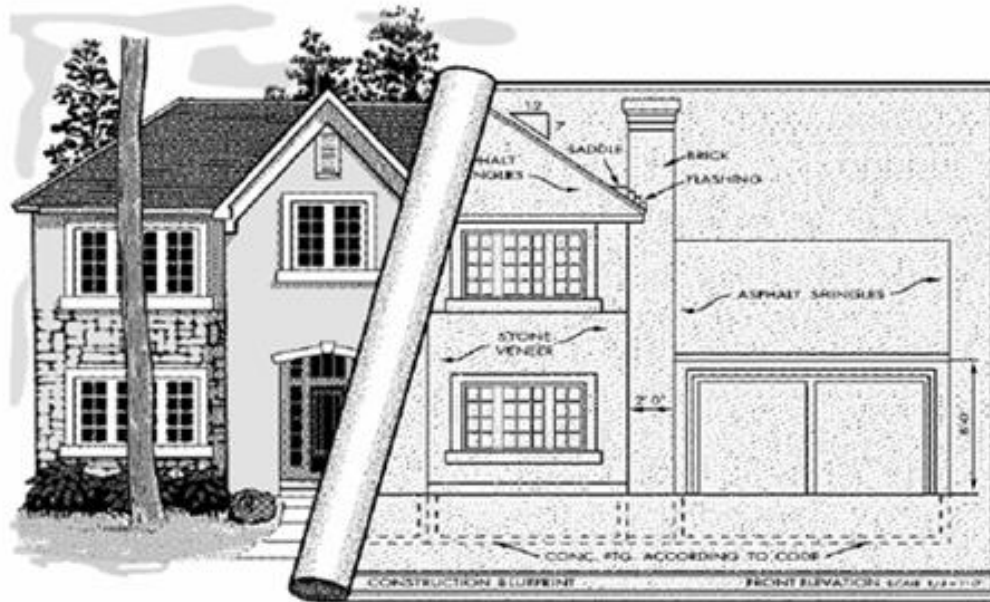
- Modelo
  - Uma descrição completa de um sistema sob uma perspectiva particular e um nível específico de abstração
- Visão
  - Uma projeção de um modelo, que é visto sob uma determinada perspectiva, e omite entidades que não são relevantes a essa perspectiva

