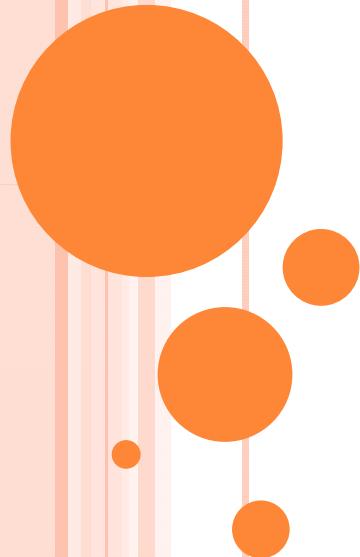


# **PROF. BRUNO DE CASTRO H. SILVA**



**APS  
INTRODUÇÃO**

# SUMÁRIO

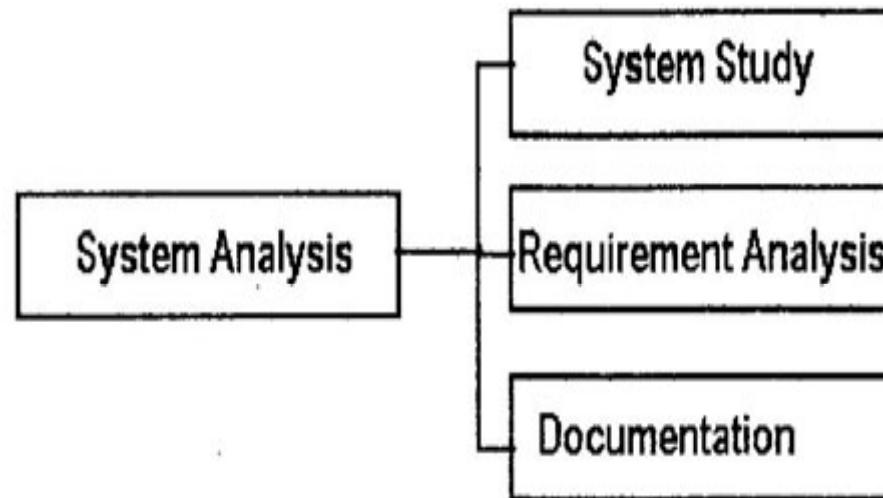
- Análise de Sistemas;
- Projeto de Sistemas;
- UML;
- Diagramas de UML;
- Leitura Complementar;

# ANÁLISE DE SISTEMAS

3

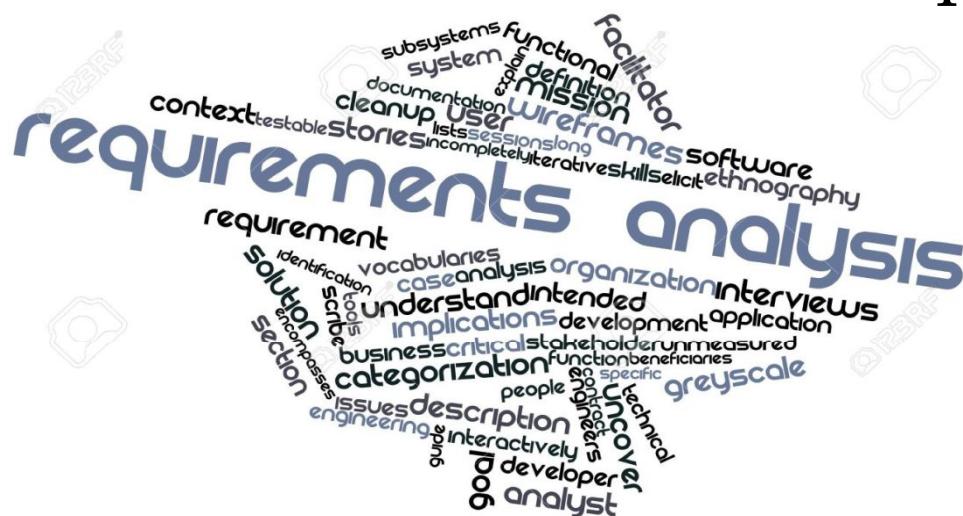
# ANÁLISE DE SISTEMAS

- A Análise de Sistemas é a solução conceitual dada ao problema.
- Marca o início da definição computacional, mas sem levar em conta detalhes da implementação.



