

---

# Engenharia de Software

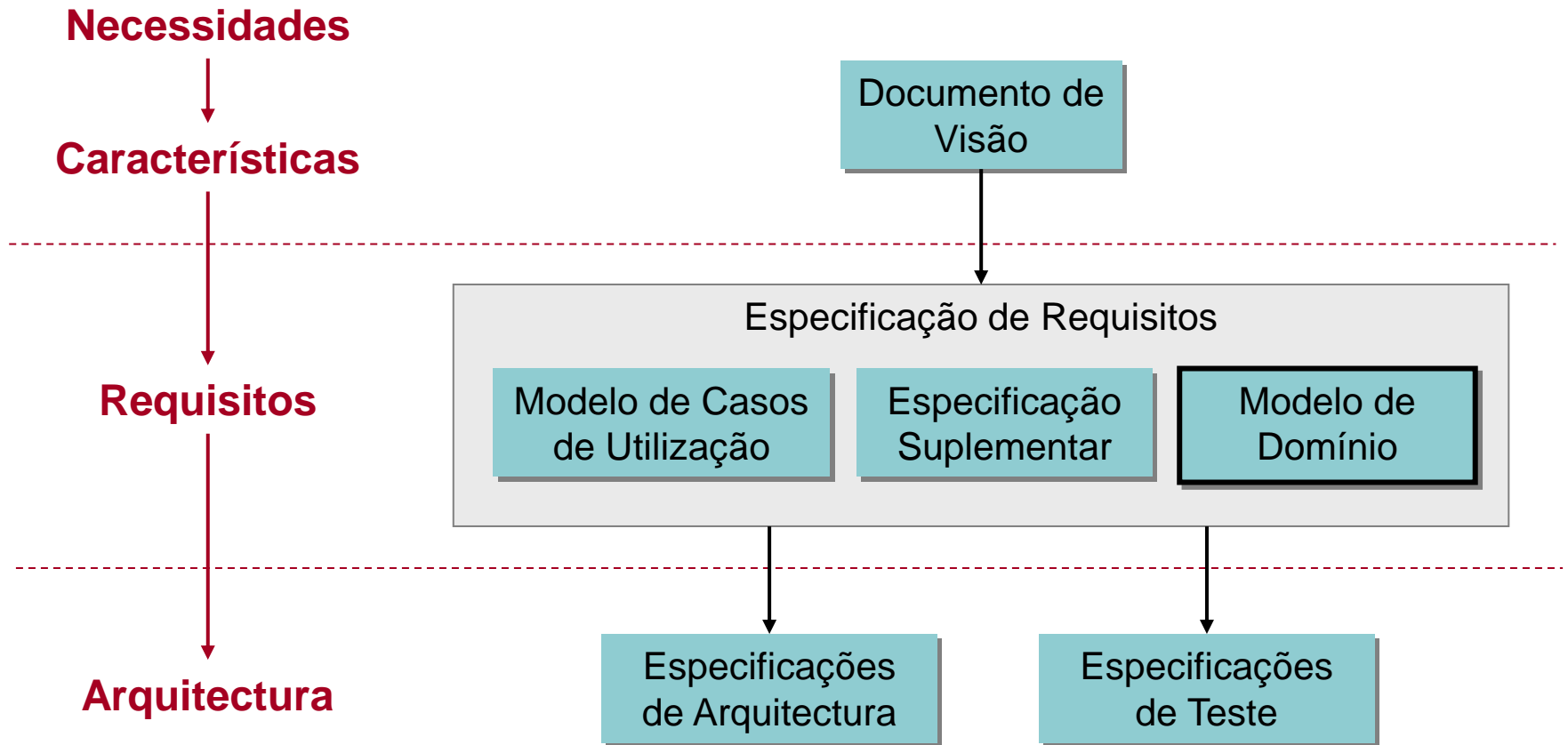
Modelo de Domínio

**Luís Morgado**

Instituto Superior de Engenharia de Lisboa  
Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

