

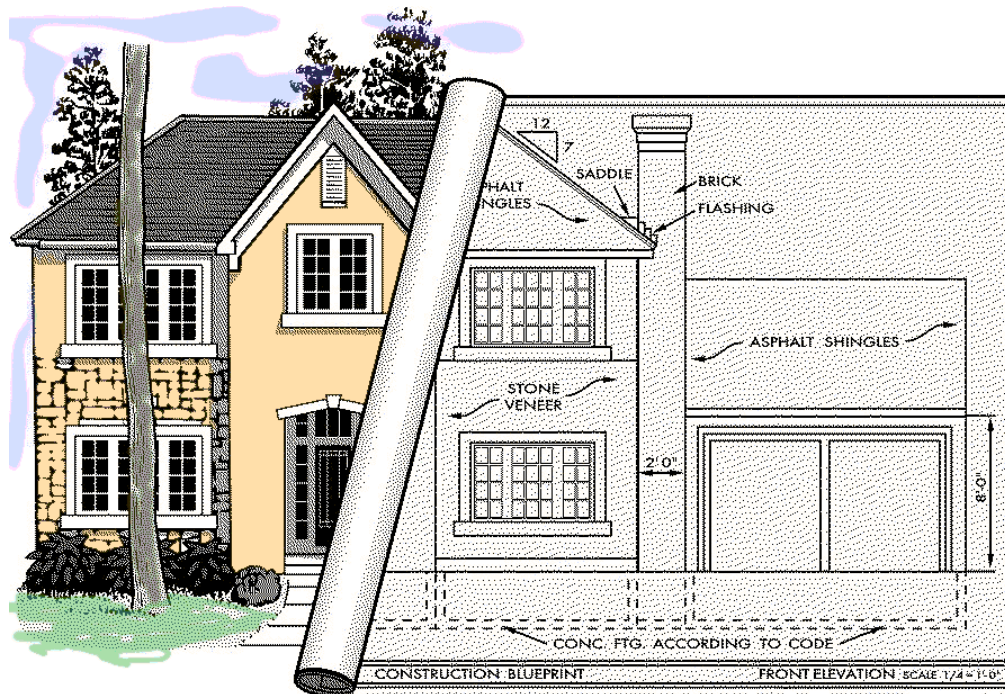
# ENGENHARIA DE SOFTWARE

## UML



- Cristina Becker Matos

## Complexidade

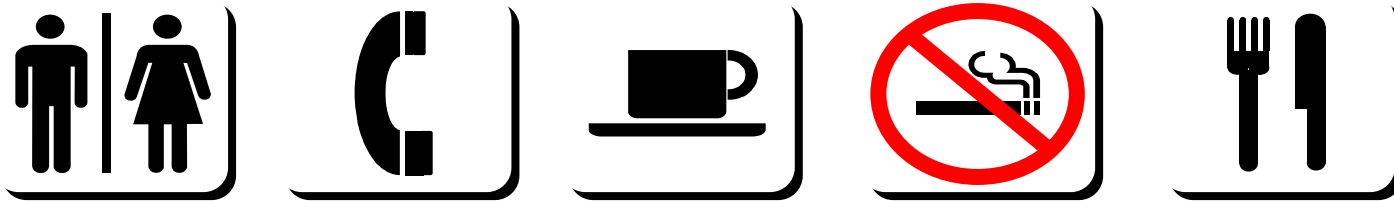
- ▶ Na construção de sistemas de software, assim como na construção de sistemas habitacionais, também há uma gradação de complexidade.
- ▶ A construção desses sistemas necessita de um planejamento inicial.



# Razões para construção de modelos

- 
- Gerenciamento da complexidade inerente ao desenvolvimento de software.
  - Comunicação entre as pessoas envolvidas.
  - Redução dos custos no desenvolvimento.
  - Predição do comportamento futuro do sistema.
- 

- Diagramas fornecem uma representação concisa do sistema. “**uma figura vale por mil palavras**”.



- No entanto, modelos também são compostos de informações textuais.
- Dado um modelo de uma das perspectivas de um sistema, diz-se que o seu diagrama, juntamente com a informação textual associada, formam a **documentação** deste modelo.


# UML



## • UML é...


- ★ uma linguagem visual.
- ★ independente de linguagem de programação.
- ★ Independente de processo de desenvolvimento.