# CONCEITOS

## VISÃO ARQUITETURAL



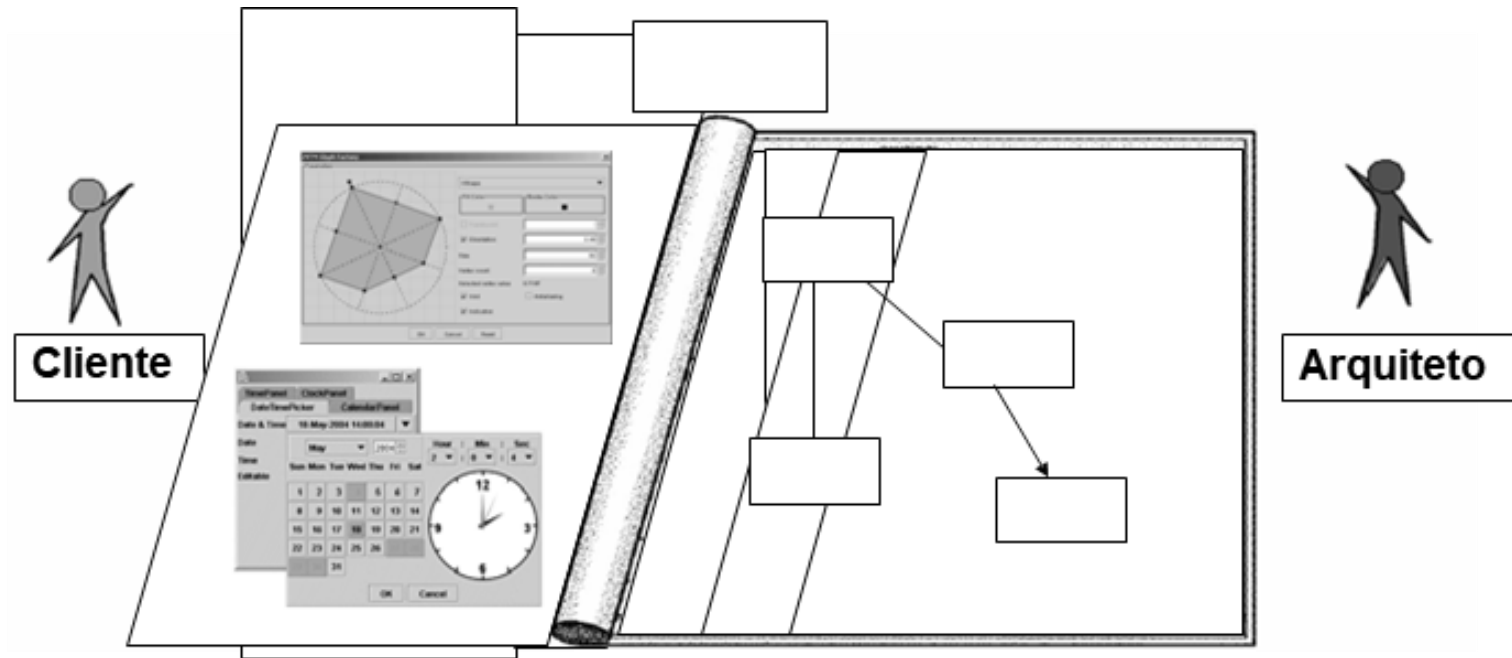
Cliente



Arquiteto

# CONCEITOS

## VISÃO ARQUITETURAL



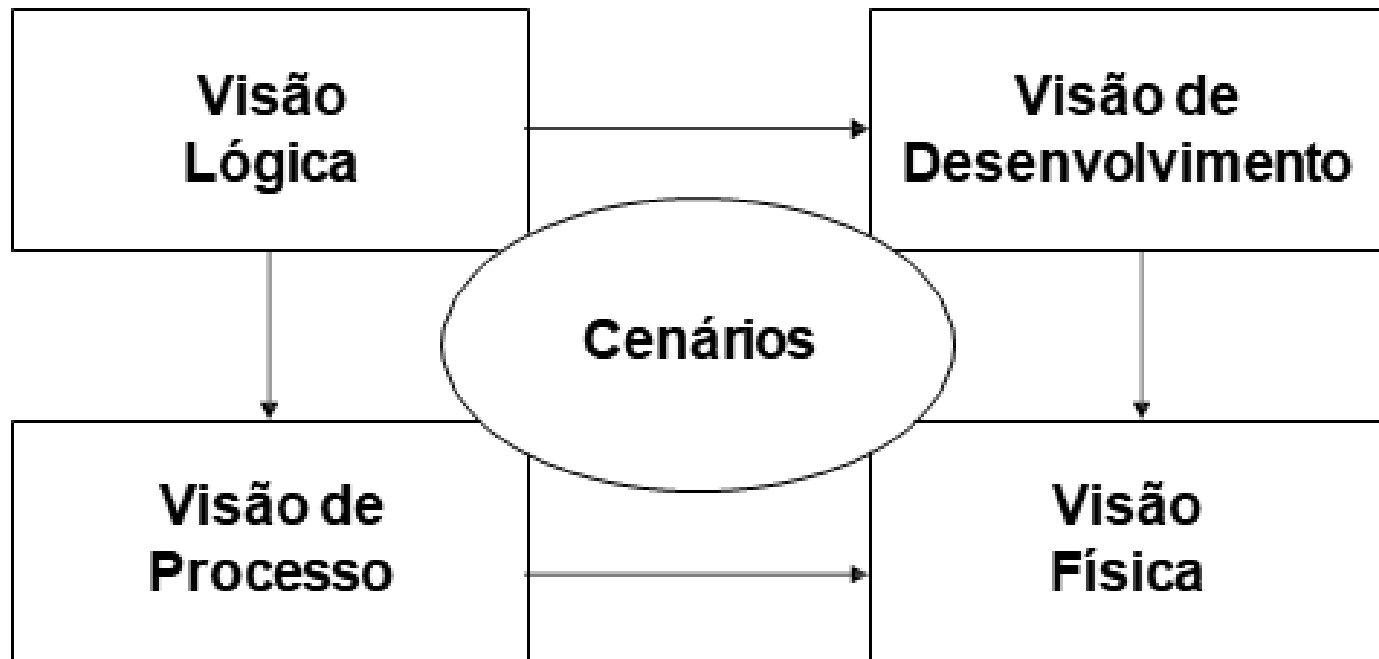
# PARA PENSAR...