---

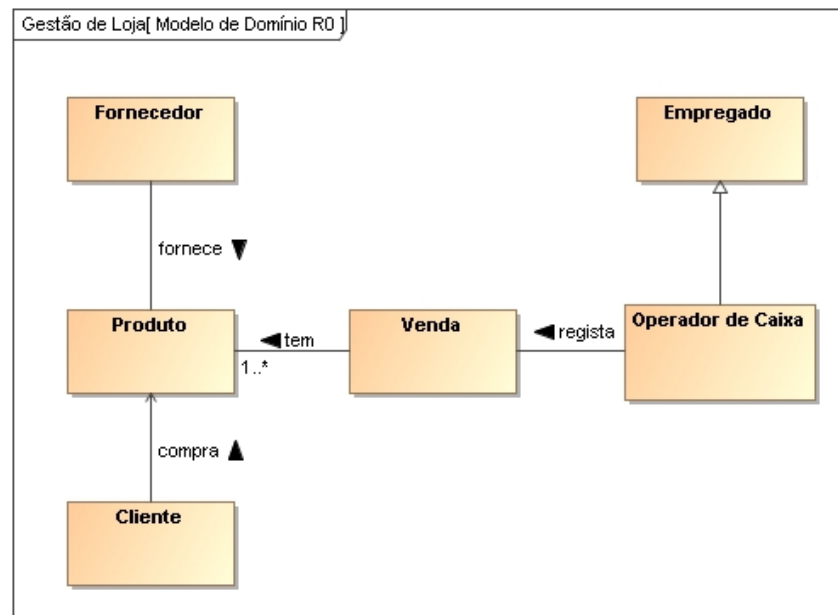
# Análise de Requisitos



# Modelo de Domínio



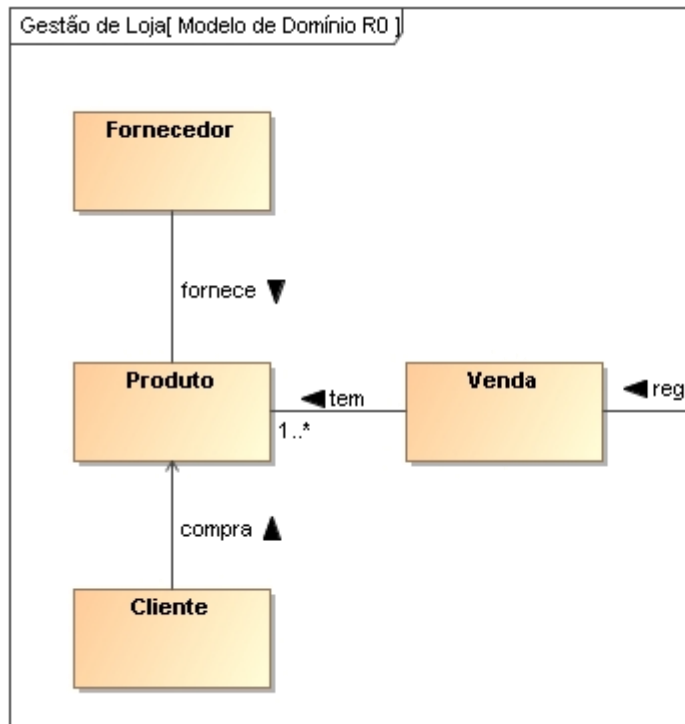
**Domínio do  
problema**



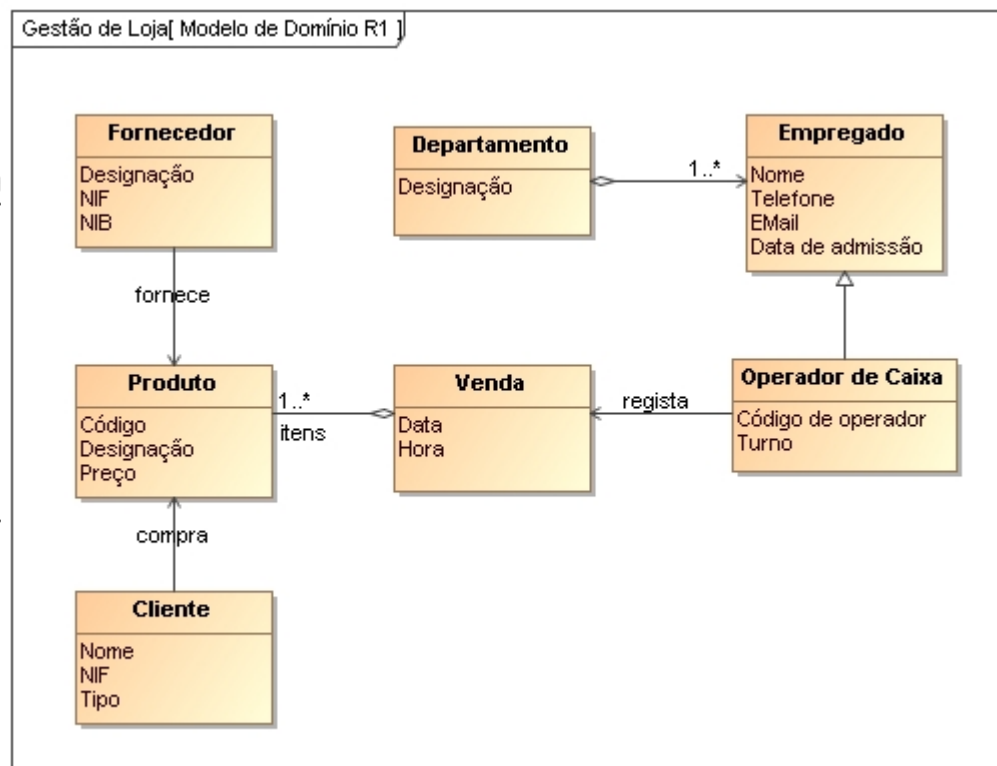
**Domínio da  
solução**

# Modelo de Domínio

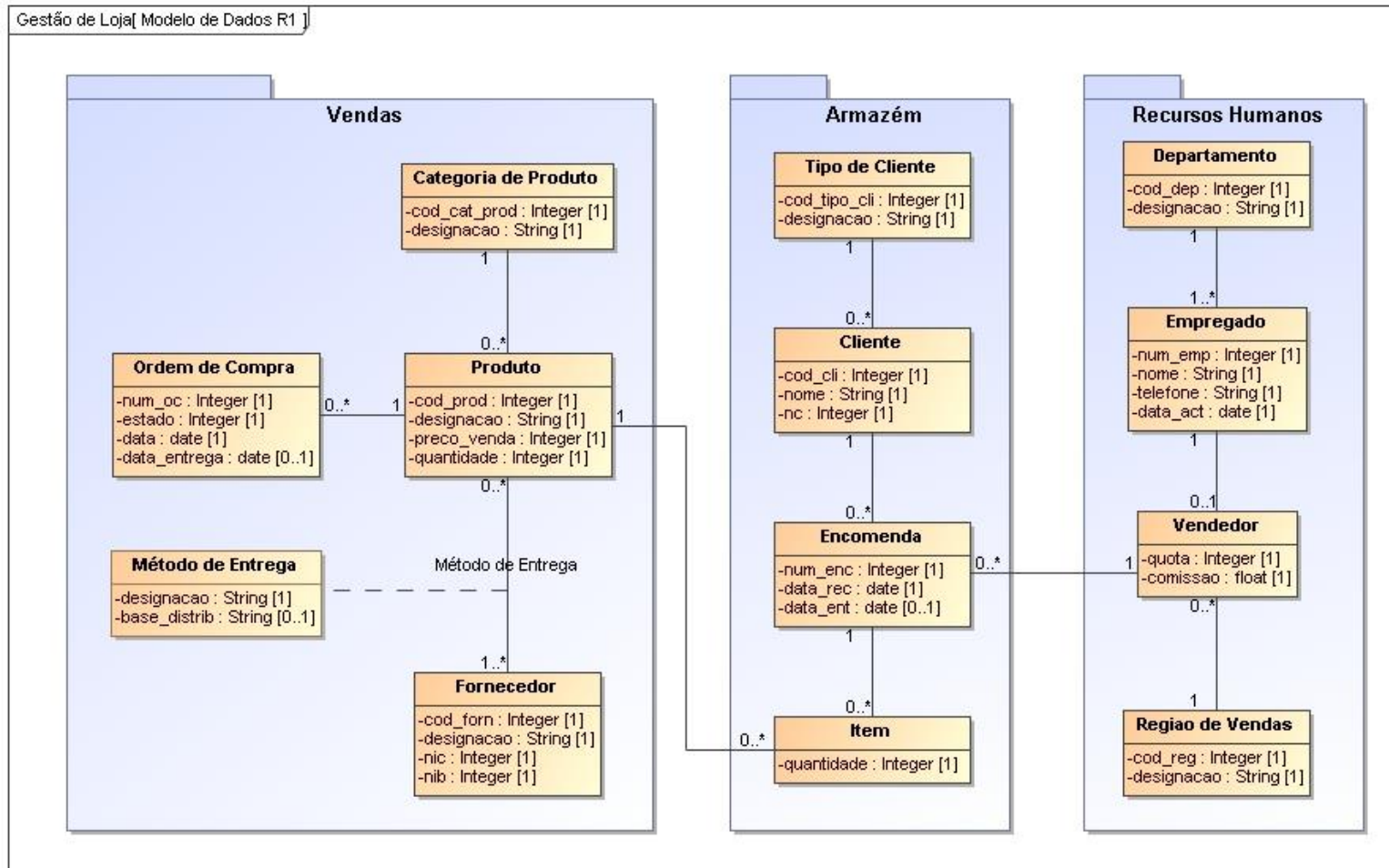
## Início



## Refinamento



# Modelo de Dados



# Modelo de Domínio

---

- Aspecto chave da análise *orientada-a-objectos*
  - Descrição do **domínio do problema** em termos dos **objectos relevantes**
- Modelo de Domínio
  - Representação das **classes e relações conceptuais** que **descrevem as situações reais** de um determinado domínio
  - Foco nos **conceitos**
  - **Não representa classes do domínio da solução (implementação)**
  - Ponte para o domínio da solução (arquitectura e implementação)
  - Classes com estereótipo ***Entidade* (*Entity*)**

# Elementos do Modelo de Domínio

---

- **Entidades (*Classes de Domínio*)**
  - Representam tipos de objectos do domínio do problema
- **Atributos**
  - Representam características das entidades com identificação nominal
- **Associações**
  - Representam relações entre duas ou mais entidades que descrevem as ligações entre as instâncias dessas entidades
- **Restrições**
  - Indicam requisitos adicionais representados através de anotações complementares