## • UML não é...

- ★ uma linguagem programação.
  - ★ uma técnica de modelagem.
- 

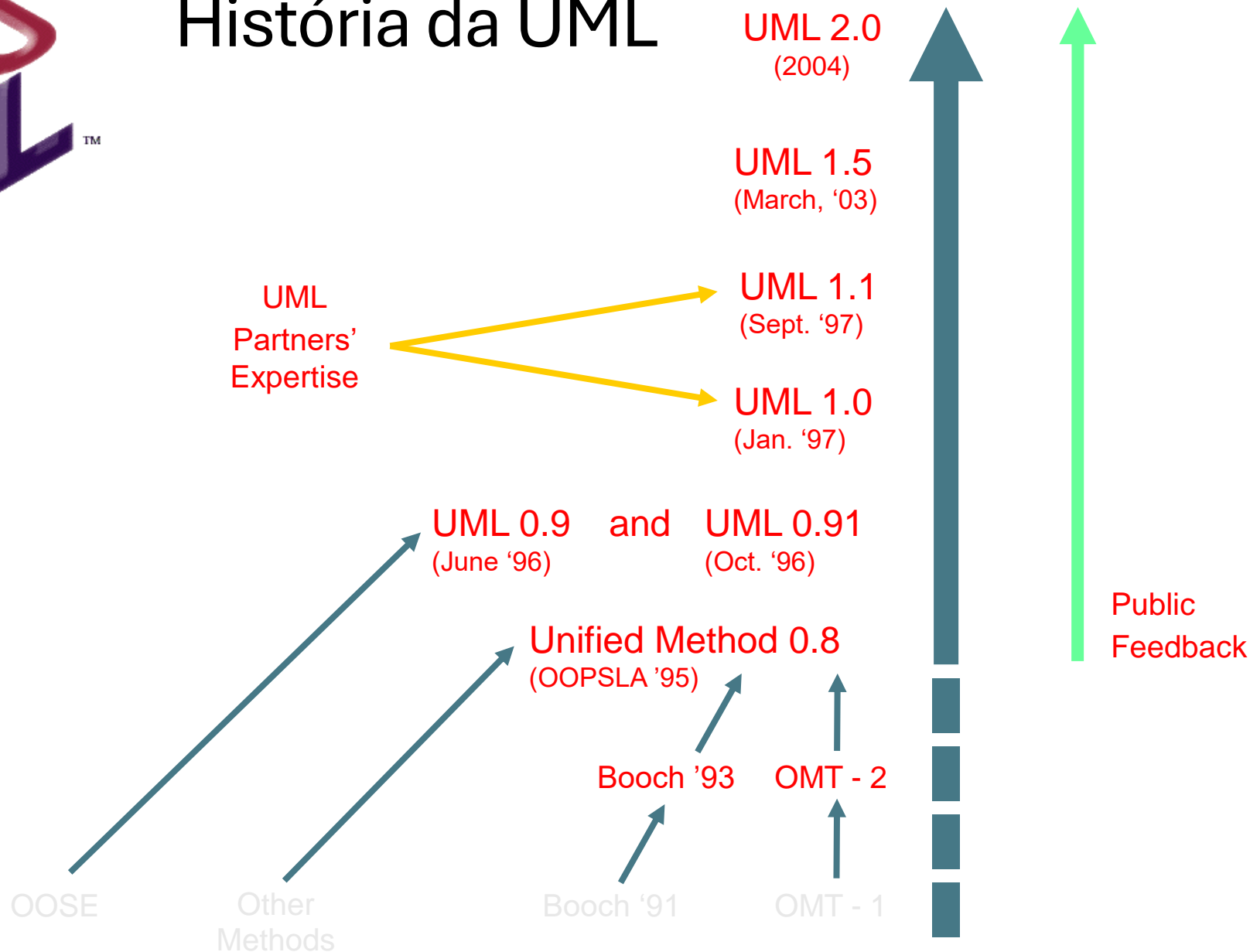
## UML



- Surge a UML (Unified Modeling Language) em 1996 como a melhor candidata para ser linguagem “unificadora” de notações.
  - Em 1997, a UML é aprovada como padrão pelo OMG(Object Management Group) – grupo de empresas que define e ratifica padrões na área da orientação à objetos.
  - Desde então, a UML tem tido grande aceitação pela comunidade de desenvolvedores de sistemas.
  - Em 2004 resultou em uma nova versão --UML 2
  - UML 2.4.1 foi formalmente lançado em agosto de 2011
  - UML 2.X.....13 diagramas.
  - Webiste: <http://www.uml.org/>
- 



# História da UML



# A história da Orientação à Objetos

- Maiores marcos da Orientação à Objetos

Simula



1967

C ++



Late 1980s

The UML



1997

1972



Smalltalk

1991



Java

2004



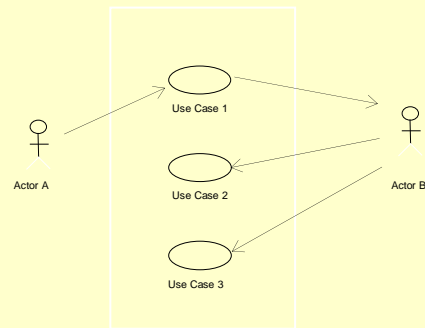
UML 2



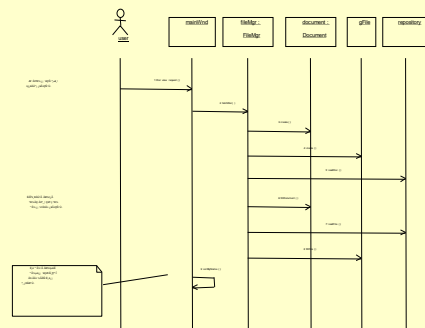
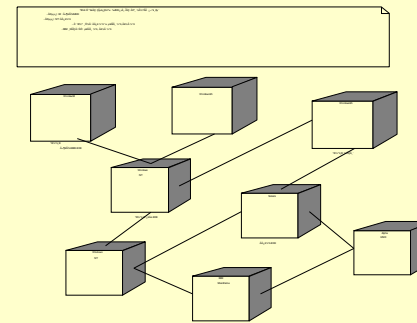
# A UML é uma linguagem para Documentação