# ANÁLISE DE SISTEMAS

- Na Análise de Sistema, deve-se focar em:
    - Determinar os serviços que o usuário espera do sistema;
    - E as condições (restrições) sob as quais o sistema será desenvolvido e operar.



## ANÁLISE DE SISTEMAS

- O analista deve ser capaz de retirar os requisitos:

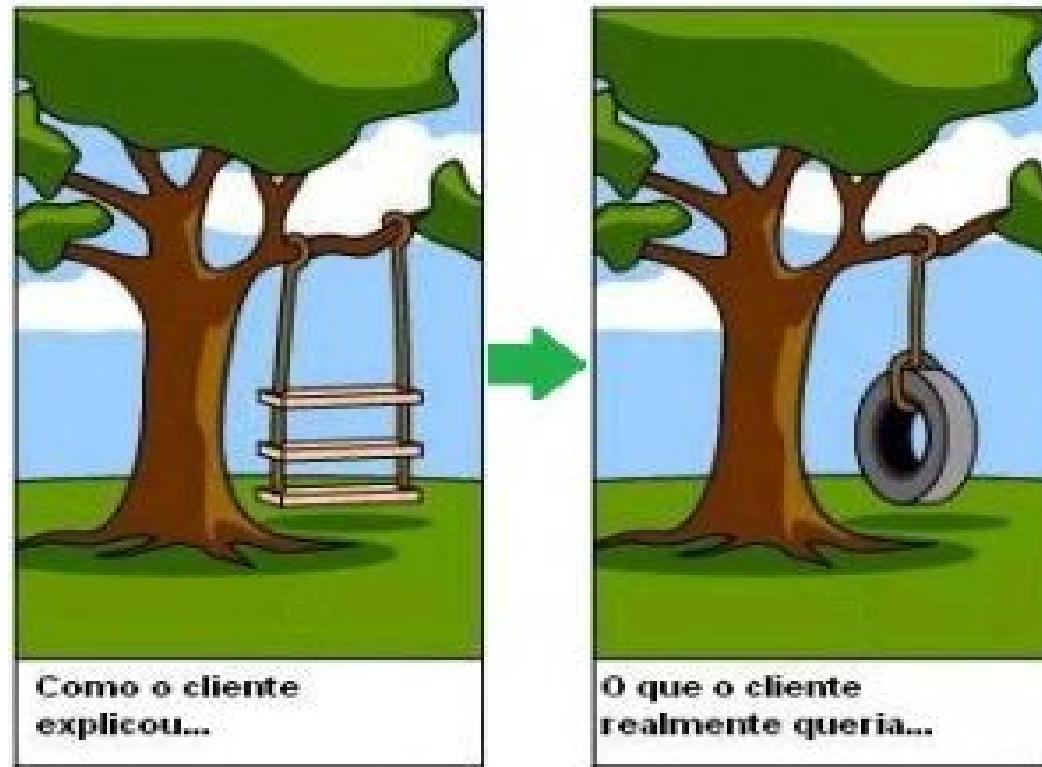
- Funcionais;
- Não Funcionais;
- De Domínio;



- Segundo as necessidades do usuário.

## ANÁLISE DE SISTEMAS

- Os *Requisitos Funcionais* descrevem as funcionalidades, como o sistema deve reagir a entradas de dados, bem como o seu comportamento esperado.



