



Pós-Graduação Engenharia de Software

Arquitetura de Software e Padrões de Projeto

Aula 01

Prof. Msc Rogério Augusto Rondini
rarondini.paradygma@gmail.com



Conteúdo

- Introdução a arquitetura de Software
 - Conceitos principais
 - Atributos de qualidade e requisitos não-funcionais
 - Modelagem Arquitetural
 - Diagramas UML
 - Visão 4+1
 - Estudo de Caso
 - Apresentação do Estudo de Caso
 - Atividade 01
-



Introdução a Arquitetura de Software



Brainstorming

- O que você entende por Arquitetura de Software ?
- Existe uma área específica de arquitetura dentro da sua empresa? Qual o papel desta área
- Em sua opinião, qual o papel de um arquiteto de software?

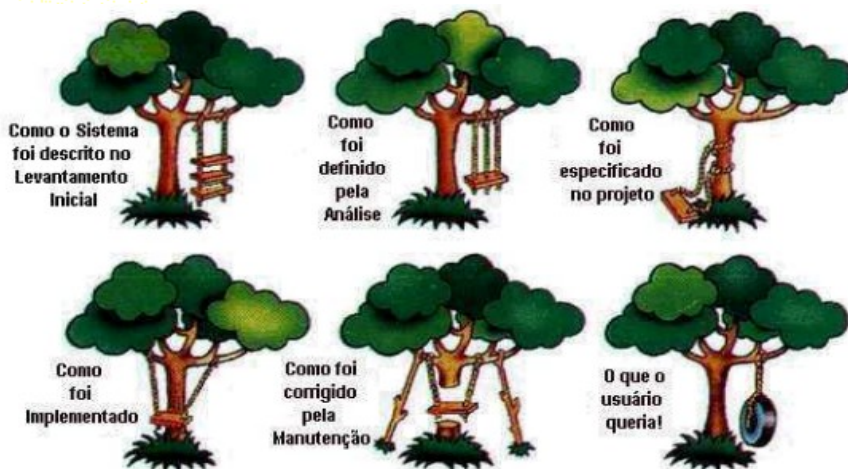
<http://www.youtube.com/watch?v=U1n8bFHz9Y8>

<http://www.infoq.com/br/news/2010/09/real-papel-arquiteto>

http://www.wthreex.com/rup/process/workers/wk_archt.htm



Arquitetura





Arquitetura de Software

- Arquitetura de Software compreende um conjunto de decisões (***técnicas e de negócio***) sobre a organização de um software
 - Escolha do estilo arquitetural que guiará o projeto
 - Definição dos elementos e suas interfaces e relacionamentos
 - Decisões tecnológicas



