

Projeto de Software

Nivelamento em Engenharia de Requisitos e Algoritmos e Estrutura de Dados

Lesandro Ponciano

2024

Objetivos de Aprendizagem

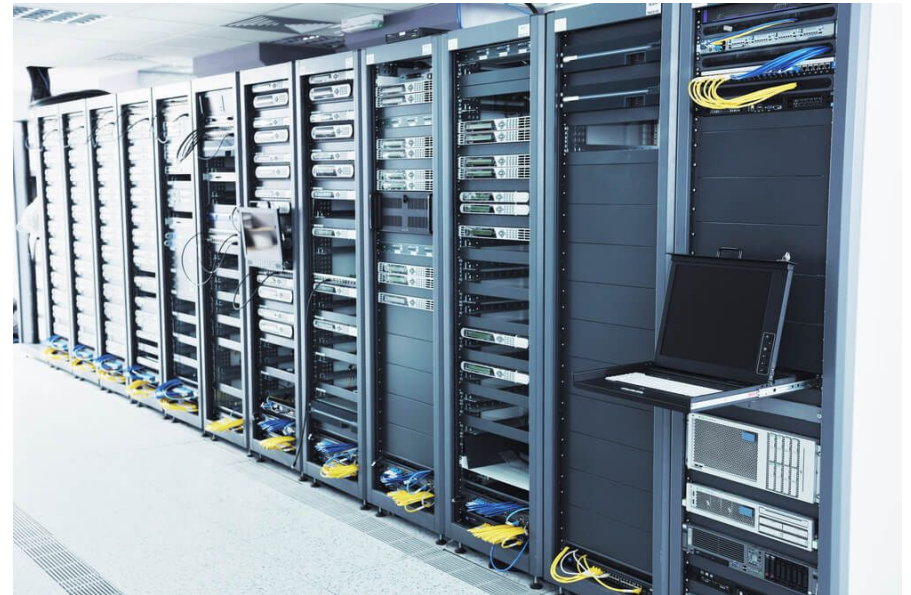
- **Revisar** Programação Orientada a Objetos e Engenharia de Requisitos
- **Contextualizar** a Engenharia de Software
- **Contextualizar** o Projeto de Software na Engenharia de Software

Sistemas de Informação

- Sistema
 - Conjunto de elementos inter-relacionados que estão articulados e formam um todo organizado
- Sistema de Informação (automatizados)
 - Conjunto de componentes inter-relacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informações
 - hardware, software, redes e instalações de comunicação, banco de dados e pessoas

Hardware

- Armazenamento
- Processamento
- Transmissão de dados
- Autonomia
- Mobilidade



Pessoas

- Interessados (*stakeholders*), exemplos:
 - Cliente, Usuários, Acionistas, Mantenedores
- Fornecedores do software
 - Entendem o que o cliente quer e o que os usuários precisam
 - Especificam, projetam, implementam e validam o software
 - Mantêm e evoluem o software ao longo do tempo

Softwares

... são elementos do sistema lógico, não do sistema físico

... são instruções (programas de computador) que quando executadas fornecem características, funções e desempenho desejados

... incluem estruturas de dados que possibilitam aos programas manipular informações adequadamente

... incluem informações descritivas sobre a operação e uso dos programas

Algoritmos

... são “descrição de um conjunto de comandos que, obedecidos, resultam numa sucessão finita de ações” (Farrer et al. 1999)

... são “Sequência de passos computacionais que transformam uma entrada em uma saída” (Cormen et al. 2002)

... são "um processo sistemático para a resolução de um problema". (Szwarcfiter et al. 2010)

Solução de um problema computacional

Estrutura de Dados

- É a forma de armazenamento e organização de dados que é usada para resolver o problema
 - A estrutura pode afetar a eficiência da solução
- Existem estruturas de dados estáticas e dinâmicas
 - Estruturas de dados **estáticas** são compostas por um número fixo (finito) de elementos
 - Estruturas de dados **dinâmicas** podem ter um número indefinido de elementos