## VISÕES ARQUITETURAIS

- Que modelos você utilizaria para os seguintes stakeholders:
  - Arquiteto
  - Desenvolvedor
  - Cliente
  - Analista de Teste

# VISÕES ARQUITETURAIS

- Modelo “4+1” (*Rational Software*)



# VISÕES ARQUITETURAIS

## LÓGICA

- Visão Lógica
  - “Retrato estático” dos relacionamentos existentes entre as entidades do sistema
  - Pode possuir duas ou mais representações, dentre elas, uma conceitual e outra de esquema de banco de dados

**Visão  
Lógica**

# VISÕES ARQUITETURAIS

## PROCESSO

- Visão de Processo
  - Descreve aspectos de sincronização e concorrência
  - Descrição de processos concorrentes
  - Diferentes linhas de execução (threads), entidades ativas

**Visão de  
Processo**

# **VISÕES ARQUITETURAIS**

## **DESENVOLVIMENTO**

- Visão de Desenvolvimento
  - Descreve a organização do software em seu ambiente de desenvolvimento
    - Componentes
    - Linguagens

**Visão de  
Desenvolvimento**



# **VISÕES ARQUITETURAIS**

## **FÍSICA**

- Visão Física
  - Descreve o mapeamento do software para o hardware
  - Distribuição de componentes
  - Também chamada “deployment”

**Visão  
Física**

# VISÕES ARQUITETURAIS

## CENÁRIOS (+1)

- Cenários (+1)
  - Cenários de funcionamento do sistema diretamente ligados à arquitetura
  - Principais casos de uso
  - Lembram de RUP?
  - Centrado em arquitetura!



# BIBLIOGRAFIA RECOMENDADA

- Bibliografia Básica:
- 1. TAYLOR, R. N.; MEDVIDOVIC, N.; DASHOFTY, E. M. Software architecture: Foundations, Theory, and Practice. Wiley, 2009. 750 p.
- 2. CLEMENTS, Paul et al. Documenting software architectures: views and beyond. 2. ed. Massachusetts: Addison-Wesley Professional. 2010. 592 p.
- 3. BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. Software architecture in practice. 2. ed. Boston, Massachusetts: Addison-Wesley Professional, 2003. 560 p.

# BIBLIOGRAFIA RECOMENDADA

- Bibliografia Complementar:
- 1. SHAW, Mary; GARLAN, David. Software architecture: perspectives on an emerging discipline. São Paulo: Prentice Hall. 1996. 242 p.
- 2. SOMMERVILLE, I. Engenharia de software. 7. ed. São Paulo: Pearson Addison-Wesley, 2007.
- 3. GORTON, Ian. Essential software architecture. Berlin: Springer, 2006. 283 p. ISBN 3540287132 (enc.).
- 4. REEKIE, John. A software architecture primer. Sydney, Australia: Angophora Press, 2006. 179 p. ISBN 0646458418 (broch.).
- 5. BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2. ed. Rio de Janeiro: Elsevier, 2007.

# DÚVIDAS?