- A UML aborda a documentação da arquitetura do sistema, requisitos, planejamento e gerenciamento de versão

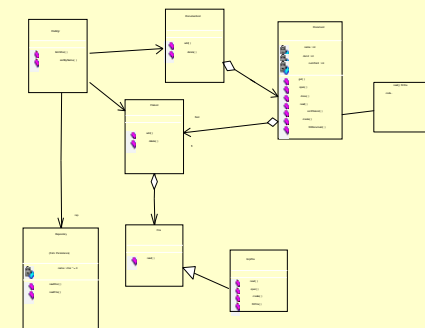
Use Case Diagram



Deployment Diagram





Sequence Diagram



Class Diagram

## Diagramas da UML

- 
- Um processo de desenvolvimento que utilize a UML como linguagem de modelagem envolve a criação de diversos documentos.
    - ★ Estes documentos podem ser textuais ou gráficos.
    - ★ Estes documentos são denominados artefatos de software.
    - ★ São os artefatos que compõem as visões do sistema.
  - Os artefatos gráficos produzidos durante o desenvolvimento de um sistema de software são definidos através da utilização dos diagramas da UML.
- 

## Diagramas Estruturais

- Diagrama de classes
- Diagrama de objetos
- Diagrama de componentes
- Diagrama de instalação
- Diagrama de pacotes
- Diagrama de estrutura