Arquitetura de Software

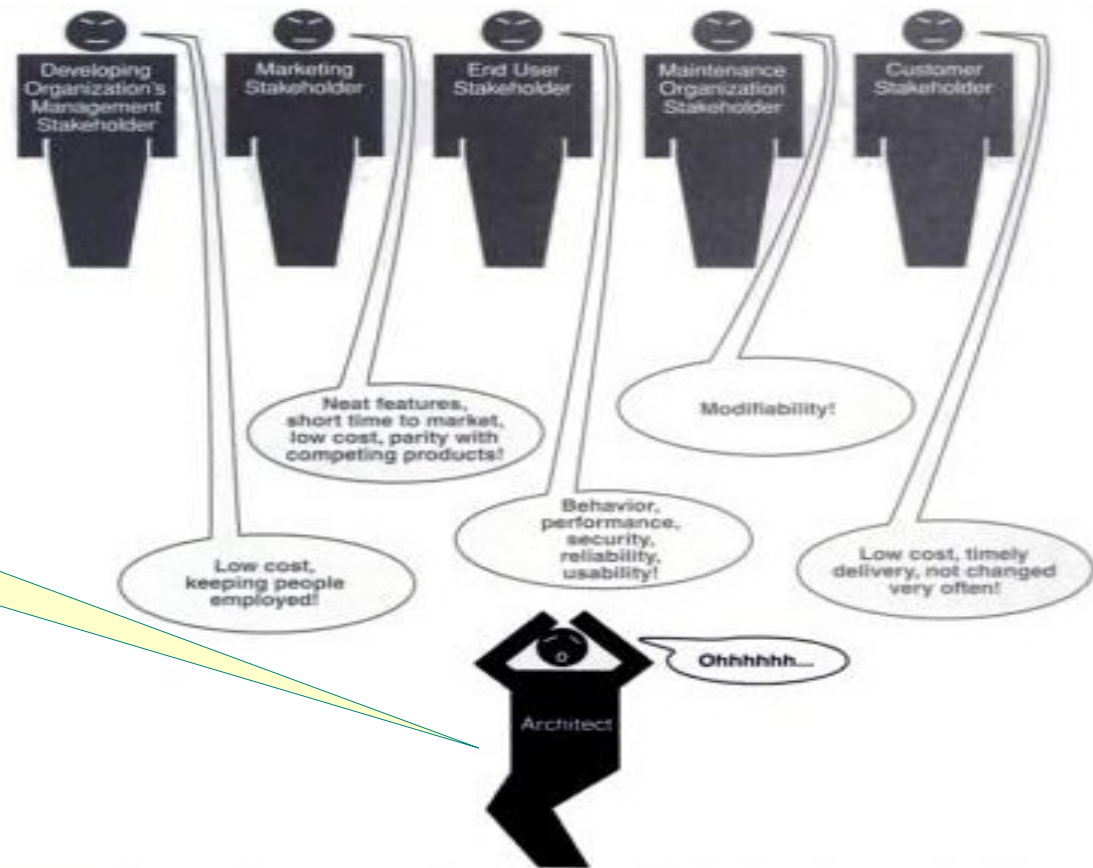
- Definição

- Estrutura do sistema em termos de elementos de software, suas propriedades externas e seus relacionamento internos e com outros softwares

*(Bass, Clements e
Kazman)*

Arquitetura de Software

Arquitetura é totalmente influenciada pelos envolvidos no projeto

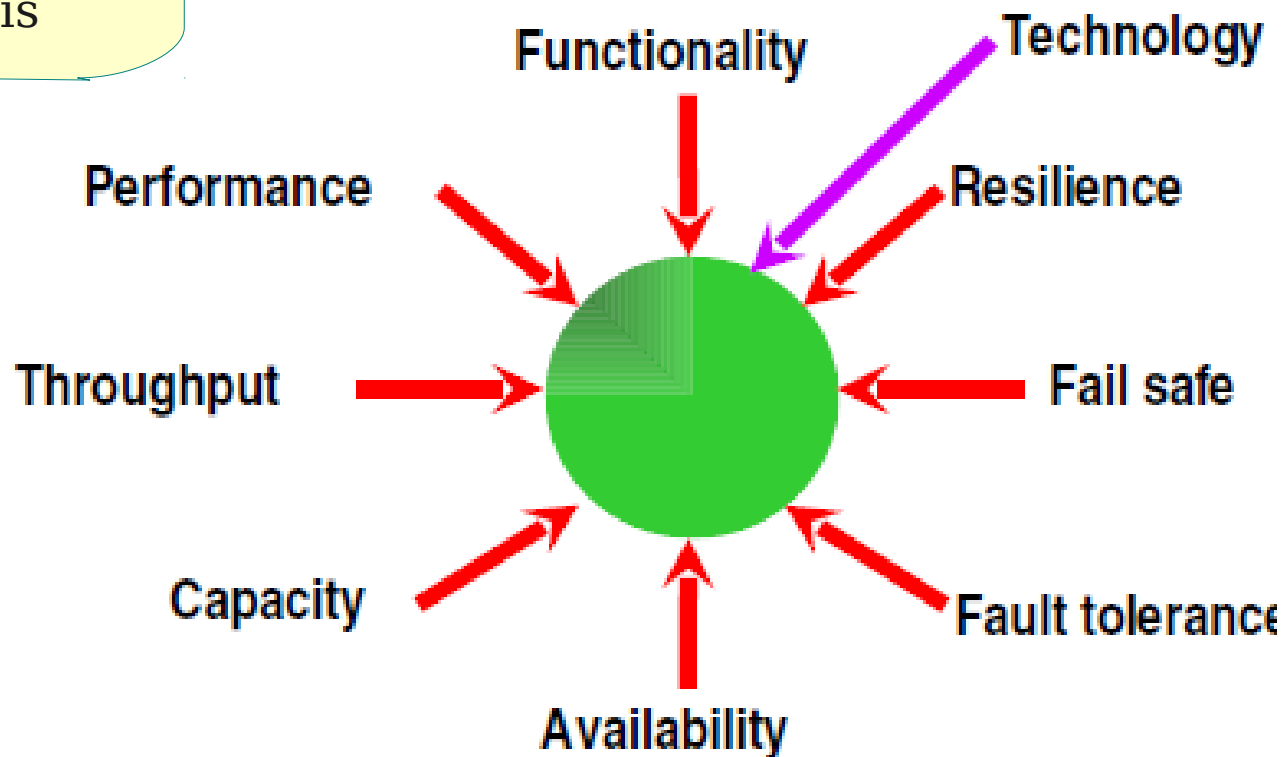


Ref: Software Architecture in Practice (ver Referências)



Arquitetura de Software

Baseada nos requisitos funcionais, porém, deve resolver requisitos não funcionais



Ref: www.cs.toronto.edu/~wl/teach/407/w11a.pdf



Arquitetura de Software

- Outros fatores que influenciam a arquitetura

- Ambiente tecnológico

- Tendência em seguir o ambiente tecnológico da empresa.

