

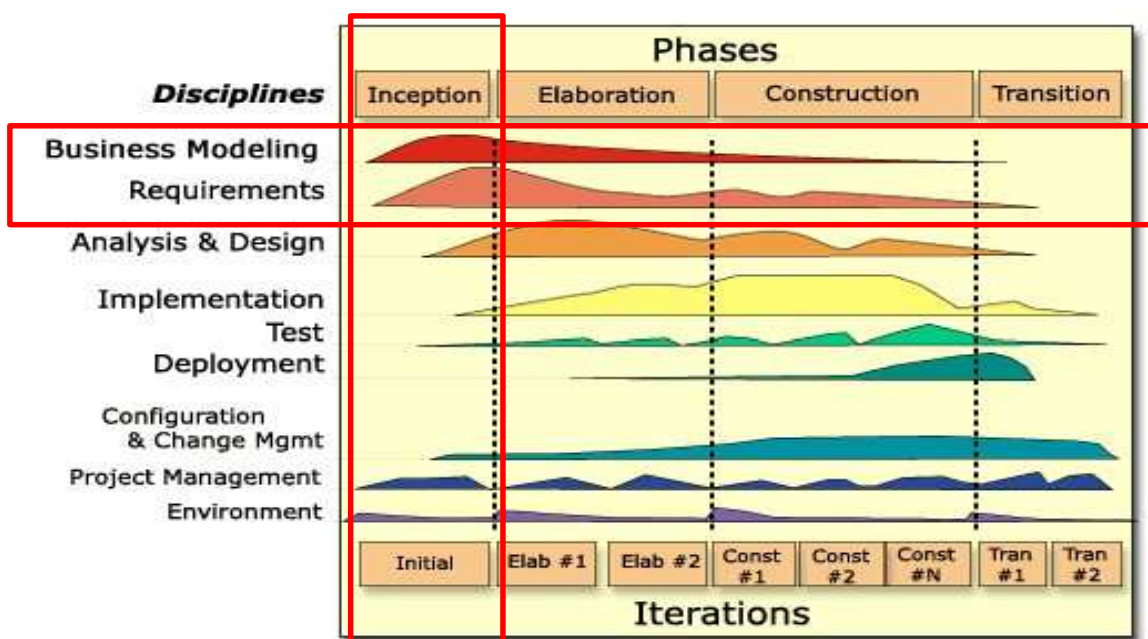


# Desenvolvimento de Sistemas Software

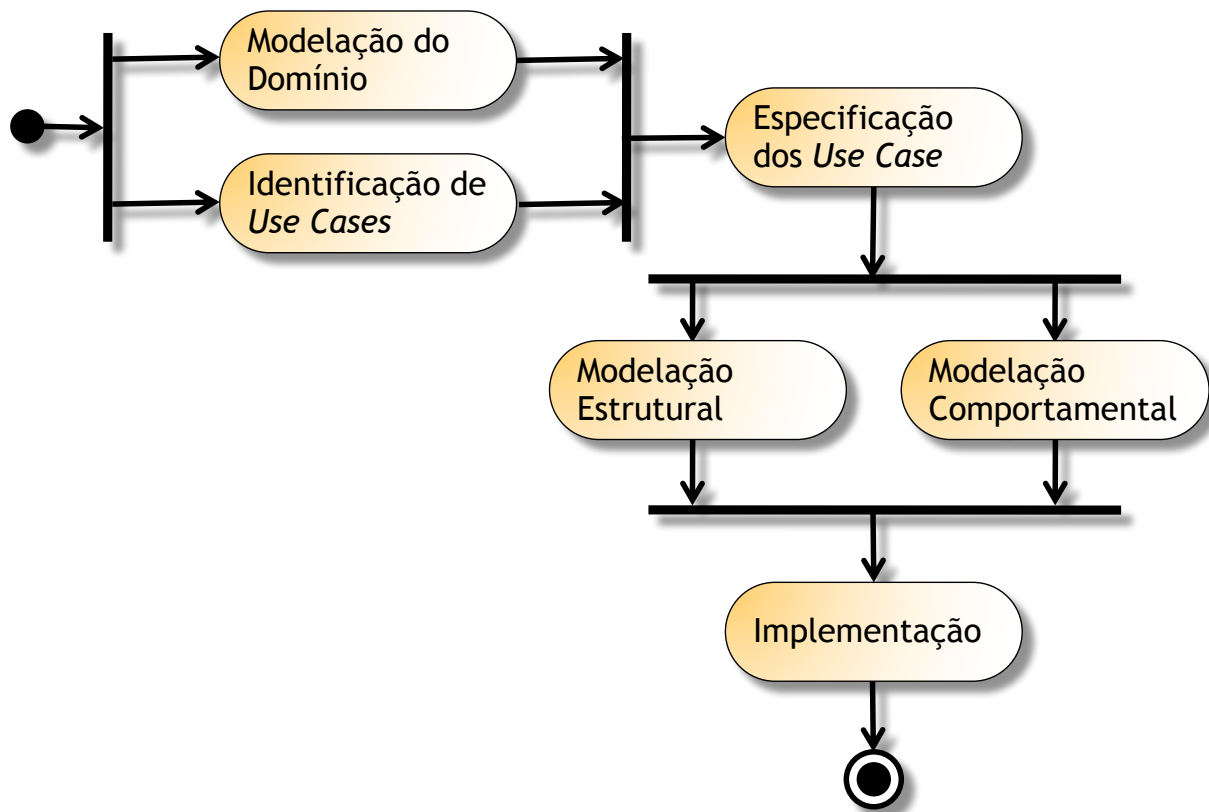
## Aula Teórica 4: Modelação do Domínio



## Resumo das aulas anteriores...



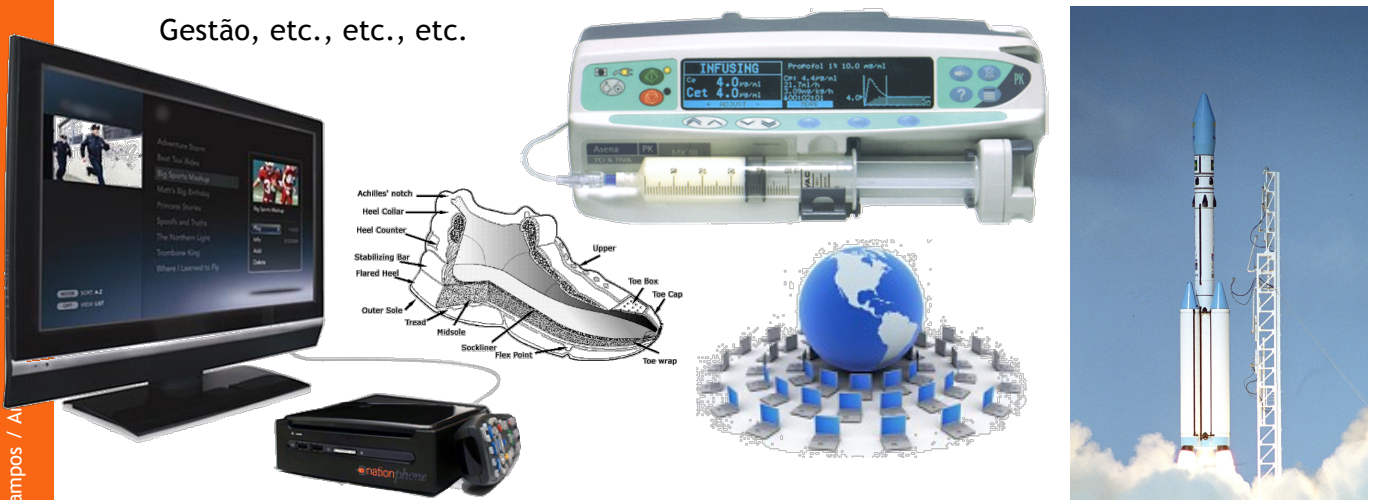
## Resumo das aulas anteriores...



v. 2017/18

## Modelos do Domínio

- Os “produtos” da Engenharia Informática são produzidos nas mais variadas áreas (domínios) de negócio e da sociedade em geral:
  - Medicina, Calçado, Energia, Aero-espacial, Automóvel, Segurança, Telecomunicações, Gestão, etc., etc., etc.



- Em cada projeto, torna-se necessária uma forma de capturar as informações relevantes sobre o *domínio* do projeto.