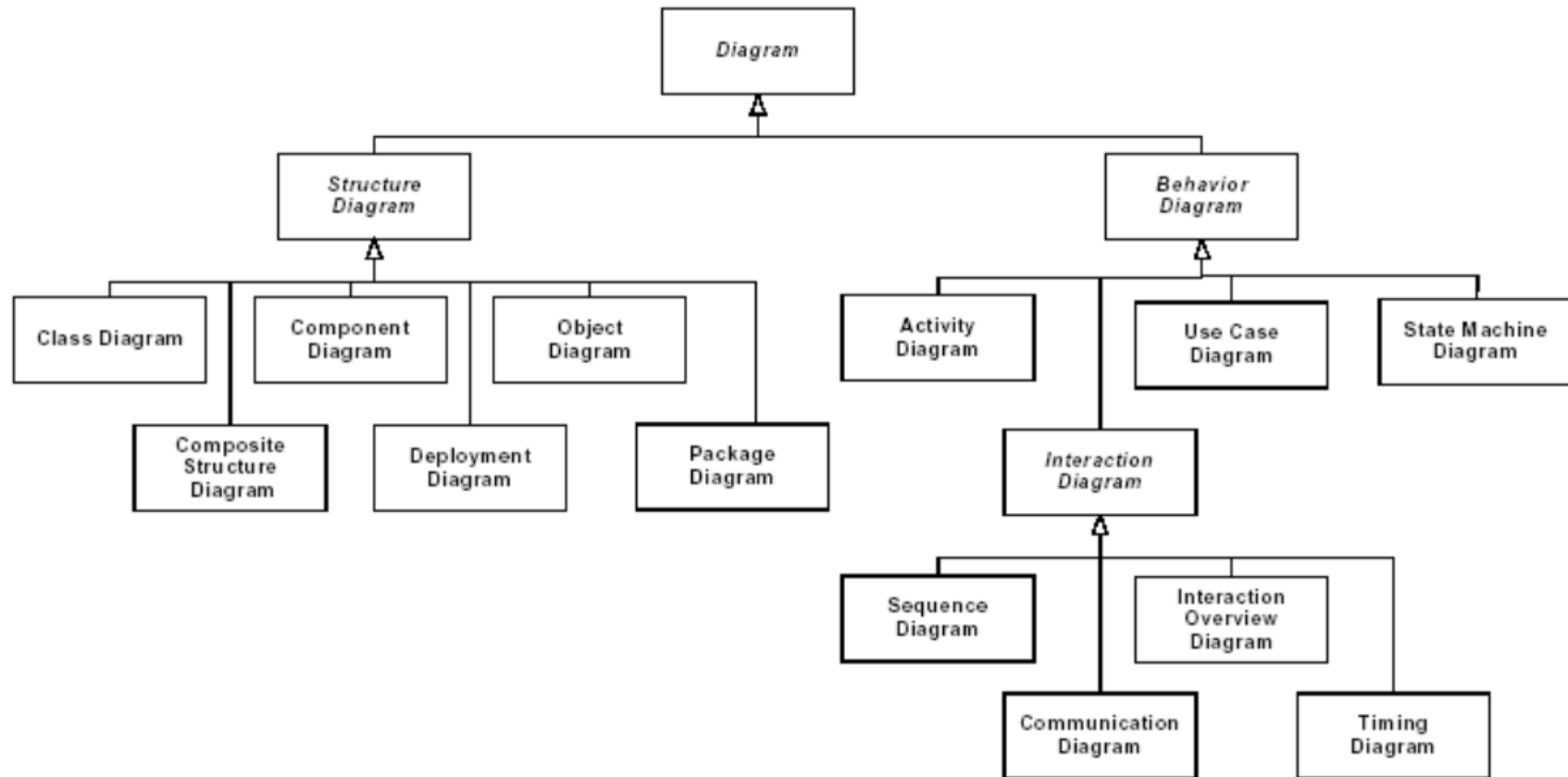
## Diagramas Comportamentais

- Diagrama de Caso de Uso
- Diagrama de transição de estados
- Diagrama de atividade