Tipos de Software

- Software de sistema
- Software de aplicação
- Software de engenharia ou científico
- Software embutido (ou embarcado)
- Software de linhas de produto
- Aplicações para Web
- Software de inteligência artificial

Softwares Grandes

- Máquinas de busca, como Google
 - Algoritmos como *pagerank* e *map-reduce*
- Redes sociais, como Facebook
 - Algoritmos de processamento de grafo
- Sistemas Peer-to-Peer (P2P), como BitTorrent
 - Algoritmos baseados em reputação como o “olho-por-olho e dentes-por-dentes” (*tit-for-tat*)

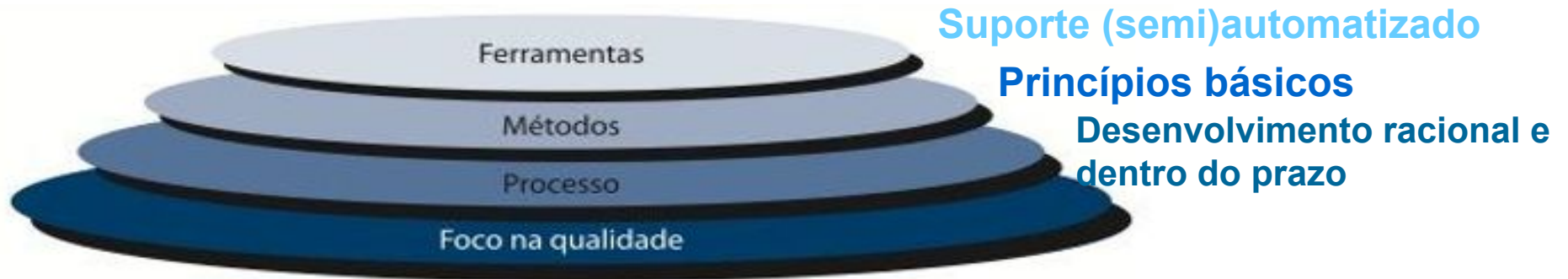
Diversos componentes, alta complexidade

Engenharia de Software

- “é 1) aplicação de uma abordagem **sistemática**, **disciplinada**, e **quantificável** no desenvolvimento, na operação e na manutenção de software, 2) estudo e abordagem do descrito em 1” (IEEE*, 1993)

*IEEE é o Instituto de Engenheiros Eletricistas e Eletrônicos (*Institute of Electrical and Electronics Engineers*)

Qualidade na Engenharia de Software



- A pedra fundamental que sustenta a engenharia de software é o foco na qualidade
- Qualquer abordagem de ES deve estar fundada em um comprometimento com a qualidade

Qualidade do Software [Pressman]

- Para Pressman (2011)
 - aderência aos requisitos funcionais e de desempenho que foram explicitamente declarados
 - aderência a padrões de desenvolvimento claramente documentados
 - características implícitas que são esperadas de todo software desenvolvido por profissionais

Qualidade do Software [ISO/IEC]

- Pela norma ISO/IEC 25010:2011, qualidade está associada às características
 - Adequabilidade funcional
 - Eficiência do desempenho
 - Compatibilidade
 - Usabilidade
 - Confiabilidade
 - Segurança
 - Manutenibilidade
 - Portabilidade

Processo de Desenvolvimento

- Desenvolver um software significa
 - Transformar a **necessidade** de um usuário ou de um mercado em um produto de software
 - **Projetar** e **implementar** um sistema computacional
- Processo de Desenvolvimento de Software, ou simplesmente
 - Processo de Desenvolvimento
 - Processo de Software

Requisitos do Software

- Requisitos são as descrições de
 - O que o software deve fazer
 - Os serviços que deve oferecer
 - As restrições ao seu funcionamento
- Ou, ainda..
 - Condição ou capacidade **necessária a um usuário** para resolver um problema ou alcançar um objetivo
 - Condição ou capacidade que deve ser alcançada ou **possuída por um sistema** para satisfazer um contrato
 - Uma representação documentada de uma condição ou capacidade como nos itens 1 ou 2