# Classes de Domínio

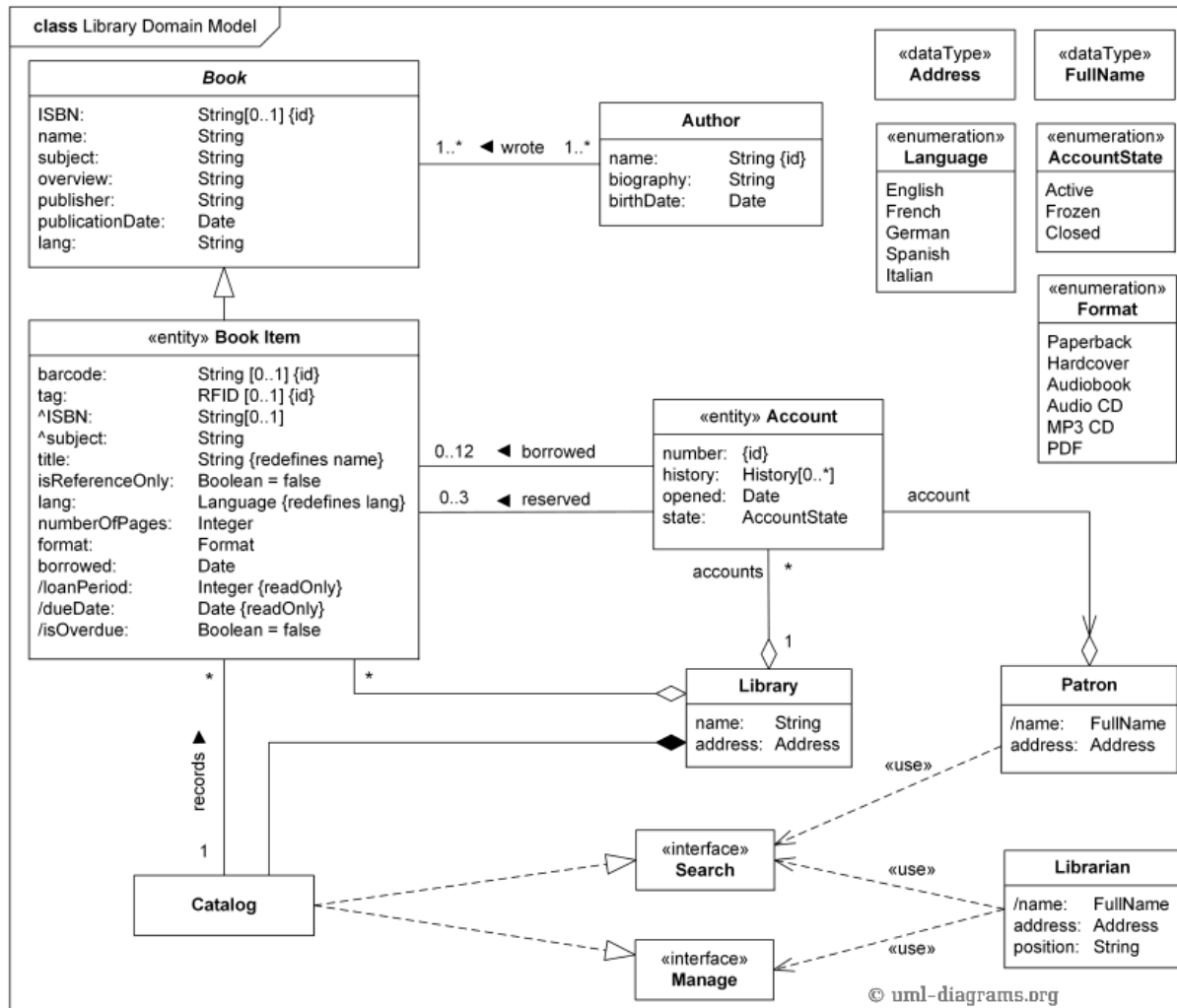
---

- Cada classe de domínio representa um tipo de objecto (representa as características comuns das instâncias)
- Exemplos:
  - **Objectos de negócio**
    - Representam coisas que são manipuladas no negócio (e.g. *Produto*, *Factura*)
  - **Objectos diversos**
    - Representam coisas que não são específicas do negócio (e.g. *Contacto*)
  - **Eventos**
    - Representam a ocorrência de condições ou situações relevantes (e.g. *Venda*)



# Modelo de Domínio

## Exemplo



# Elaboração do Modelo de Domínio

---

- **Artefactos base**
  - Visão
  - Modelo de casos de utilização
  - Especificação suplementar
- **Identificação de elementos**
  - **Nomes**
    - Indicam classes, atributos e objectos
  - **Verbos**
    - Indicam operações e relações

# Modelo de Domínio

- **Filtragem**

- **Eliminar elementos**

- Vagos
    - Ambíguos
    - Redundantes

- **Manter classes que representam**

- Elementos tangíveis (objectos físicos)
    - Entidades conceptuais
    - Categorias
    - Interfaces para o exterior do sistema
    - Papéis

- Informação fornecida pelos utilizadores pode ser redundante, ambígua e incompleta
- Interessa-nos
  - Informação útil
  - O que deve ser mantido

# Modelo de Domínio

## Exemplo: Rede ATM

---

Pretende-se implementar um sistema de gestão de uma rede bancária, que inclui caixas manuais e caixas automáticos (ATM), a ser partilhado por um consórcio de bancos. Cada banco disponibiliza um seu computador para manter contas e processar transacções sobre essas contas. Os caixas manuais pertencem aos bancos individuais e comunicam directamente com os computadores desses bancos. Os caixas manuais inserem dados de contas e de transacções. Os caixas automáticos comunicam com um computador central que coordena as transacções entre bancos. Os caixas automáticos aceitam cartões bancários, interagem com os utilizadores, comunicam com o sistema central para realizar transacções, fornecem dinheiro e imprimem recibos. O sistema necessita de mecanismos de histórico e de segurança. O sistema deve gerir correctamente acessos concorrentes à mesma conta. Os bancos disponibilizam o software dos respectivos computadores; pretende-se o software dos ATM e da rede. O custo do sistema será partilhado pelos bancos de acordo com o número de clientes com cartões bancários.