## Diagramas de Interação

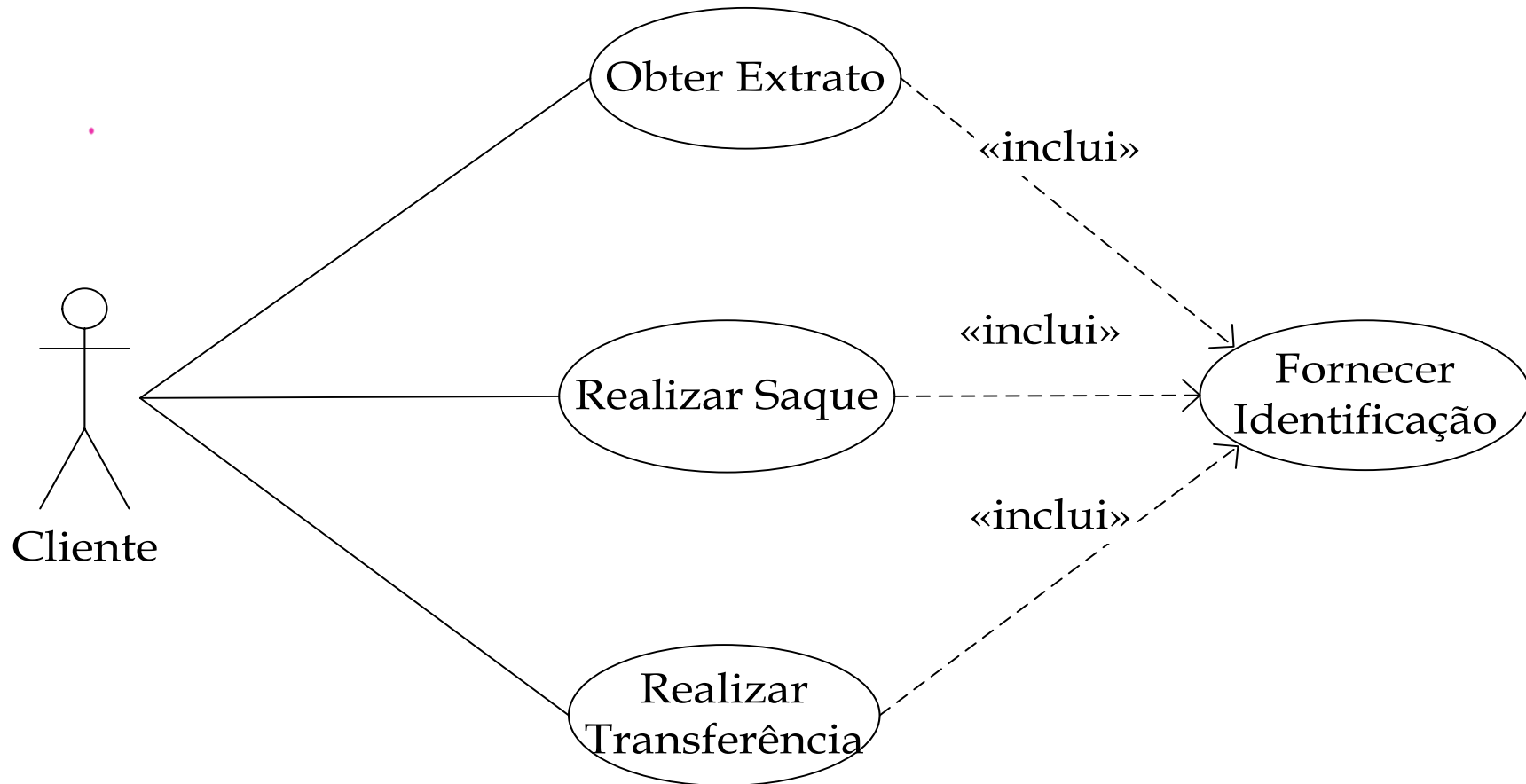
- Diagrama de sequência
- Diagrama de Interatividade
- Diagrama de colaboração ou comunicação
- Diagrama de tempo