- Ex: utilizar *front-end* Apache HTTP Server ou BIG-IP

- Experiência dos arquitetos

- Tendência em seguir experiências passadas, é “mais seguro”.



Arquitetura de Software

- Qual a sua importância ?
 - Comunicação → arquitetura representa uma abstração comum do sistema (*trataremos das visões arquiteturais*)
 - Decisão inicial de projeto → Decisões que serão adotadas durante o projeto
 - Reuso → Arquitetura de software constitui um modelo de como o software é estruturado, modelo este que pode ser reproduzido para aplicações com atributos de qualidade e requisitos não-funcionais semelhantes
-



Estudo de Caso

Sistema de Ponto de Venda



Estudo de Caso

- Considerações (resumo)
 - Produto de software desenvolvido por uma “*software house*” com objetivo de vender para comércios de diferentes seguimentos
 - Mesmo antes de ser desenvolvido, o software foi vendido para uma grande rede de supermercados, com dois marcos importantes:
 - M1 → Instalar o produto nos caixas das lojas
 - M2 → Aproveitar as principais funcionalidade do PDV para montar um site Web e uma estrutura de *Delivery*



Requisitos Não Funcionais

Atributos de Qualidade

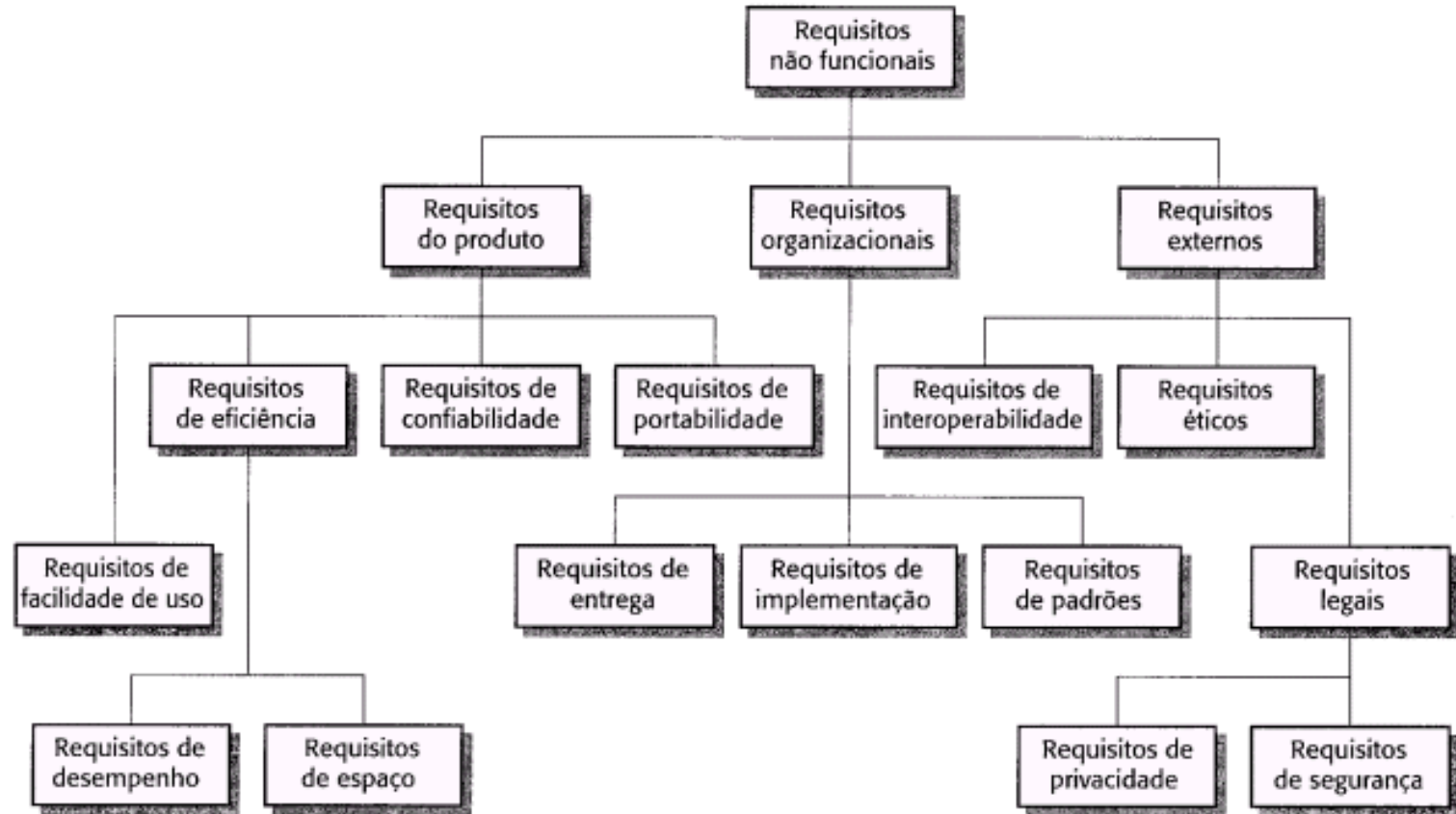


Requisitos Não Funcionais

Requisitos que não dizem respeito diretamente a funcionalidade do sistema mas expressam propriedades do sistema e/ou restrições sobre os serviços ou funções providas pelo sistema. Podem ser categorizados em:

- Requisitos de Produto
- Requisitos Organizacionais
- Requisitos Externos

Requisitos Não Funcionais



Sommerville, Ian. Engenharia de Software. 6ª edição.



Requisitos Não Funcionais

- Usabilidade/Facilidade de Uso
 - esforço para utilizar, aprender o produto)
 - Confiabilidade
 - frequência de falhas, recuperabilidade
 - Eficiência
 - Desempenho (*o que é desempenho ?*)
 - Espaço
 - Portabilidade
 - capacidade de transferir o produto para outros ambientes
-



Requisitos Não Funcionais

- Implementações
 - Ex: regras de codificação
- Padrões
 - Ex: linguagem de programação
- Legais
 - Políticas de privacidade, direitos autorais
- Interoperabilidade
 - Capacidade de integração com outros sistemas



Atributos de Qualidade