# ANÁLISE DE SISTEMAS

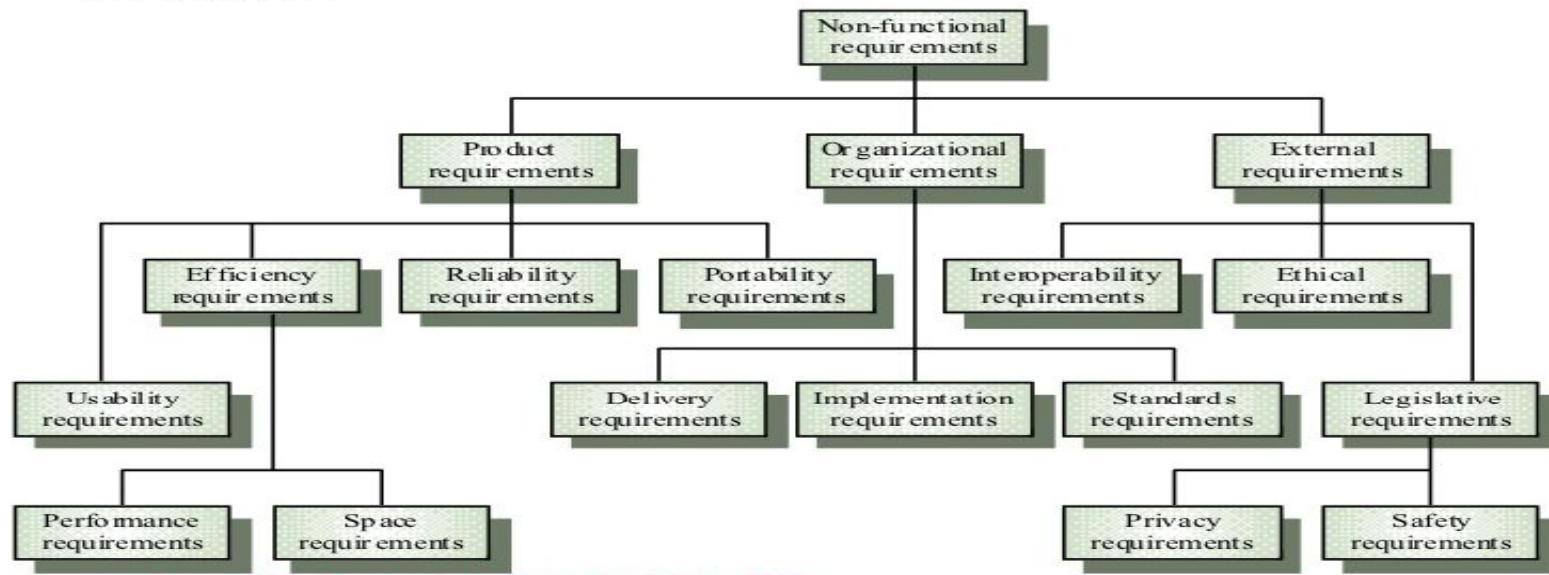
Exemplos de *Requisitos Funcionais*:

- O software deve gerar relatório de todos os clientes cadastrados;
- O módulo de contas a receber deve listar todos os clientes com contas vencidas a mais de 30 dias;
- O cliente deve ser identificado no sistema pelo seu número de CPF.

# ANÁLISE DE SISTEMAS

- Os *Requisitos Não Funcionais* expressam condições e restrições que devem ser atendidas pelo software.

Non-Functional Requirements (NFR) Structural Breakdown



## ANÁLISE DE SISTEMAS

Como exemplos de *Requisitos Não Funcionais*, pode-se citar:

- O software deve ser compatível com os navegadores web Internet Explorer e Firefox;
- O software deve exibir o resultado das consultas em até 3 segundos em operações que envolvam banco de dados;
- Os dados dos clientes devem ser mantidos criptografados dentro do banco de dados.

## ANÁLISE DE SISTEMAS

- Os *Requisitos de Domínio* são herdados do domínio da aplicação (financeiro, educacional, industrial, hospitalar etc.) e refletem diretamente as suas características.
- São considerados superiores aos Requisitos Funcionais e geralmente descrevem características gerais a toda aplicação.

## ANÁLISE DE SISTEMAS

Como exemplos de *Requisitos de Domínio*, pode-se citar:

- O desconto máximo em operações de venda no sistema será de 10%;
- O cálculo da média final nas disciplinas presenciais dar-se-á pela média aritmética entre todas as notas.