## Diagramas da UML



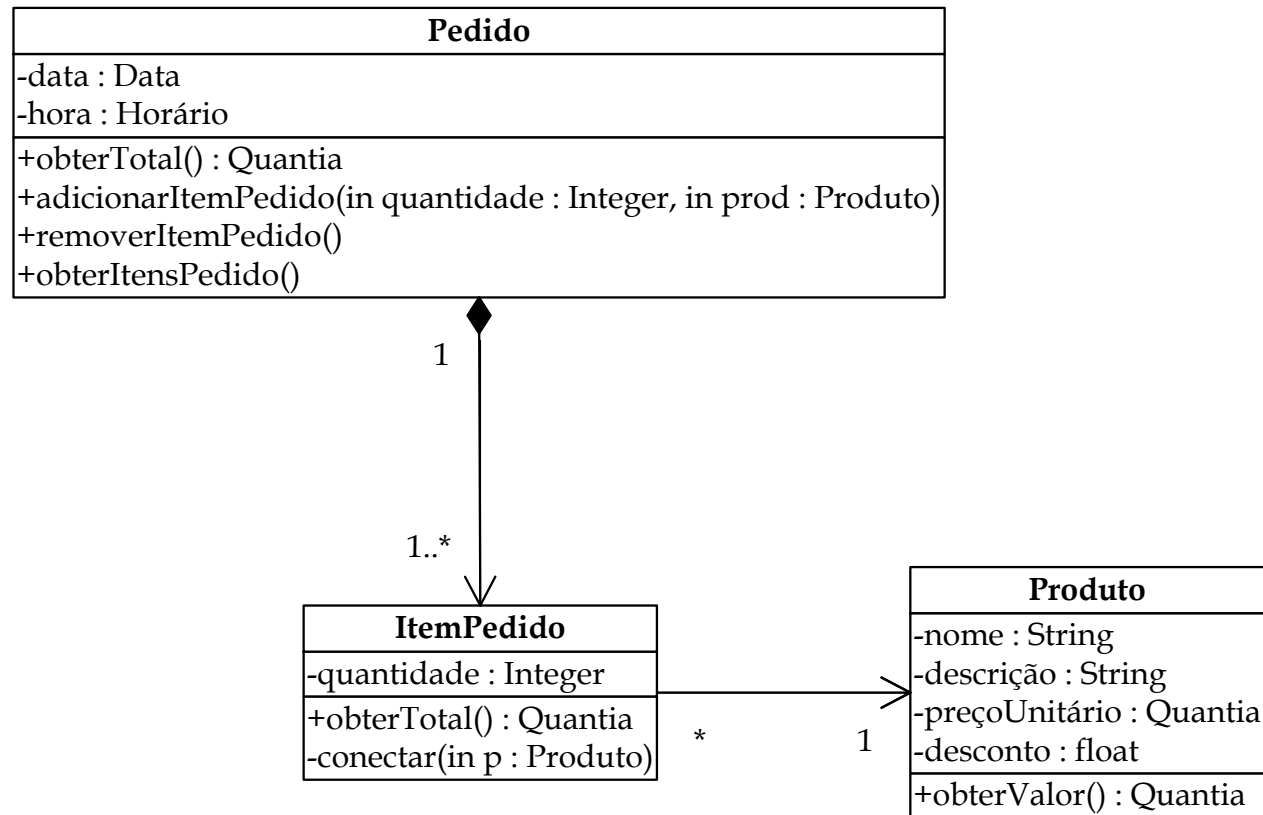
Hierarquia UML 2

## Diagramas de Casos de Uso



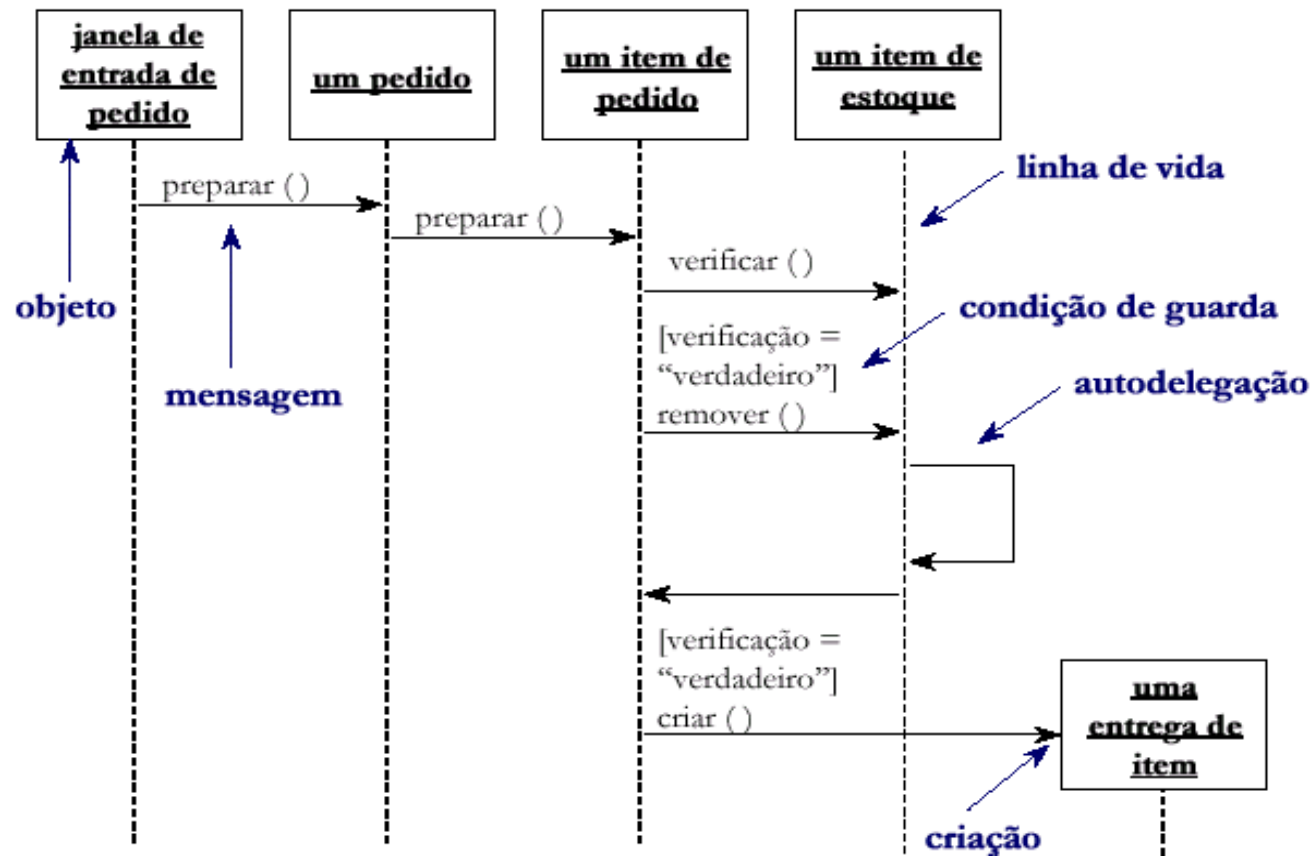
- **Visão "use-case":** Descreve a funcionalidade do sistema desempenhada pelos atores externos do sistema (usuários).

## Diagramas de Classes



- **Visão Lógica ou Projeto:** Descreve como a funcionalidade do sistema será implementada. É feita principalmente pelos analistas e desenvolvedores. **Estuda o sistema internamente.** Trata-se da estrutura estática .