- Atributos de qualidade é o termo utilizado pela arquitetura de software para referenciar requisitos não funcionais
 - Desempenho
 - Disponibilidade
 - Testabilidade
 - Segurança
 - Usabilidade
 - ...



Atributos de Qualidade

- São ortogonais às funcionalidades
 - Podem afetar vários serviços ou funções
 - São independentes da funcionalidade
 - Funcionalidades não devem ser modificadas por atributos de qualidade



Atributos de Qualidade

- Envolve aspectos arquiteturais e não arquiteturais, vejamos:
 - Modificabilidade (facilidade de manutenção)
 - ***Estilo e padronização de código*** é um aspecto não arquitetural que afeta a facilidade de manutenção
 - Componentes e módulos com ***alta coesão e baixo acoplamento*** é um aspecto arquitetural que também afeta a facilidade de manutenção



Atributos de Qualidade

- Para se atingir atributos de qualidade, é necessário ter uma **estratégia arquitetural**
- Estratégia Arquitetural se refere a um **conjunto de táticas** ou **decisões de projeto**
- Exemplo → Táticas para recuperação de falhas
 - Redundância Ativa
 - Redundância Passiva



Análise Arquitetural



Análise Arquitetural

- Preocupada em identificar e resolver requisitos não funcionais do sistema
 - Deve começar antes mesmos da primeira iteração de desenvolvimento, pois alguns problemas devem ser tratados logo no início
 - Sistema deve tratar 500 transações concorrentes com tempo de resposta de 1 segundo
-



Análise Arquitetural

- Dois pontos importantes que devem ser observados, além dos requisitos não funcionais/atributos de qualidade

– Pontos de Variação

- Variação no sistema existente ou requisitos que afetam a arquitetura
- Ex: Interfaces múltiplas, regras de negócio variam para cada cliente

– Pontos de Evolução

- Variações que podem surgir no futuro, mas não estão presentes nos requisitos existentes



Análise Arquitetural

- Passos comuns na análise arquitetural
 - Identificação e análise dos requisitos não funcionais
 - Para os requisitos com impacto arquitetural significativo, analisar alternativas ou criar soluções que resolvam o impacto
 - Normalmente, requisitos não funcionais de alto impacto são, confiabilidade, desempenho, usabilidade e interfaces.