## ANÁLISE DE SISTEMAS

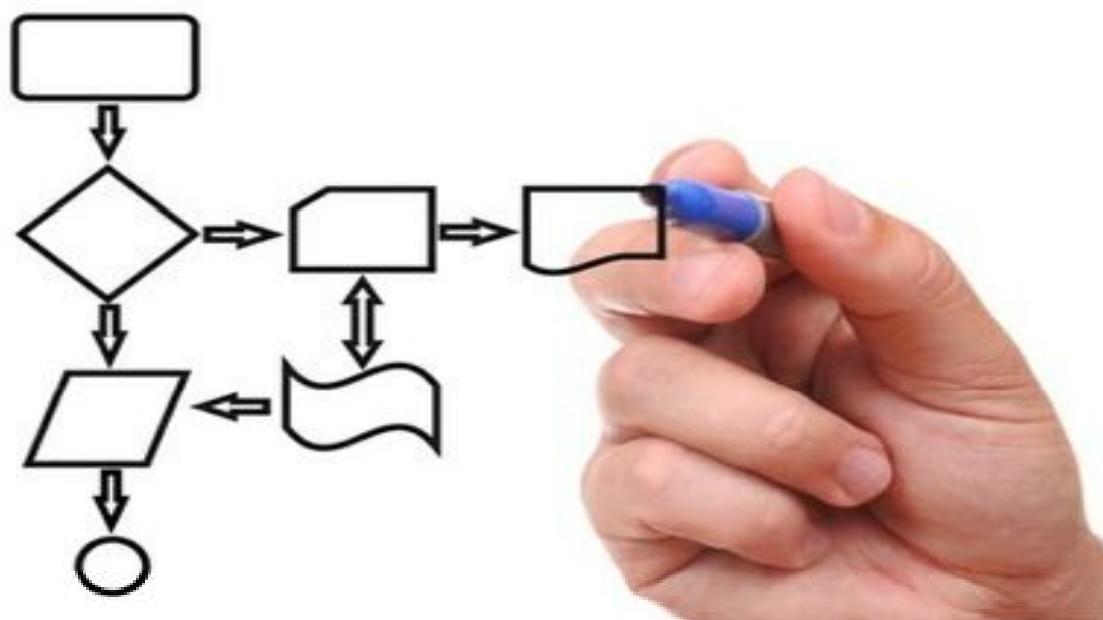
○ Eis algumas formas de levantar requisitos:

- Entrevistas
- Questionários
- Observação / Visitas instalações  
(próprias ou outras)
- Demonstrações
- Pesquisa extena
- Análise da Documentação
- Joint Application Design (JAD) - IBM
- Brain Storm
- Brain Writing

# PROJETO DE SISTEMAS

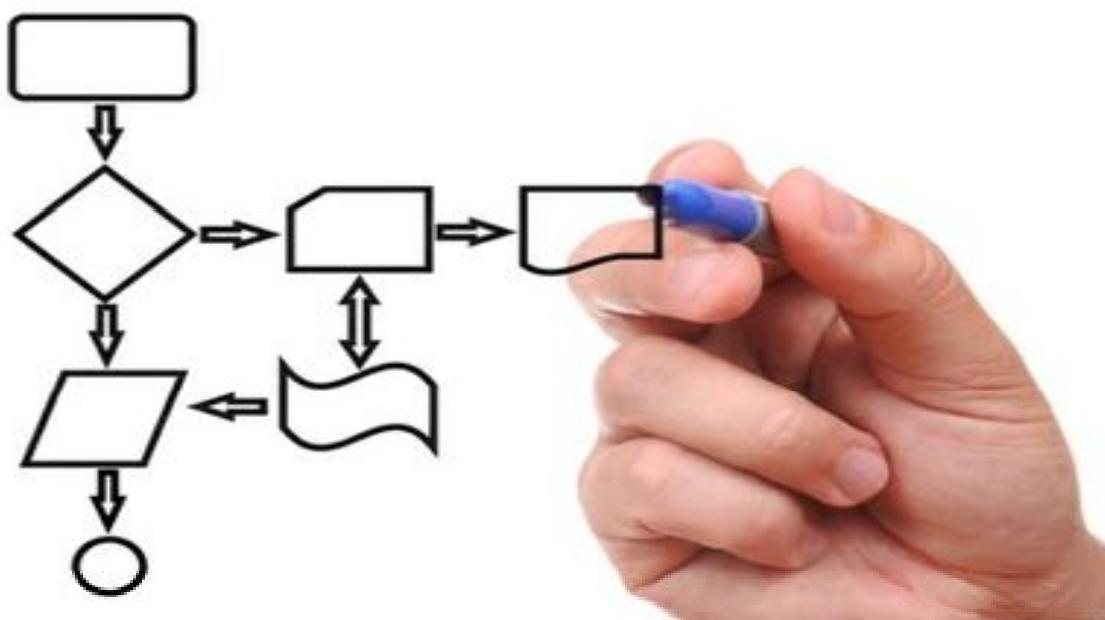
## PROJETO DE SISTEMAS

- Na etapa de Projeto e Arquitetura de Software define-se “COMO” o software deve ser construído, sem, entretanto, chegarmos a codificá-lo de fato.



