

# Modelagem de Software



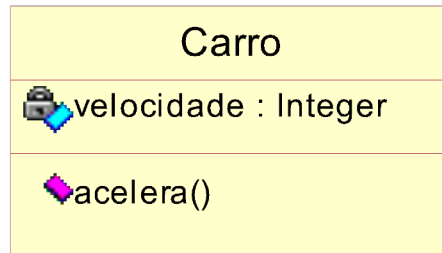
# Fronteira entre análise e projeto

- A orientação a objetos diminui a distância entre as fases do processo de desenvolvimento;

**Código**

**Mundo Real**

**Análise & Projeto**



```
public class Carro
{
    private int velocidade;

    public void accelera()
    {
        velocidade++;
    }
}
```

# Fronteira entre análise e projeto

- A orientação a objetos torna nebulosa a fronteira entre análise e projeto:

## Análise

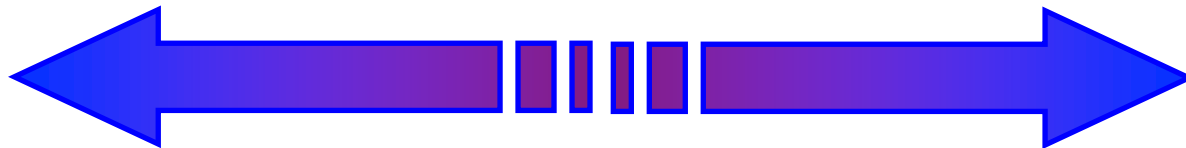


- O que?
- Requisitos
- Investigação

## Projeto



- Como?
- Solução lógica



Onde termina a análise e começa o projeto?

# Fronteira entre análise e projeto

- Principais funções de AOO:
  - Identificar as funcionalidades e entidades do sistema
  - Encontrar abstrações adequadas para representar o problema
- Principais funções de POO:
  - Atribuir responsabilidades às entidades do sistema
  - Encontrar abstrações adequadas para representar a solução
- Ambos são representados através de modelos

**Análise**



Investigação



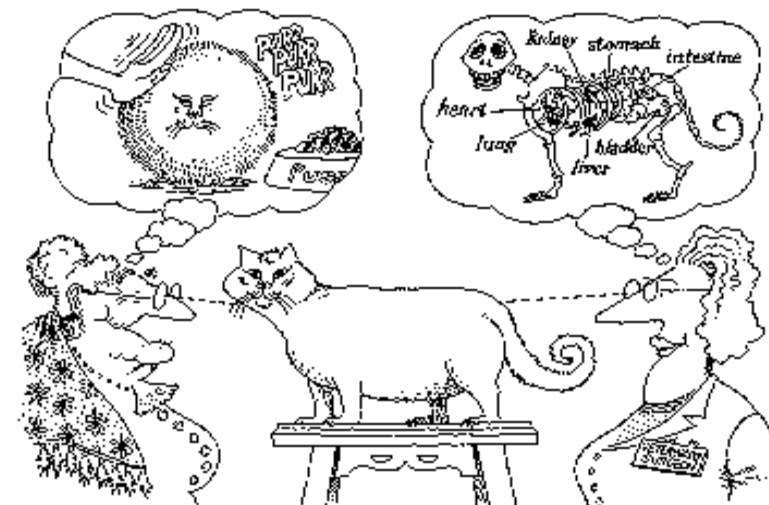
**Projeto**



Solução

# Modelos

- O que são modelos?
  - Abstrações da realidade
  - Focam somente no que realmente interessa para um determinado observador em um dado momento