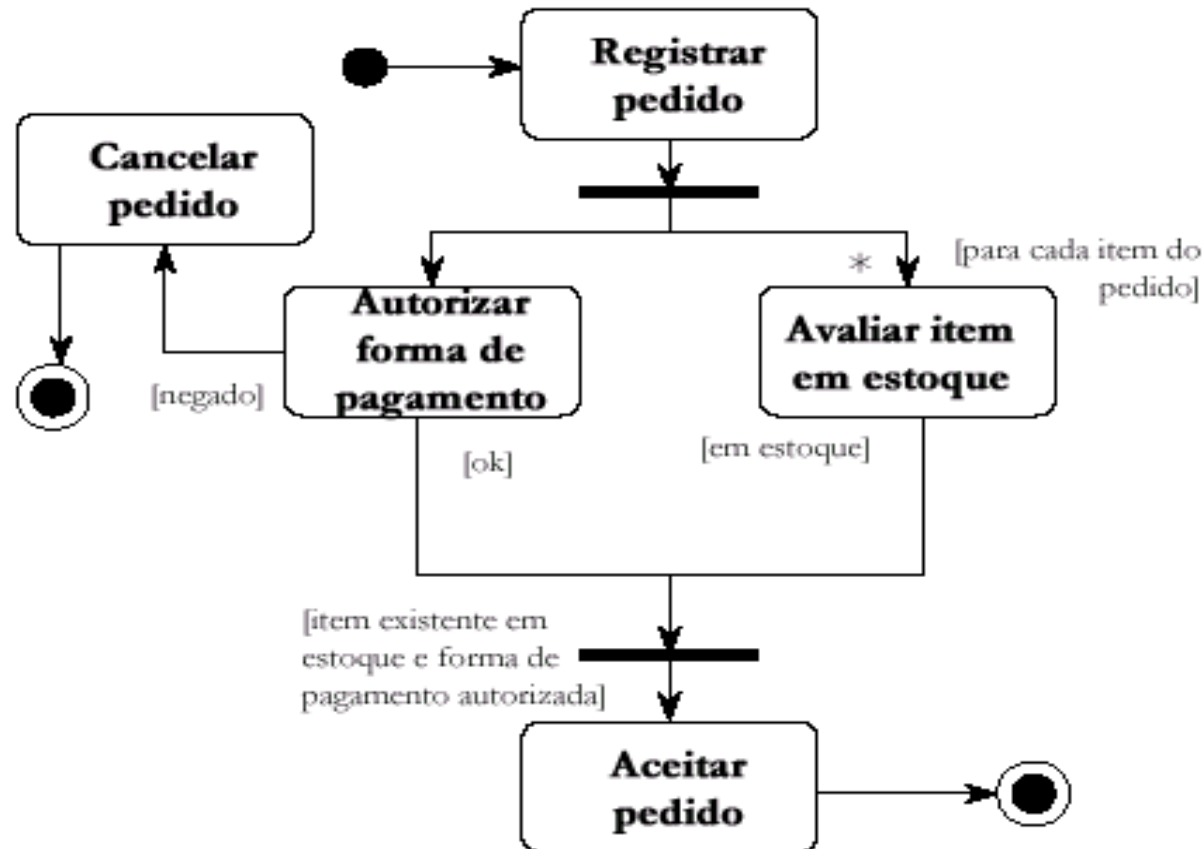
## Diagramas de Seqüência



- **Visão Lógica ou Projeto:** Descreve como a funcionalidade do sistema será implementada. O modelamento dinâmico é descrito pelos diagramas de estado, seqüência, colaboração e atividade.

## Diagramas de Atividades

### Diagrama de atividade

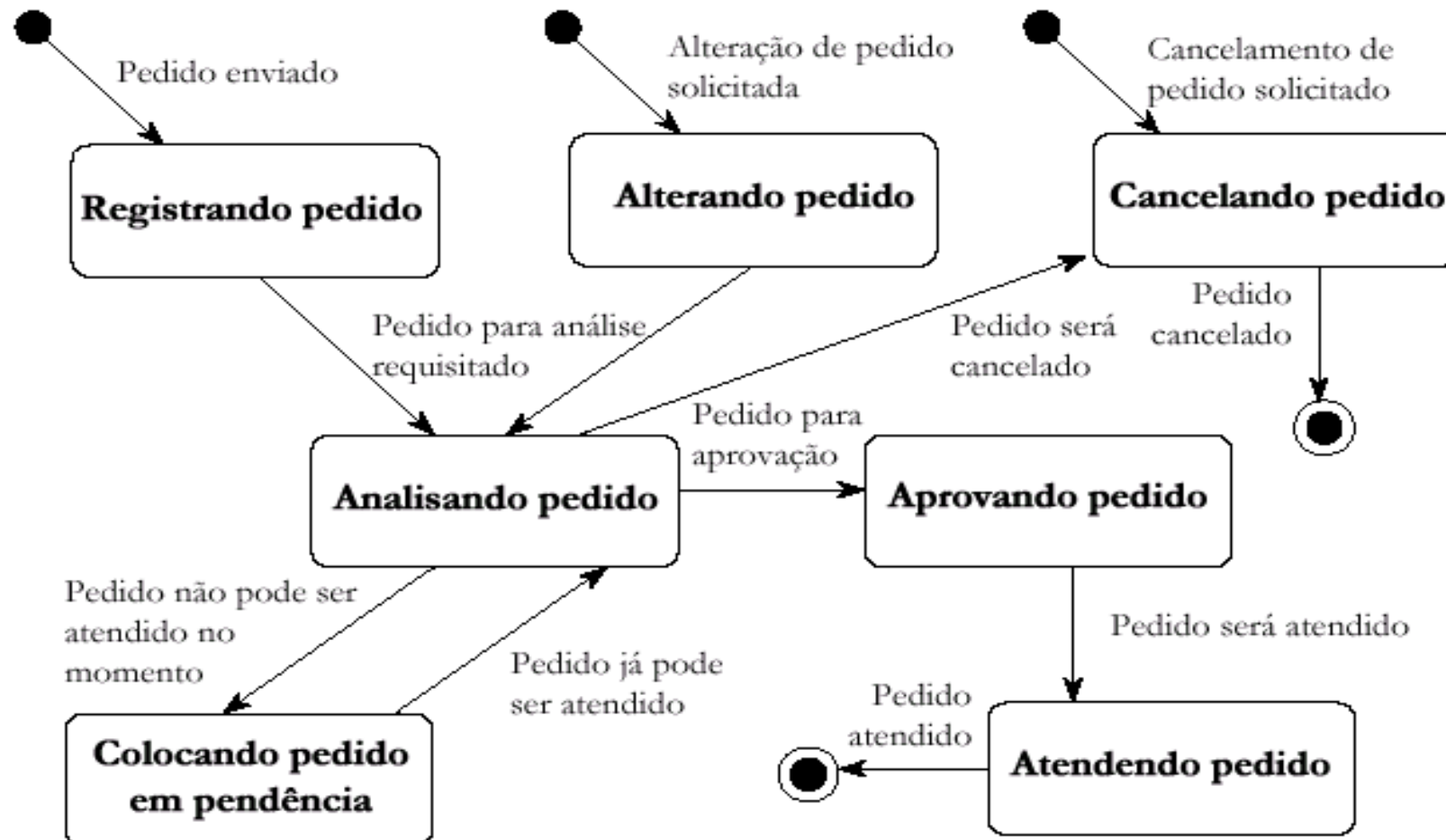


- **Visão de concorrência ou Processo:** Permite uma melhor utilização do ambiente onde o sistema se encontrará.