v. 2017/18



## Modelos do Domínio

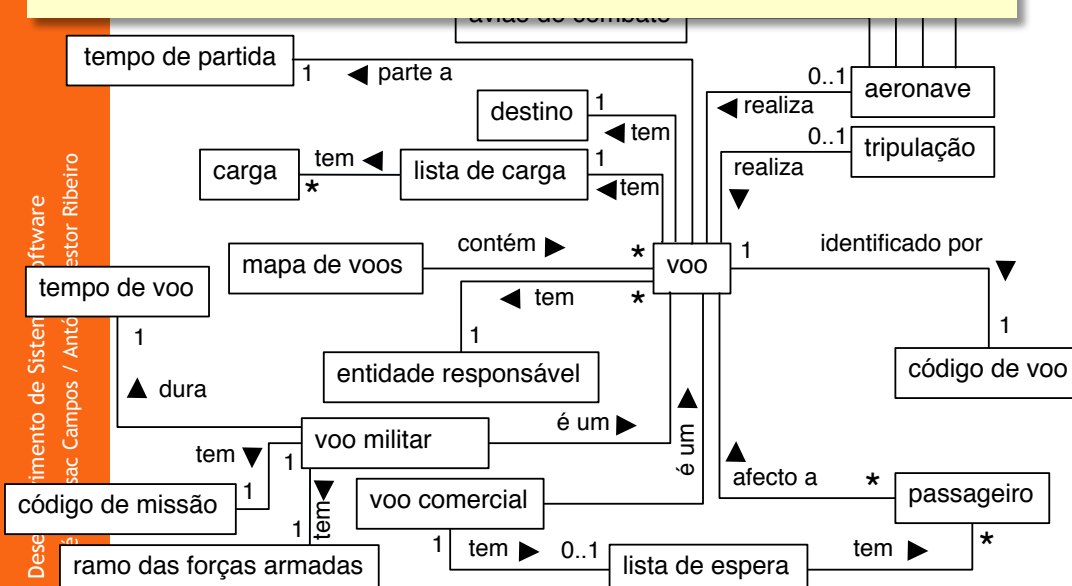
- “A domain model captures the most important types of objects in the context of the business. The domain model represents the ‘things’ that exist or events that transpire in the business environment.” - I. Jacobsen
- O Modelo de Domínio captura as Entidades do problemas e os Relacionamentos entre elas.
- Captura o vocabulário do domínio do problema - fornece um glossário de termos.
- Fornece uma *framework* conceptual para raciocinar sobre o problema - ajuda a pensar
- É uma visão estática do problema - permite representar regras de negócio invariantes no tempo
- É a base para a análise de requisitos.

No **Mapa de Voos** do dia do AEROGEST, cada voo é identificado por um código de voo, tem uma entidade responsável, um conjunto de passageiros afectos a tal voo (caso seja um voo comercial, poderá ter, ou não, uma eventual lista de espera de passageiros substitutos), um conjunto de cargas a embarcar (definida numa **lista de carga de produtos**), um destino, e um tempo de partida. Uma **aeronave** específica capaz de realizar tal voo e uma **tripulação**, ser-lhe-ão posteriormente associadas também.

Um voo comercial é o mais usual e mais bem conhecido. Um voo militar deverá ter a si associada a seguinte informação adicional: tempo de voo, ramo das forças armadas e código de missão para comunicação (exº DELTA77).

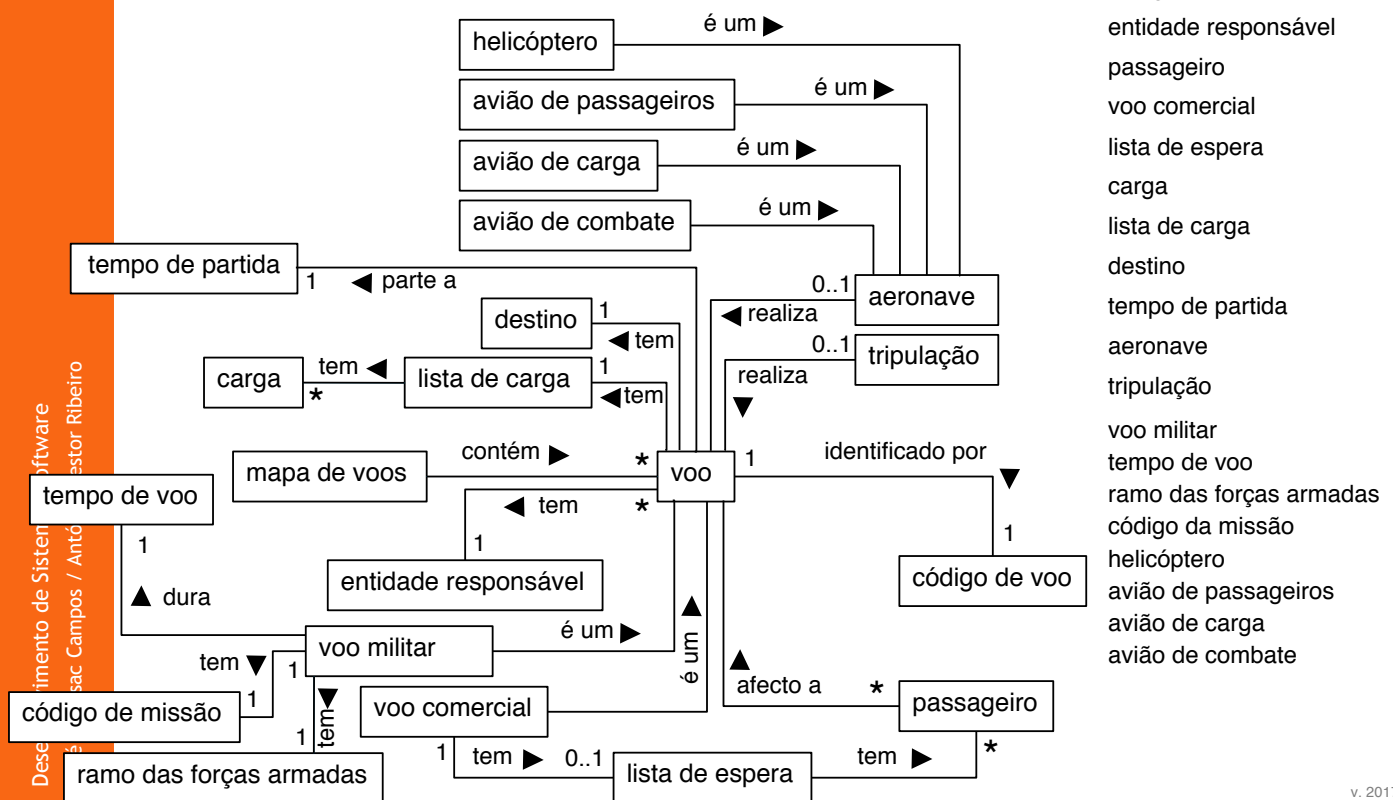
Uma **Aeronave** é uma entidade genérica capaz de voar, que poderá representar um helicóptero, um avião de passageiros, um avião de carga, um avião de combate, etc.