[Rumbaugh 1991]

# Exemplo

---

## Identificação de Nomes:

sistema  
rede bancária  
mecanismo de histórico  
mecanismo de segurança  
software  
caixa automático  
utilizador  
dados de conta  
dados de transacção  
dinheiro  
recibo  
caixa manual  
ATM  
consórcio  
banco  
funcionário  
computador do banco  
conta  
transacção  
computador central  
cartão bancário  
cliente

[Rumbaugh 1991]

# Exemplo

## Eliminação de classes não relevantes:

sistema  
rede bancária  
mecanismo de histórico  
mecanismo de segurança  
software  
caixa automático  
utilizador  
dados de conta  
dados de transacção  
dinheiro  
recibo

**caixa manual**

**ATM**

**consórcio**

**banco**

**funcionário**

**computador do banco**

**conta**

**transacção**

**computador central**

**cartão bancário**

**cliente**

Vago

Vago

Vago

Vago

Vago

Redundante

Redundante

Atributo

Atributo

Atributo

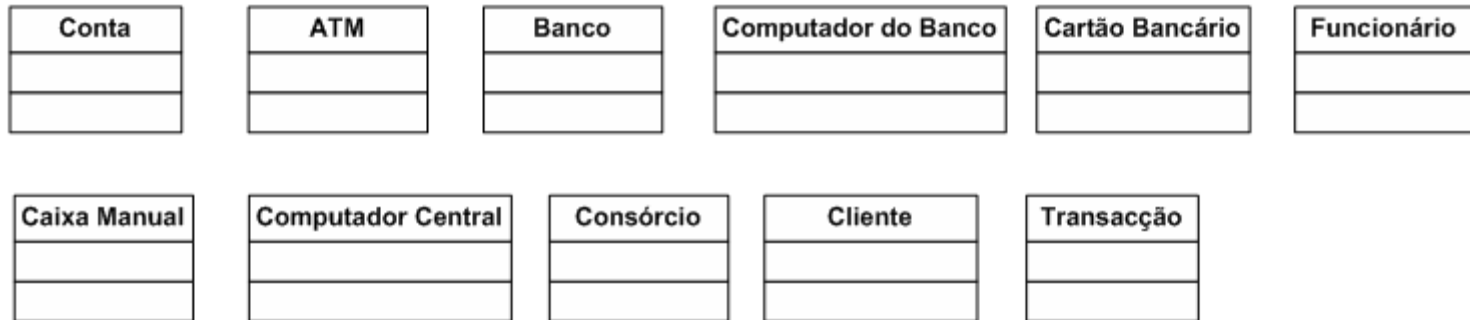
Atributo

← (nomes que descrevem objectos individuais  
devem ser descritos como atributos)

[Rumbaugh 1991]

# Exemplo

**Classes candidatas:**



[Rumbaugh, 1991]

# Exemplo

## Frases com verbos:

consórcio partilha ATMs  
banco disponibiliza computador do banco  
computador do banco mantém contas  
computador do banco processa transacções sobre contas  
caixas manuais pertencem aos bancos  
caixas manuais comunicam com os computadores dos bancos  
funcionários inserem transacções em contas  
caixas automáticos comunicam com um computador central  
computador central coordena transacções entre bancos  
ATMs aceitam cartões bancários  
ATMs interagem com os utilizadores  
ATMs fornecem dinheiro  
ATMs imprimem recibos

## Frases implícitas com verbos:

consórcio é formado por bancos  
bancos gerem contas  
consórcio possui um computador central  
clientes possuem cartões bancários