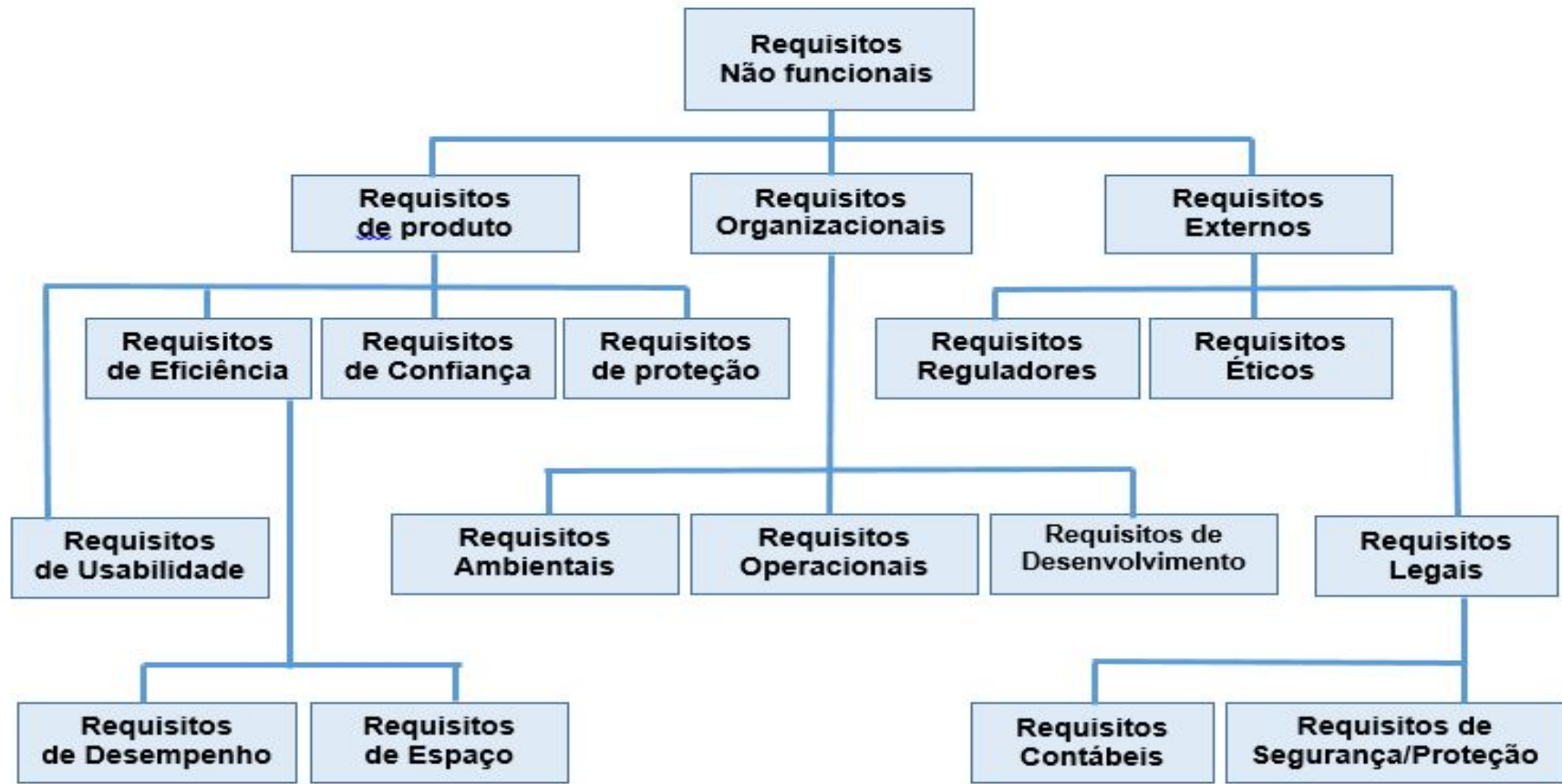
Requisitos Funcionais

- Requisitos funcionais (RF) são declarações de
 - Serviços que o sistema deve oferecer
 - Como o sistema deve reagir a entradas específicas
 - Como o sistema deve se comportar em determinadas situações
- Estão diretamente relacionados aos **objetivos** do sistema