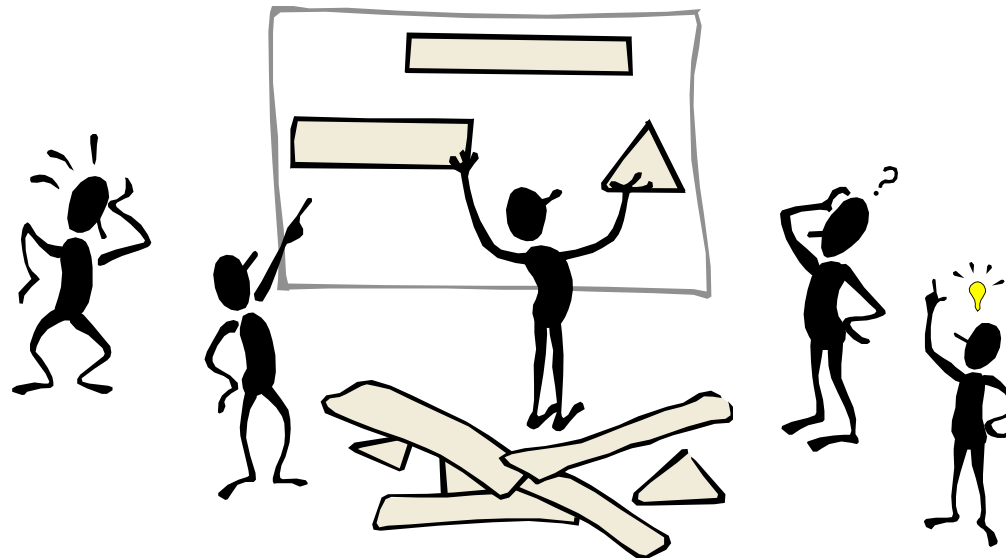


Abstraction focuses upon the essential characteristics of some object, relative to the perspective of the viewer.

[Fonte: BOOCH, G., 1993]

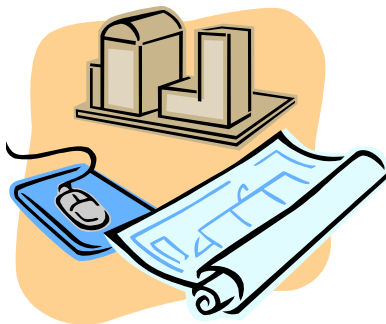
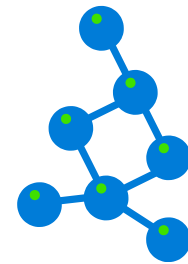
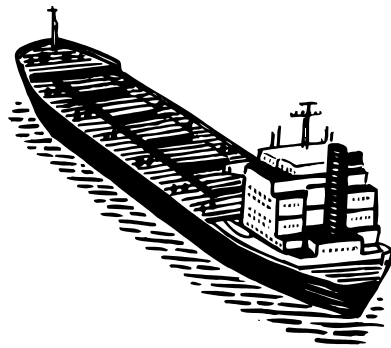
# Modelos

- Para que modelos são úteis?
  - Possibilitar a comunicação entre pessoas
  - Permitir lidar com problemas complexos
  - Testar hipóteses antes de realizá-las



# Modelos

- Quais são as formas de modelos?
  - Croquis
  - Maquetes
  - Manequins
  - Plantas
  - Diagramas
  - Etc.