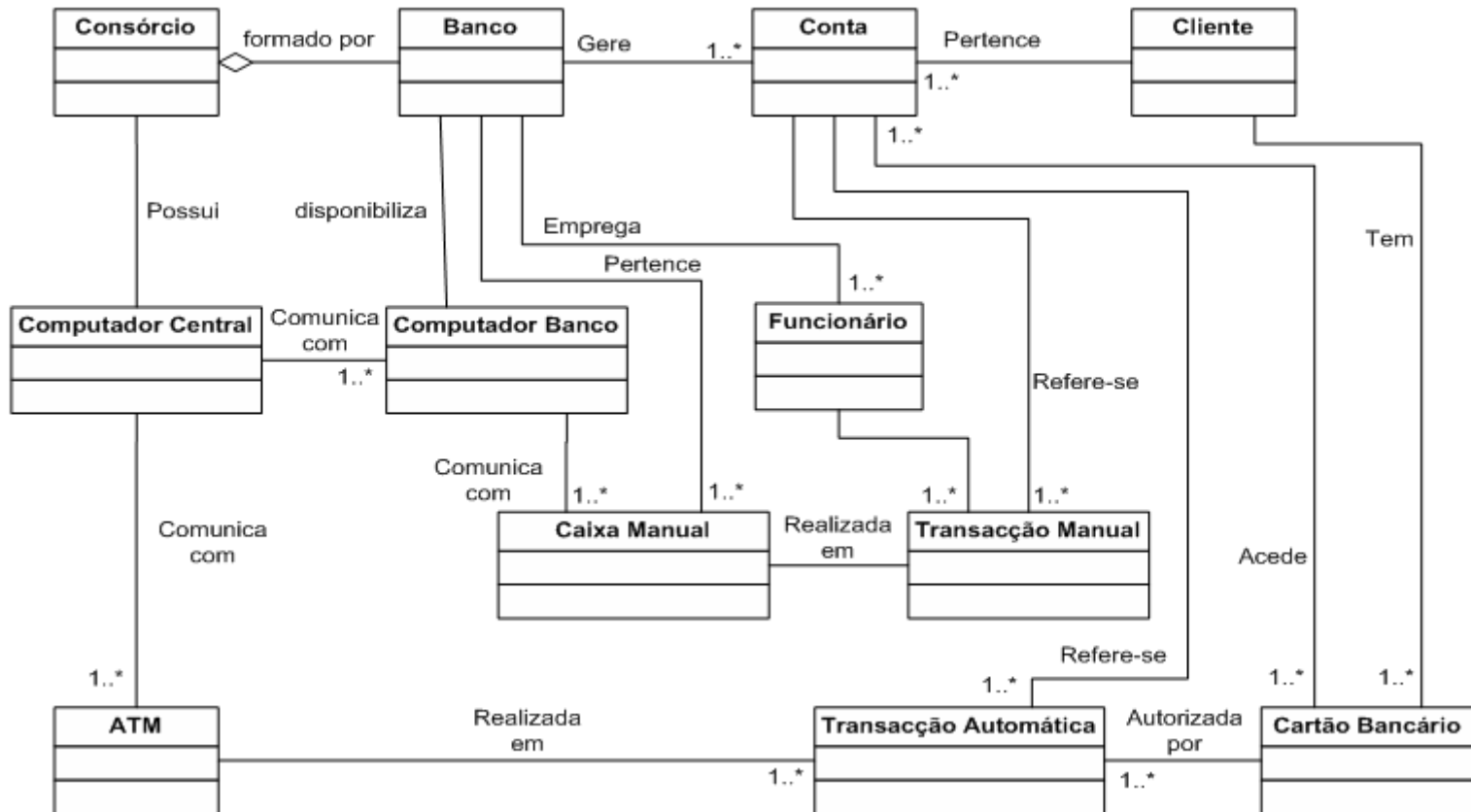
## Conhecimento do domínio do problema:

cartões bancários permitem o acesso a contas  
banco emprega funcionários

[Rumbaugh 1991]