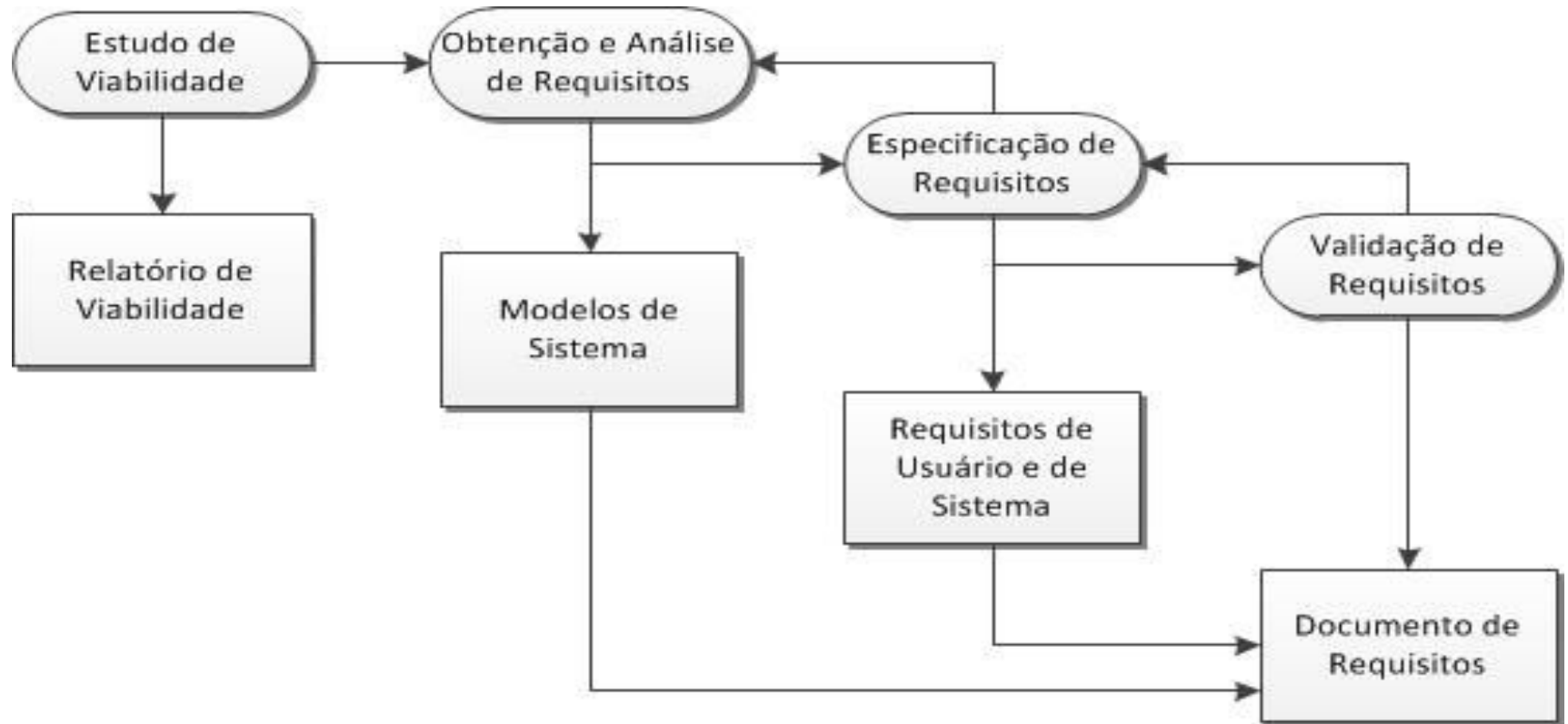
Requisitos Não-Funcionais

- Requisitos não-funcionais (RNF), são **restrições** aos serviços ou funções oferecidos pelo sistema
 - Restrições de tempo (*timing*)
 - Restrições de processo de desenvolvimento
 - Restrições impostas por plataforma
 - Restrições legais
- Surgem de restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade e fatores externos