## PROJETO DE SISTEMAS

- Também conhecido como a fase de Design (Desenho), esta fase influênciaria o sucesso da fase seguinte – codificação (programação).



# UML

## UML

- UML significa *Unified Modeling Language* ou Linguagem de Modelagem Unificada de projetos orientados a objetos.
- Trata-se de uma linguagem (e não um método) padrão de notação de projetos.



# UML

- A origem da UML data nos anos 1990;
- De lá até 1996, as notações foram refinadas e as melhores unificadas na UML que por sua vez representa a unificação das notações de *Booch*, *Rumbaugh* e *Jacobson*.



## UML

- Em março de 2003, a OMG (Object Management Group) lançou a versão 1.5
- Em outubro de 2004, a OMG lançou versão 2.0, que até então, continua sendo adotada .



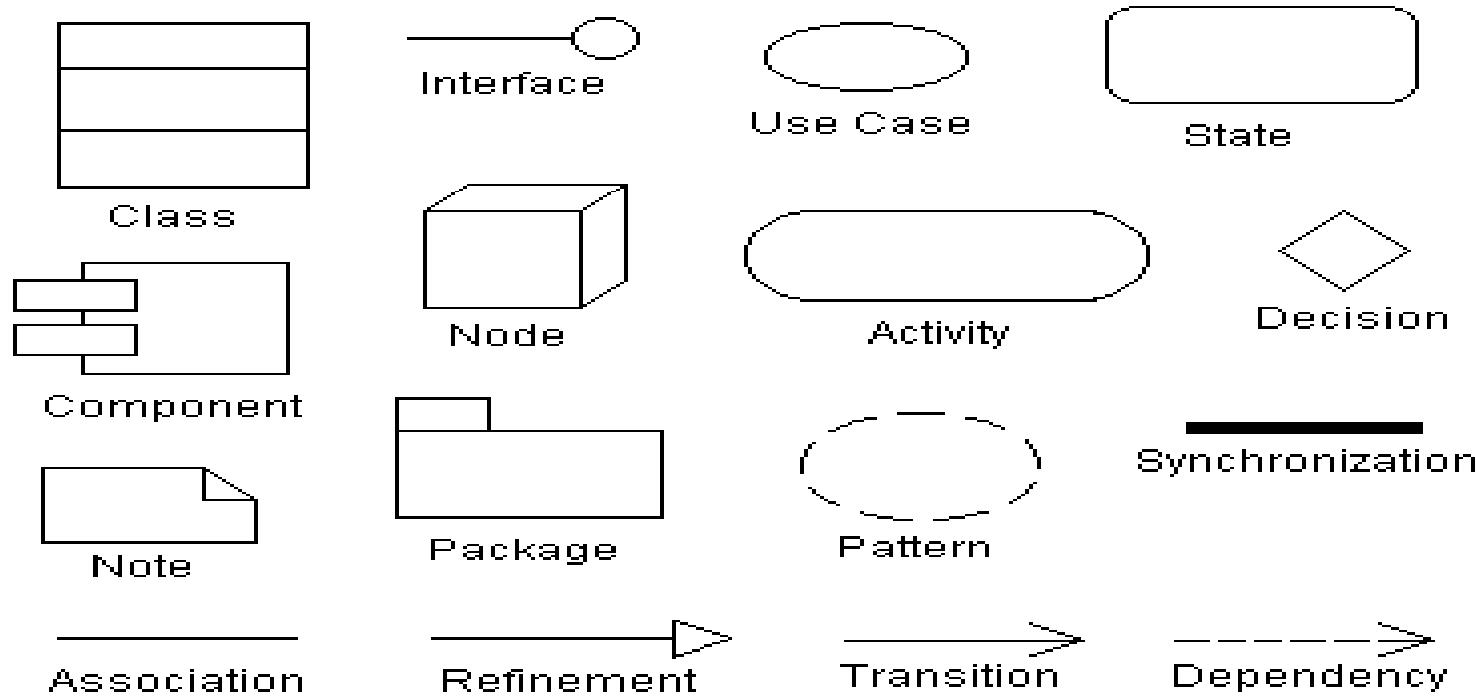
**SHOW WHAT  
YOU KNOW.**



# DIAGRAMAS UML

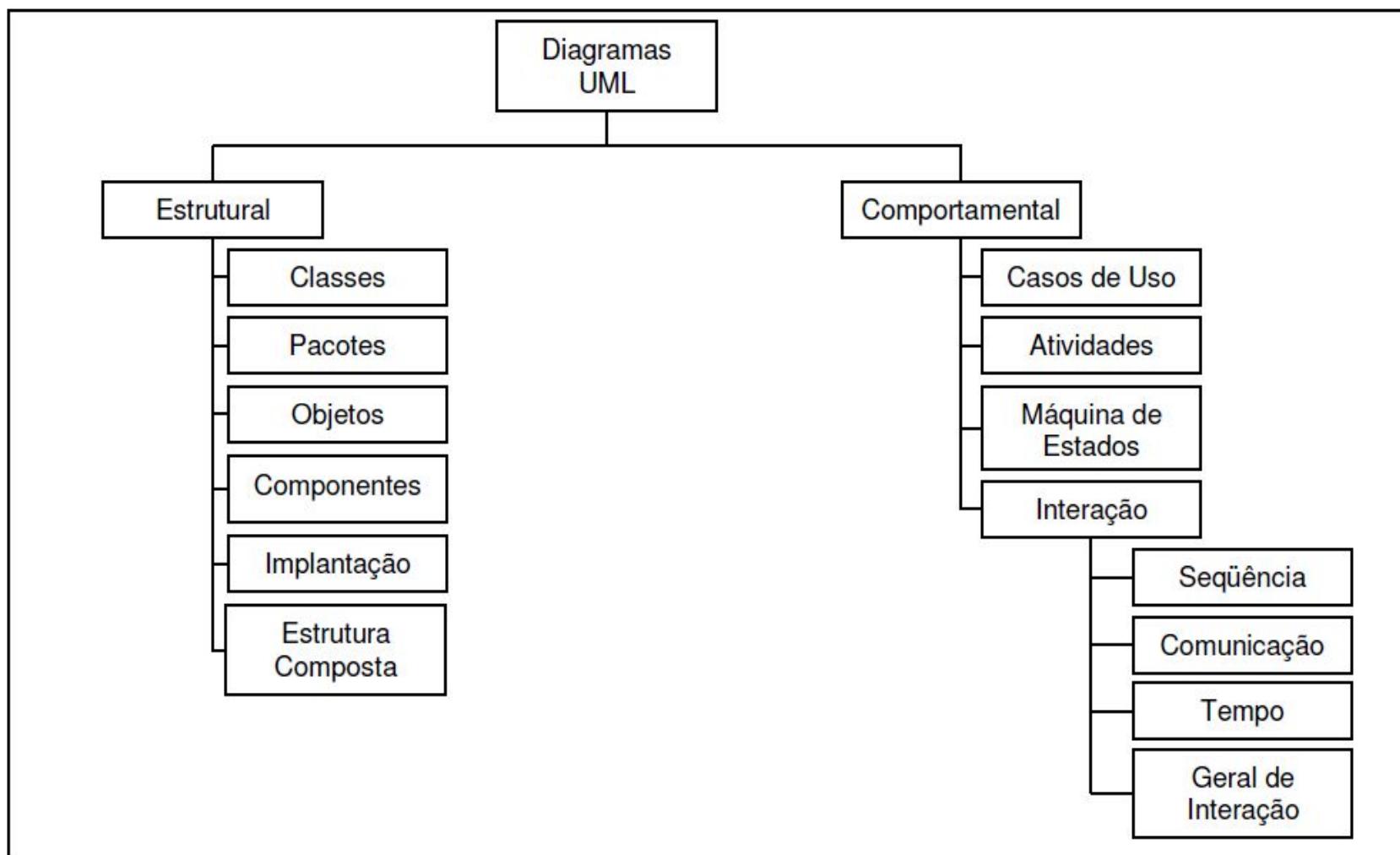
# DIAGRAMAS UML

○ Na UML, há diagramas os quais são mais indicados para modelar a estrutura do sistema e outros, o comportamento.



# DIAGRAMAS UML

- Baseado nesta premissa, foi feita a seguinte classificação:



# LEITURA COMPLEMENTAR

## LEITURA COMPLEMENTAR

Livro: UML: guia do usuário. 2 – Grady, B;  
Rumbaugh, J;  
Jackson, I. Módulo 1.