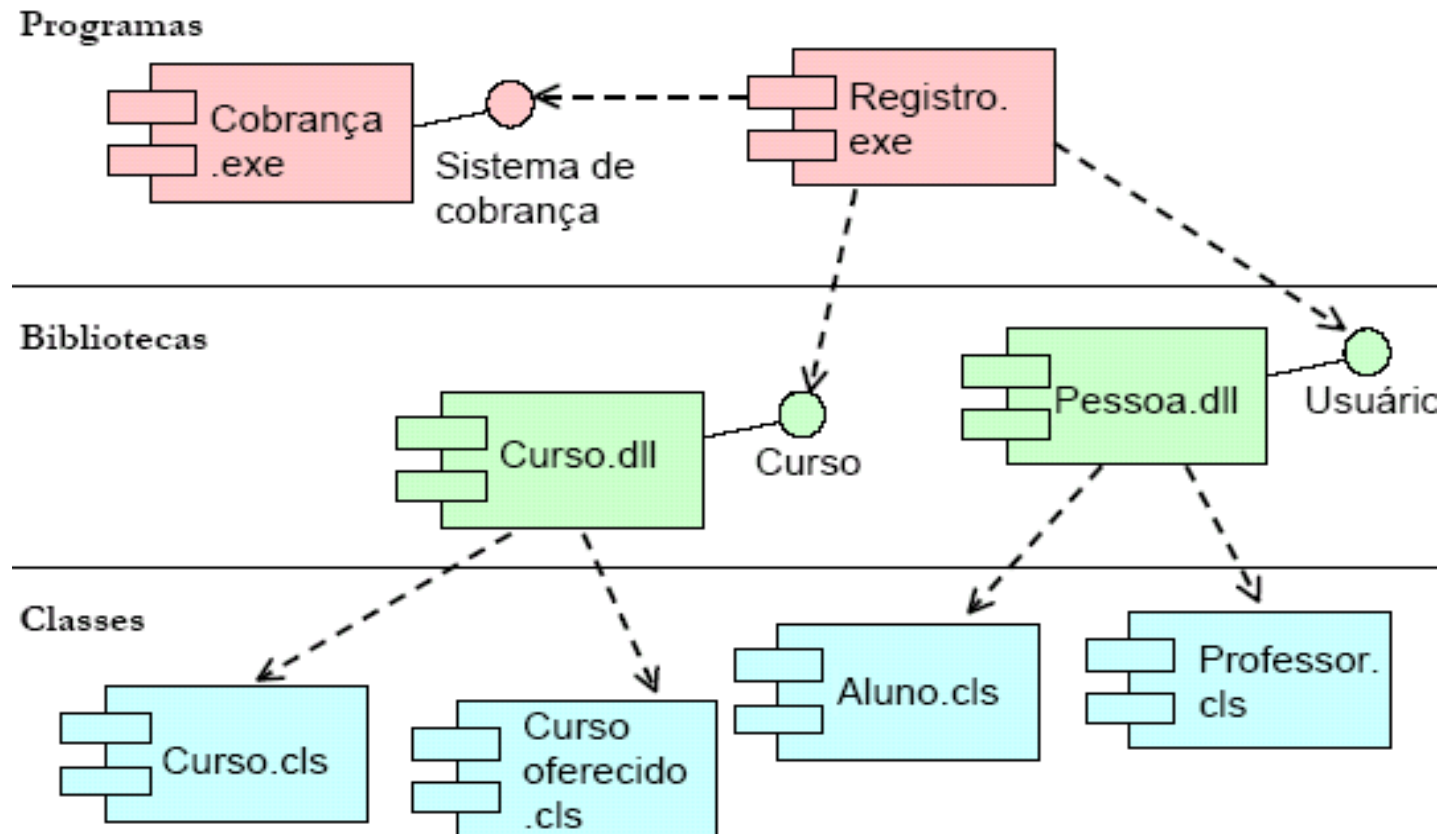
## Diagramas de Estados

### Diagrama de estado para a classe Pedido



## Diagramas de Componentes

### Exemplo de diagrama de componente



- **Visão de Componentes ou Implementação:** É uma descrição da implementação dos módulos e suas dependências.

Prova: FUMARC - 2011 - PRODEMGE - PROVA: Analista de Tecnologia da Informação

Analise as afirmativas abaixo em relação ao Diagrama de Classes da UML.


- I. É um diagrama comportamental da UML.
- II. Pode representar classes e interfaces.
- III. Pode representar relacionamentos de dependência e generalização.

Assinale a alternativa **VERDADEIRA**:

- a) Apenas as afirmativas I e II estão corretas.
- b) Apenas as afirmativas I e III estão corretas.
- c) Apenas as afirmativas II e III estão corretas.
- d) Todas as afirmativas estão corretas.



## Referência:

- 
- Princípios de Análise e Projeto de Sistemas com UML  
autor: Eduardo Bezerra  
editora: Elsevier
  - Modelagem de Objetos através da UML  
autor: José Davi Furlan  
editora: Makron Books

Fonte de imagens: Eduardo Bezerra.