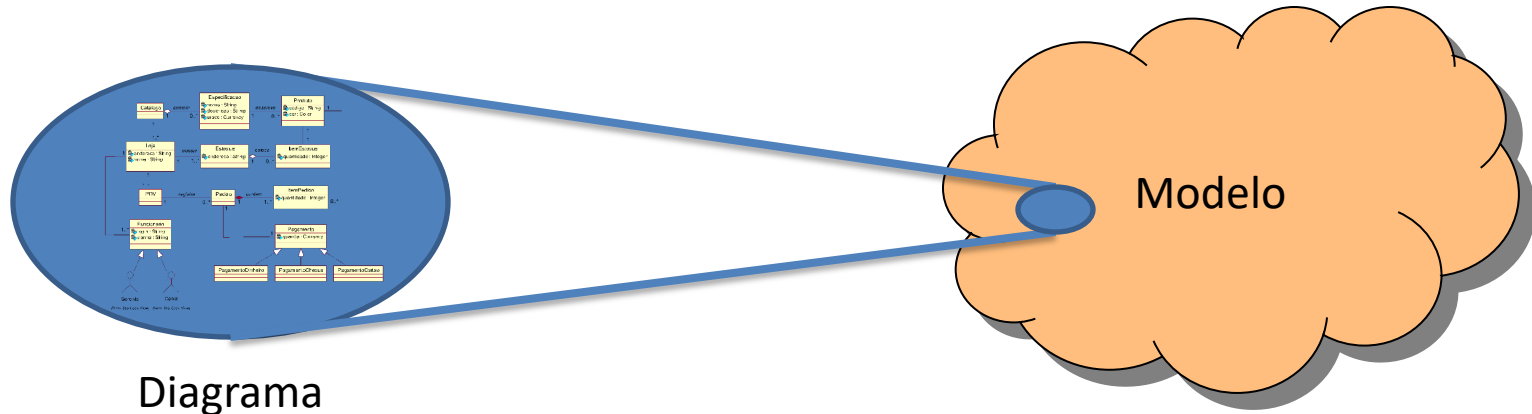
# Modelos

- Quais domínios do conhecimento utilizam modelos?
  - Todos!
- Modelos no domínio de desenvolvimento de software
  - Modelo também é software!
- Modelos servem para apoiar:
  - Derivação dos outros modelos
  - Codificação do sistema
- É preciso questionar a necessidade real de modelos que não servem para a derivação de outros modelos ou para a codificação do sistema!!!



# Modelos x Diagramas

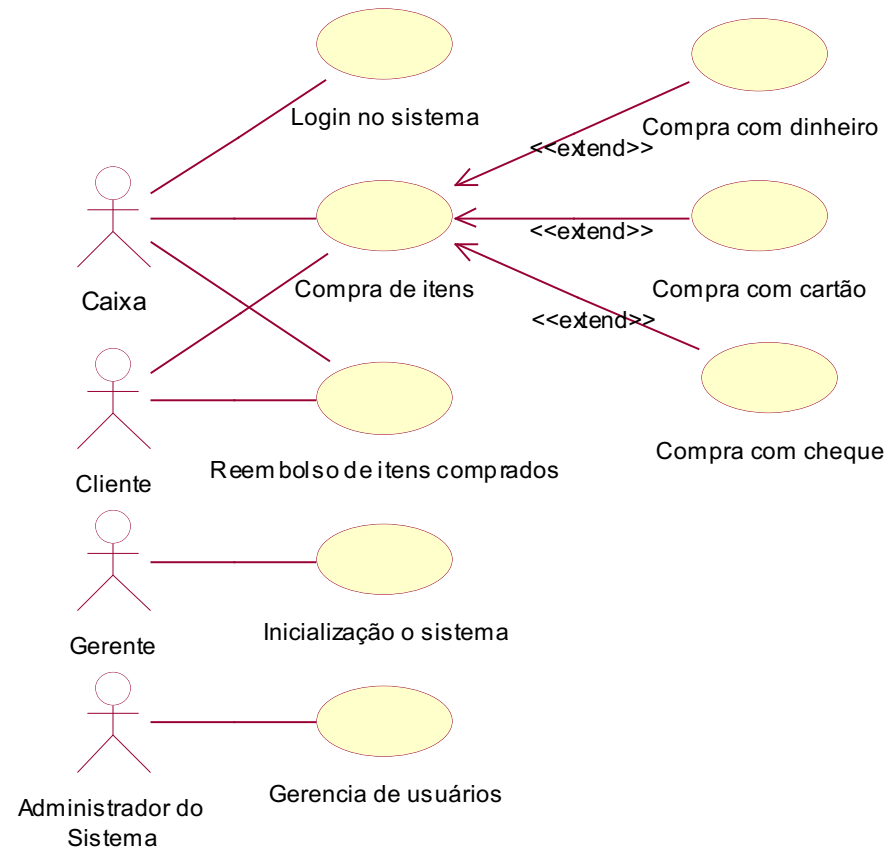
- O modelo contém toda a informação que representa o problema ou a solução
- O diagrama é uma visualização de parte de um modelo sob uma perspectiva
- Ou seja:
  - Se está no diagrama, está no modelo
  - Se não está no diagrama, pode ou não estar no modelo