Tipos de Requisitos Não-Funcionais



Engenharia de Requisitos



No fim, tem-se um "acordo" sobre o que deve ser implementado

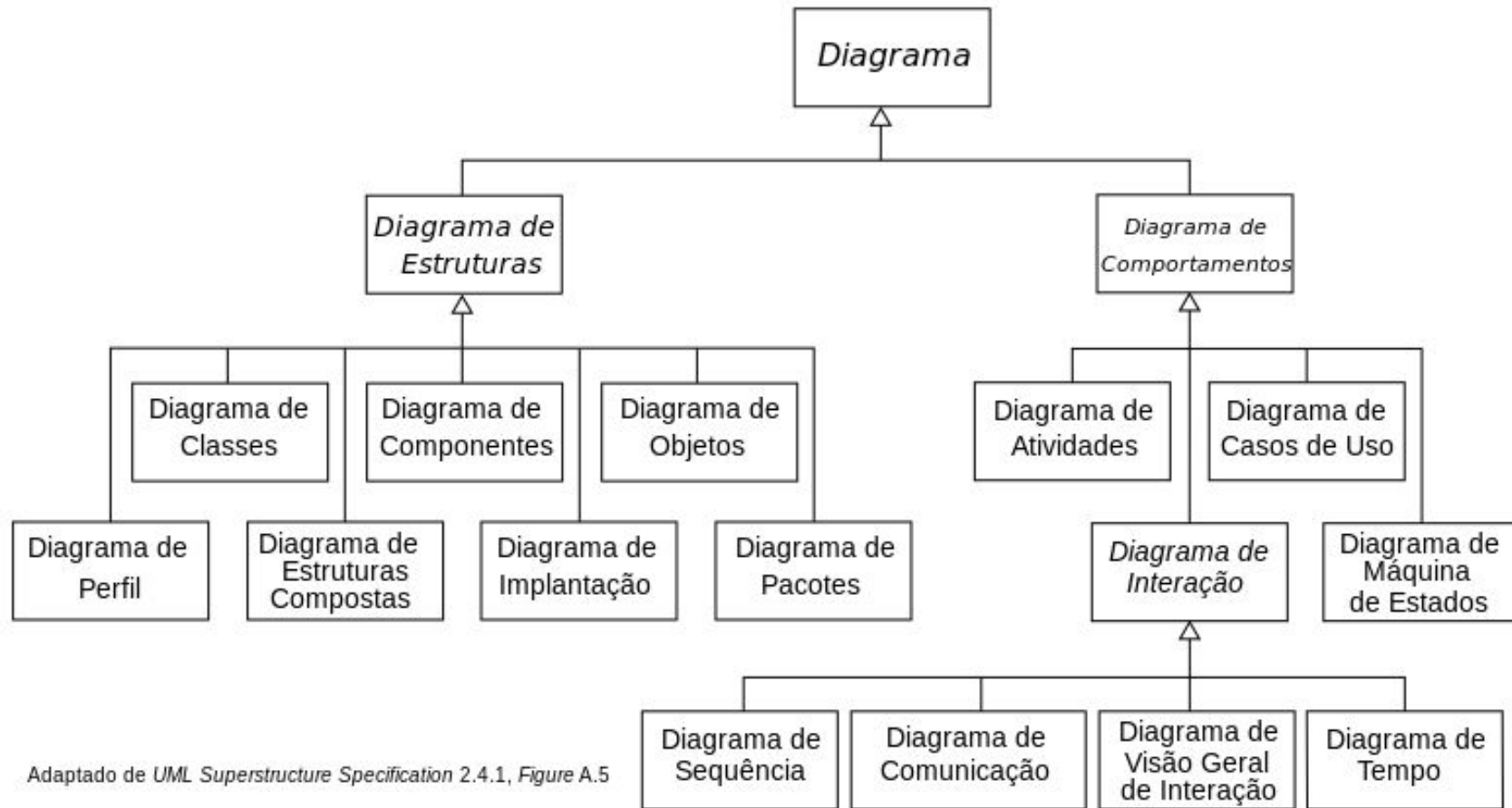
Modelagem com UML

- A UML (*Unified Modeling Language*)
 - Linguagem de modelagem visual
 - Conjunto de notações e semântica correspondente para representar visualmente perspectivas de um sistema
 - Permite que projetistas visualizem os produtos de seus trabalhos em diagramas padronizados
- A UML é independente de linguagens de programação e processos de desenvolvimento
 - não diz para você o que fazer primeiro e em seguida
 - não diz como projetar seu sistema, mas auxilia a visualizar seu desenho e a comunicação entre os objetos

Classes de Diagramas pela UML 2.0

- Diagramas de **comportamento** (ou comportamentais)
 - Permitem visualizar, especificar, construir e documentar aspectos dinâmicos de um sistema
 - Representação das suas partes que passam por alteração ao longo do tempo de execução
- Diagramas de **estruturas** (ou estruturais)
 - Referem-se ao aspecto estrutural em termos do sistema como um todo e das suas classes
 - Permitem visualizar, especificar, construir e documentar os aspectos estáticos, que é o seu esqueleto e sua estrutura relativamente estáveis ao longo do tempo de execução