Análise Arquitetural

- Documentação dos fatores
 - cenários de qualidade e
 - Afirmações curtas do tipo estímulo/resposta
 - Quando o usuário realizar uma busca de CEP, o sistema deve responder em até 1 segundo.
 - Variabilidade
 - Flexibilidade atual e evolução
 - Atualmente o sistema terá no máximo 10 usuários simultâneos, com previsão de atingir 100 usuários simultâneos em 2 anos



Análise Arquitetural

- Documentação dos fatores
 - Impacto
 - Com 10 usuários simultâneos teremos quase nenhum impacto no tempo de resposta.
 - Prioridade
 - Média



Modelagem Arquitetural



Modelagem Arquitetural

- Um dos principais objetivos da modelagem é descrever elementos de software em uma linguagem padronizada e de fácil compreensão
- Dessa forma, a principal linguagem de modelagem existente atualmente é a UML
- Alguns diagramas UML são extremamente úteis para modelagem arquitetural



Modelagem Arquitetural

- Além dos diagramas tradicionais, de classe e sequência, três outros diagramas são fundamentais para modelagem arquitetural
 - Diagrama de pacotes
 - Utilizado para representar arquitetura lógica (pacotes e camadas) de um sistema
 - Diagrama de componentes
 - Utilizado para representar a divisão de componentes de software seus relacionamentos
 - Diagrama de implantação
 - Utilizado para representar estrutura física de implantação do software