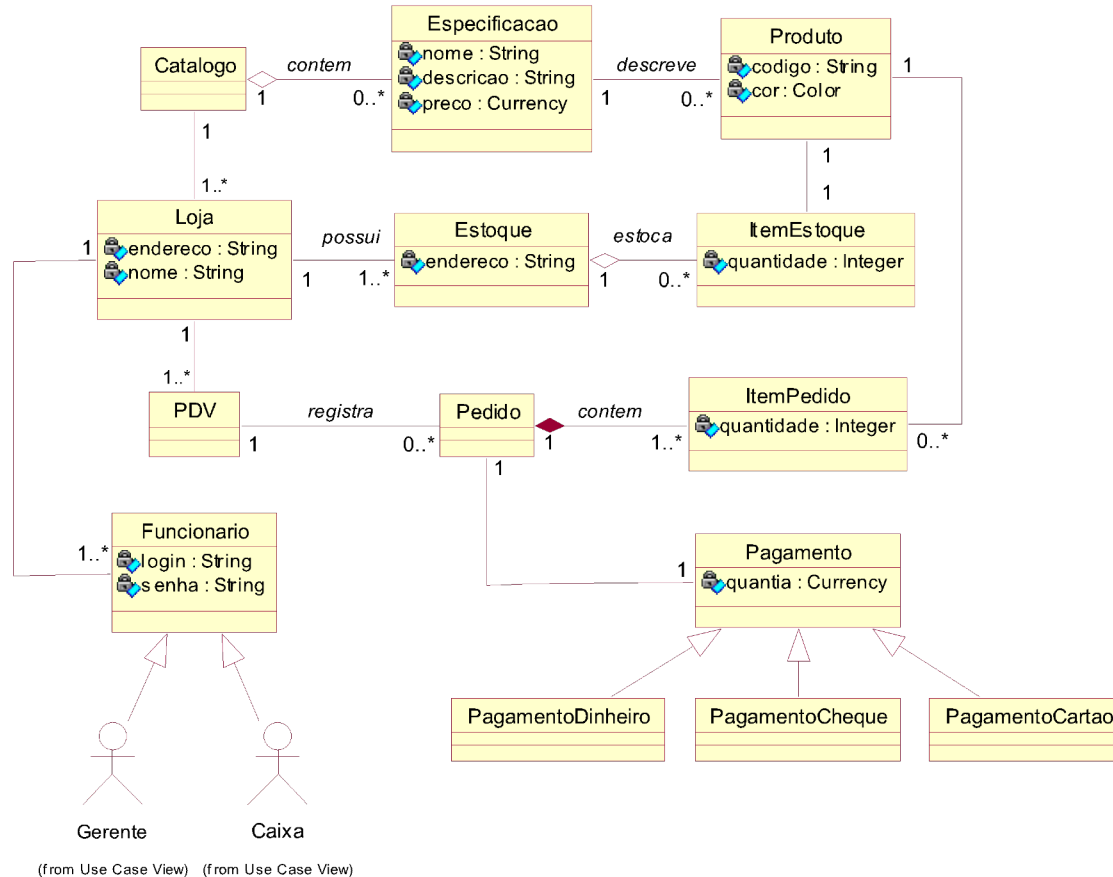
# Exercício

- Modele um sistema de agenda eletrônica usando os recursos que você achar mais relevantes
  - Quais aspectos são capturados?
  - Quais as vantagens da notação que você utilizou?
  - Quais as desvantagens?
  - O que mais você gostaria de capturar que não foi possível com a notação que você utilizou?