Diagramas da UML 2.0



Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

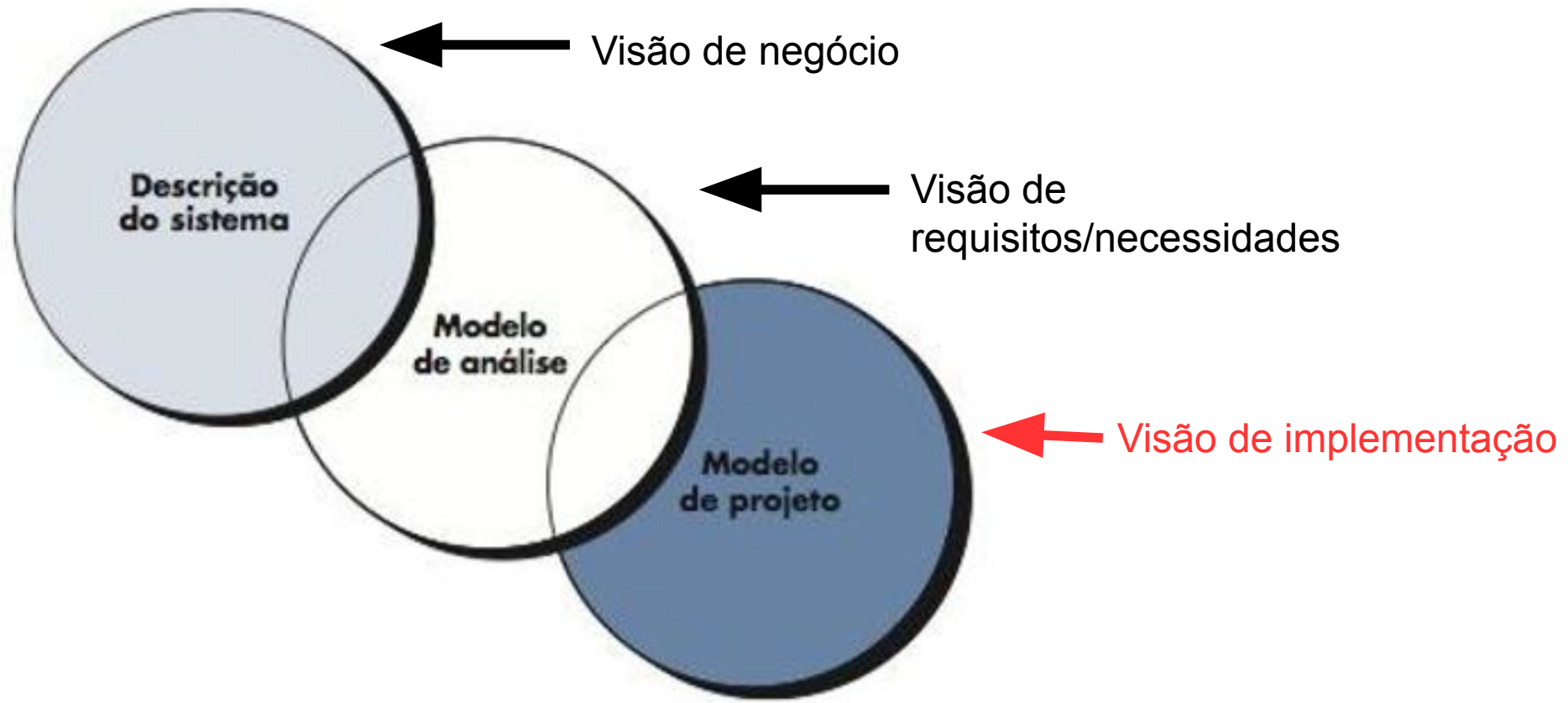
Project vs Design

- A palavra "projeto" é ambígua, pode significar duas coisas bem diferentes
- Projeto (do inglês *project*)
 - Algo a ser feito, algo a ser gerenciado
 - Ex.: "Gerência de Projetos"
- Projeto (do inglês *design*)
 - Significa desenho, esboço, modelo
 - Ex.: "Projeto de Software"

Requisitos e Projeto

- O sistema é projetado observando-se a lista de requisitos, exemplo de requisitos:
 - "O sistema deve fornecer ao cliente a compra menos custosa que satisfaça seus parâmetros de definição de produto"
 - "O sistema deve estar integrado ao sistema bancário XYZ utilizado pela organização"
 - "Nenhuma ação do usuário pode demorar mais de 1 minuto para ser processada pelo sistema"
- Os requisitos definem "o que" o sistema deve fazer e o projeto trata de "como" ele deve fazer

Negócio, Requisitos e Projeto



A Atividade de Projeto

- “Processo pelo qual os requisitos do software são traduzidos para uma representação do software que permite sua implementação/codificação”
 - Preliminar: preocupa-se com a transformação dos requisitos do software em uma arquitetura de software e de dados