Diagrama de Pacotes

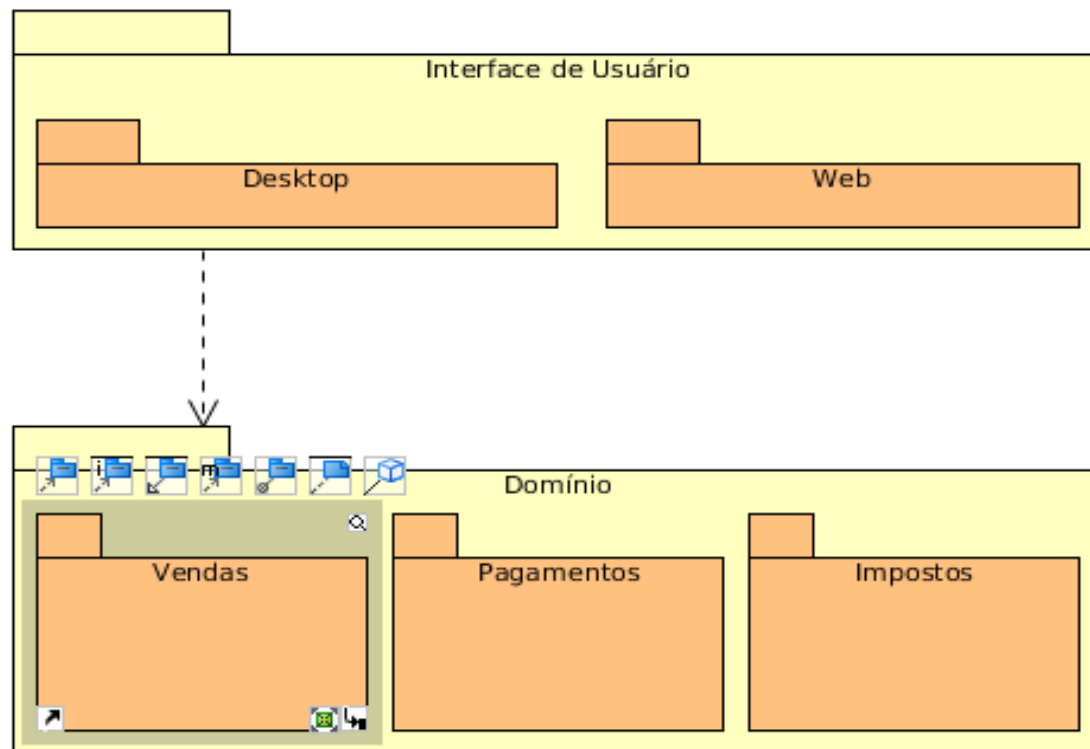




Diagrama Componentes

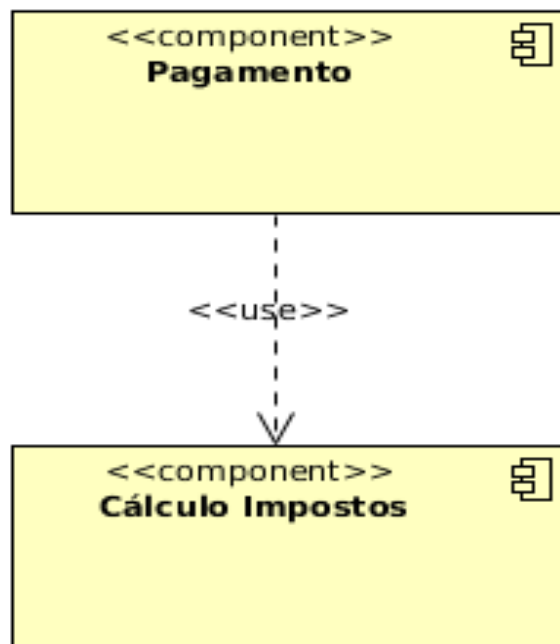




Diagrama de Implantação

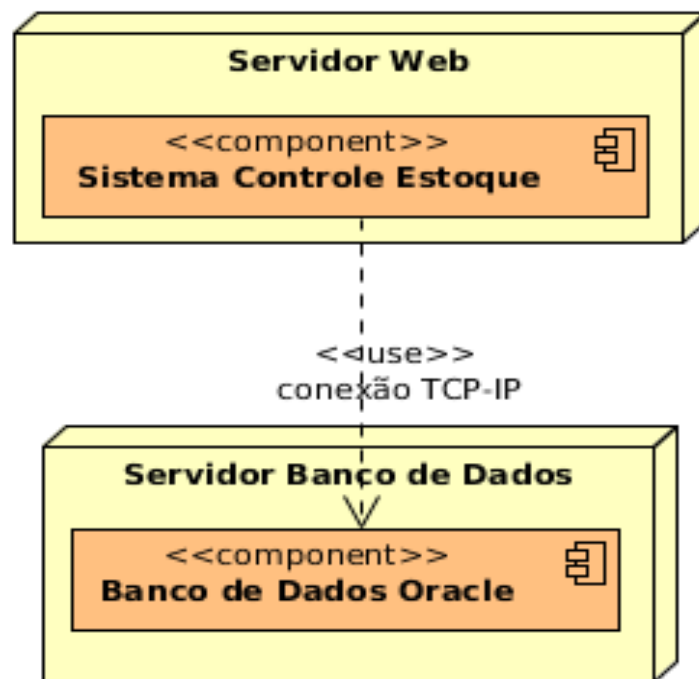
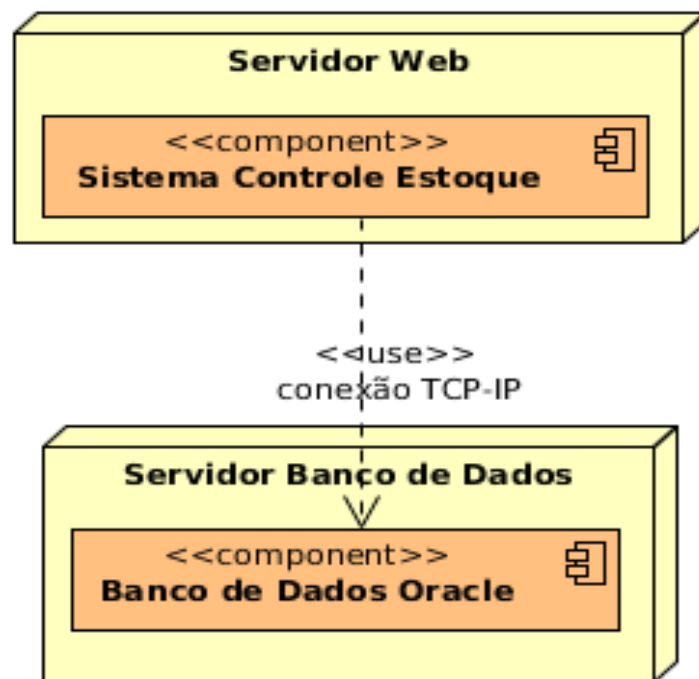