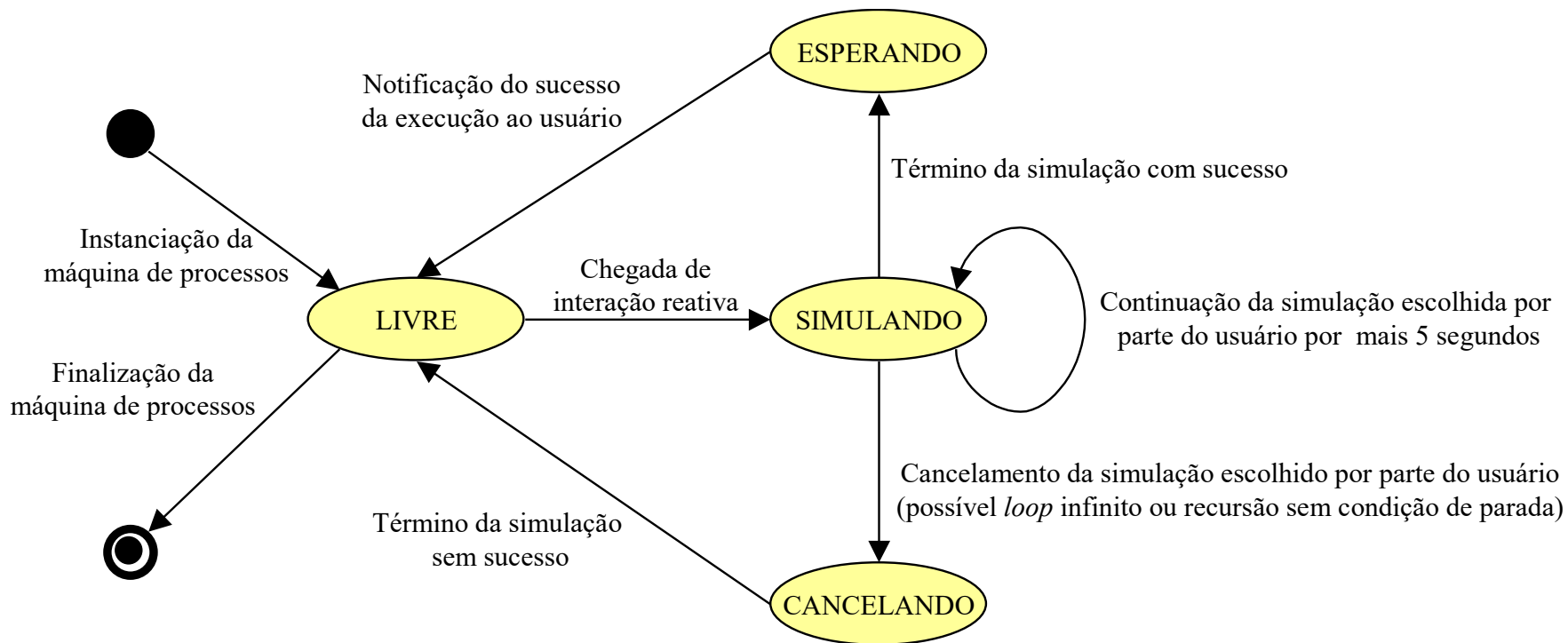
# Exemplo de modelos (diagrama de casos de uso)



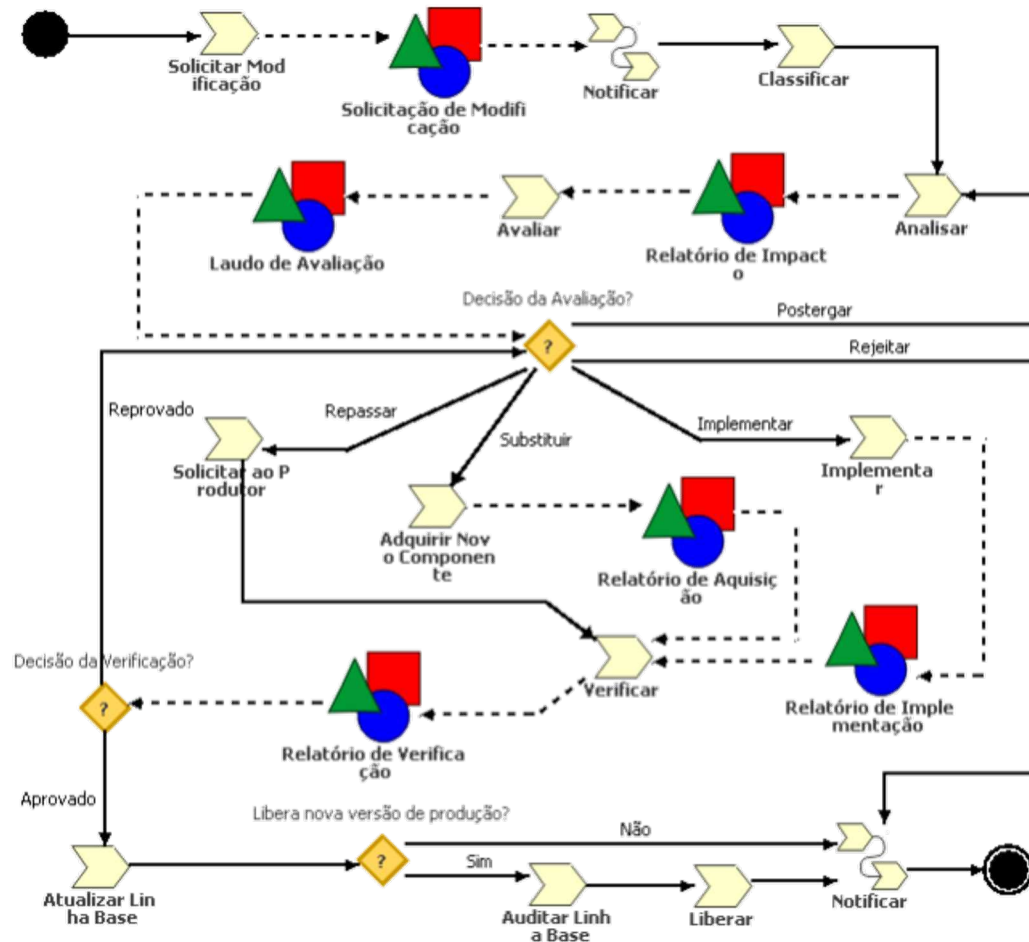
# Exemplo de modelos (diagrama de classes)