 - Detalhado: concentra-se nos aprimoramentos da representação arquitetural que levam à estrutura de dados detalhada e às representações algorítmicas do software

A Atividade de Implementação

- O software deve ser produzido para atender às especificações
- Na Engenharia de Software, o projeto é uma extensão da especificação
- O projeto visa a implementação em um computador
 - É algo a ser compreendido pelo programador
- A implementação é a escrita do código
 - Tarefa relativamente **simples se o projeto for bem-feito**

Exercício de Fixação

Para que serve o projeto (design) do software?

- a) Para enunciar os requisitos funcionais e não-funcionais do software.
- b) Para modelar como o software deve ser estruturado e seu comportamento.
- c) Para testar se o software tem a qualidade desejada pelo cliente e usuários.
- d) Para prover uma visão de negócio acerca do software a ser implementado.

Referências

- SOMMERVILLE, Ian. Engenharia de Software - 9a edição. Pearson ISBN 9788579361081. (Capítulos de 1 a 4)
- PRESSMAN, Roger. Engenharia de software. 8. Porto Alegre ISBN 9788580555349. (Capítulos de 1 a 4)

Projeto de software é uma visão de implementação que é intermediária entre a visão de requisitos/necessidades e a codificação.

Projeto de Software

Prof. Dr. Lesandro Ponciano

<https://orcid.org/0000-0002-5724-0094>