Diagrama de Implantação



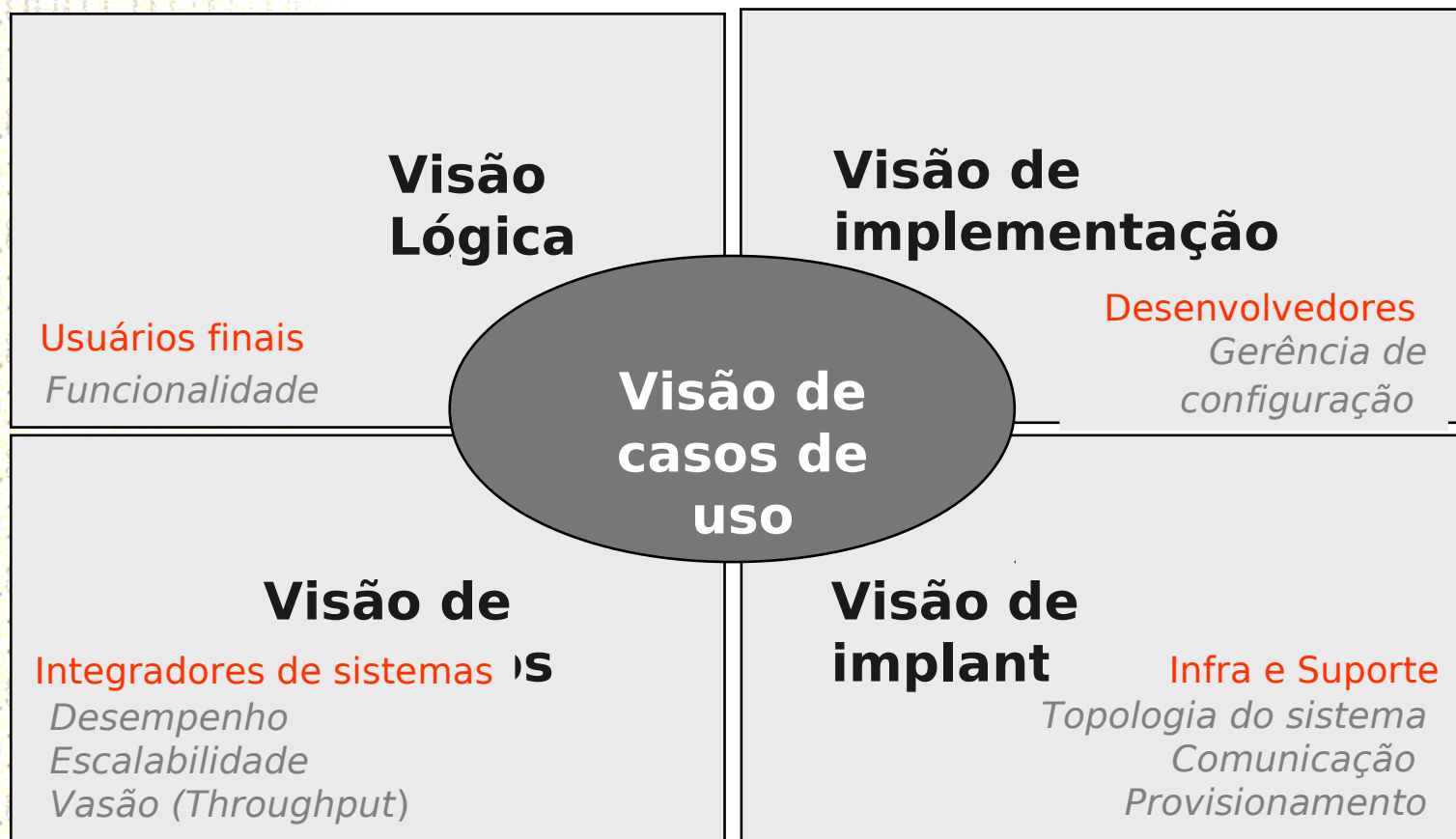


Visões Arquiteturais

Modelo 4+1



Visão 4+1



◀lógico

físico▶



Visão 4 + 1

- Visão de Caso de Uso

Deve mostrar um subconjunto
significante para definição de
solução arquitetural

O Modelo de Caso de Uso



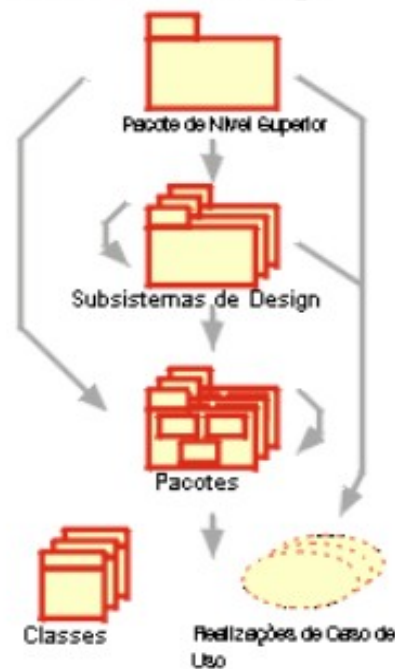


Visão 4 + 1

- Visão Lógica

Visão de pacotes, componentes, camadas, realização de caso de uso e classes, referentes aos casos de uso selecionados na Visão de Caso de Uso