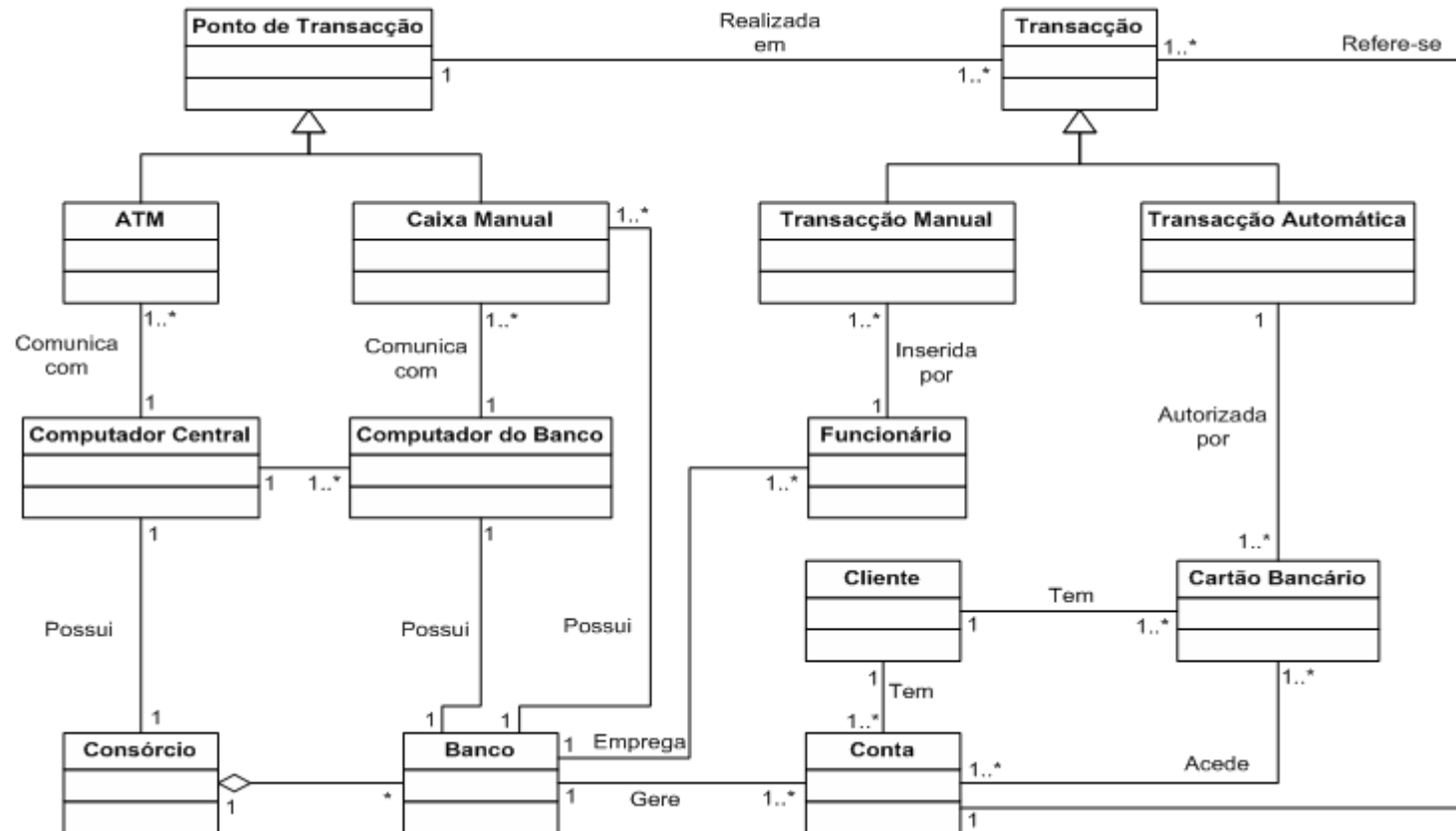
# Exemplo



[Rumbaugh, 1991]

# Exemplo: Rede ATM

## Diagrama de classes – simplificação e generalização



[Rumbaugh, 1991]

# Modelo de Domínio

---

- **Vantagens**

- Proporciona um enquadramento conceptual das entidades do domínio do problema
- Ajuda a perceber o problema
  - Foco na semântica do problema
- Perspectiva estrutural
  - Identificação de requisitos não funcionais
- Perspectiva funcional
  - Identificação de regras de negócio
    - Modelação de fluxos de trabalho
- Com base no modelo de domínio é possível descrever o estado do domínio do problema em qualquer situação

# Modelo de Domínio