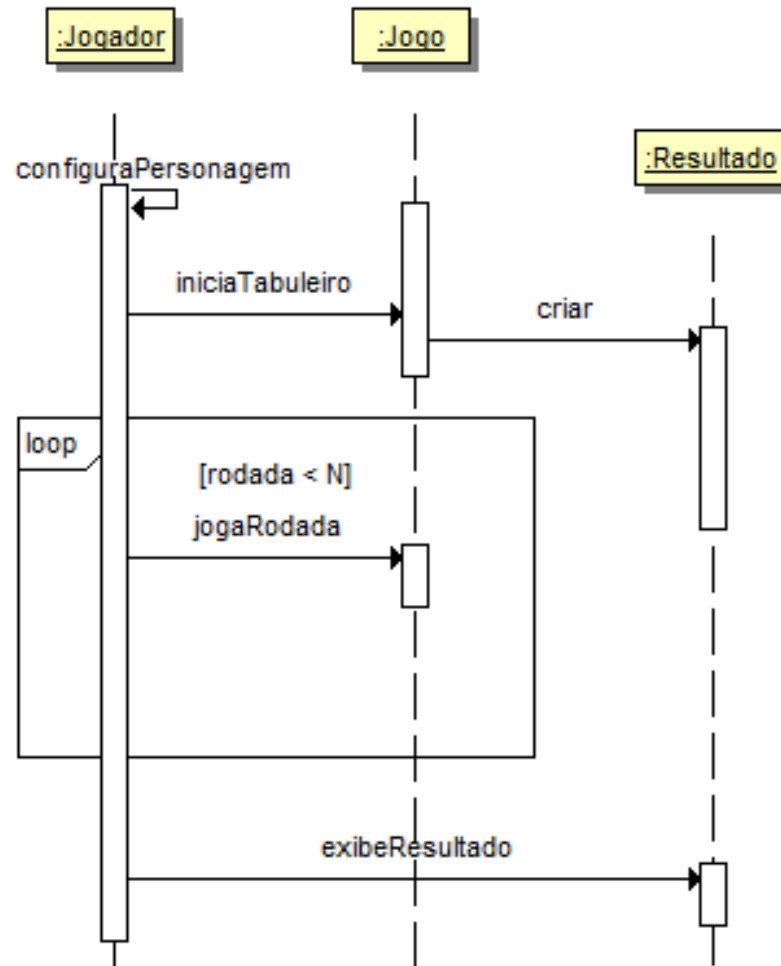
# Exemplo de modelos (diagrama de transição de estados)



# Exemplo de modelos (diagrama de atividades - SPEM)



# Exemplo de modelos (diagrama de seqüência)



# Modelagem de Software