O Modelo de Design





Visão 4 + 1

- Visão Lógica

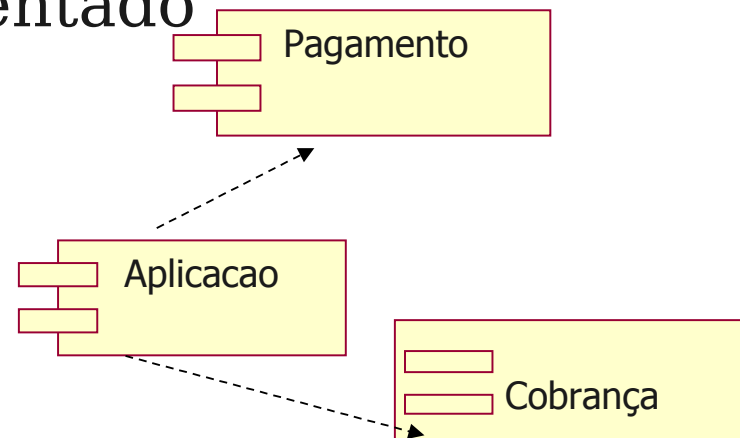
- Diagramas de classes de domínio representando as entidades do caso de uso
- Diagrama de classes de projeto e realização de caso de uso, já considerando a arquitetura definida
- Diagrama de classes representando arquitetura de referência, principalmente se utilizado padrões (*patterns*)



Visão 4 + 1

- Visão de Implementação

- Normalmente representado por diagrama de componentes, representando as dependências de compilação



- Visão de subsistemas

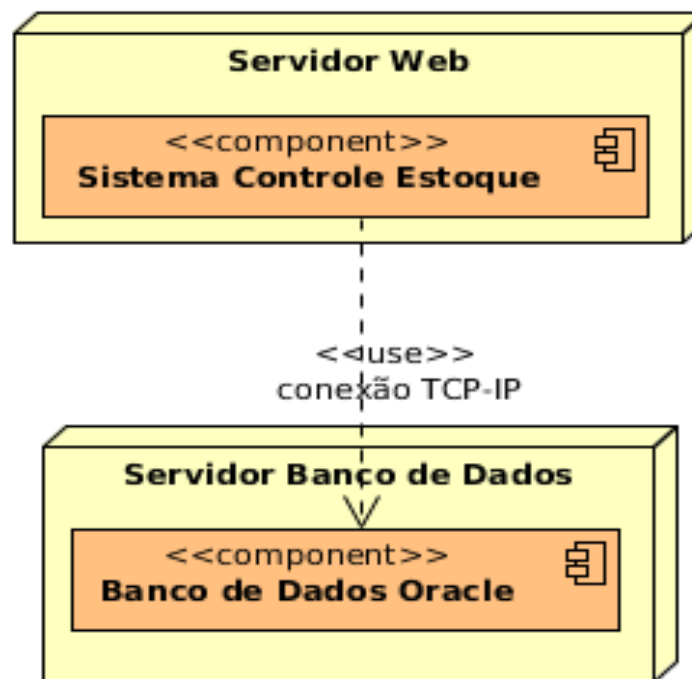
- Pode conter descrição detalhada dos subsistemas

- Pode conter exemplo de código detalhando cenários específicos



Visão 4+1

- Visão de Implantação
 - Estrutura física de implantação





Visão 4 + 1

- Visão de Processo
 - Descreve os fluxos e processos (comunicação, processos concorrentes, etc.)
 - Pode-se utilizar diagrama de atividades ou diagrama de estados para representar os fluxos e processos
 - Descrição detalhada das necessidades de desempenho da aplicação (número de usuários, acessos simultâneos, vazão, etc..)



Visão 4 + 1

- Nem todos os sistemas necessitam de todas as visões
 - Por exemplo, sistema com um único processo, pode-se ignorar a Visão de Processo
- Alguns sistemas podem precisar de visões adicionais
 - Visão de Dados
 - Visão de Segurança
 - ...



Dicas Arquitetura

- Uso extensivo de padrões arquiteturais e de projeto (*patterns*)
 - Muitos arquitetos experientes já tiveram problemas semelhantes ao seu.
- Decomposição do sistema baseada em pacotes e componentes
- Simplicidade



Atividade 01

- Atividade referente ao Estudo de Caso (Sistema de Ponto de Vendas)
 - Tarefa 01 → Identificar atributos de qualidade e requisitos não-funcionais importantes. Adicionalmente, identificar possíveis pontos de variação e pontos de evolução.
 - Tarefa 01 → Identificar e documentar (Visão de Caso de Uso) um ou dois casos de uso que julgue importante para definição da arquitetura