mapa voos  
voo  
código de voo  
entidade responsável  
passageiro  
voo comercial  
lista de espera  
carga  
lista de carga  
destino  
tempo de partida  
aeronave  
tripulação  
voo militar  
tempo de voo  
ramo das forças armadas  
código da missão  
helicóptero  
avião de passageiros  
avião de carga  
avião de combate





## Exemplo



v. 2017/18



## Modelos de Domínio

- O Modelo de Domínio é estático - não representa fluxos de dados
- O Modelo de Domínio representa o problema - não inclui o sistema (software) a desenvolver
- As entidades no Modelo de Domínio são apenas candidatas a serem classes na solução
- As entidades no modelo de domínio podem ter atributos, mas devem ser de tipos simples (números, strings, etc.) e nunca outras entidades
  - Na dúvida, optar por entidades e relacionamentos, em vez de atributos

v. 2017/18



## Algumas notas sobre entidades

- Entidades no modelo de domínio correspondem a “substantivos” na descrição
- Algumas regras para ponderar rejeição de entidades (a partir dos substantivos)
  - É sinónimo de outra entidade?
    - tempo de partida / tempo de voo? código de voo vs. código de missão?
  - Está fora do âmbito da análise?
    - AEROGEST...
  - Refere-se a relações entre outras entidades?
    - “cada voo tem um **conjunto de** passageiros”
    - “voo comercial poderá ter tem uma **lista** de espera de passageiros”
  - É fruto do estilo de escrita?
    - “Um voo militar deverá ter a si associada a seguinte **informação adicional:**”

v. 2017/18



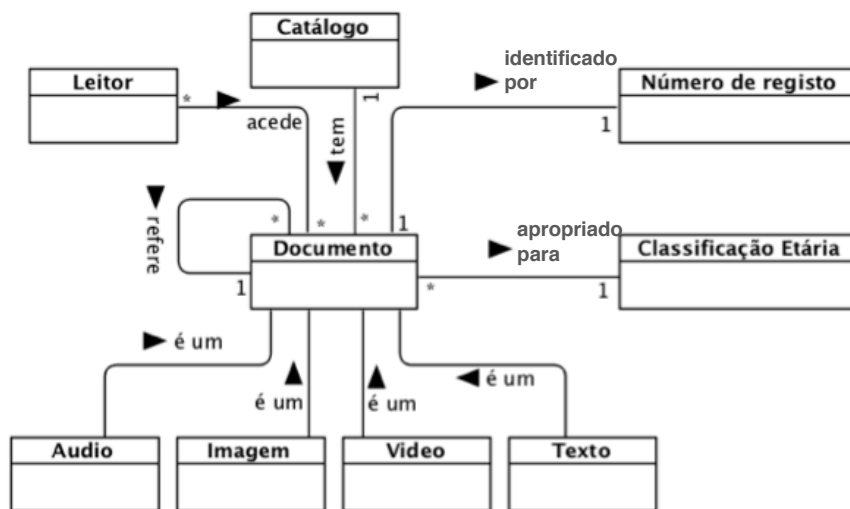
## Algumas notas sobre relações

- Relações no modelo de domínio correspondem a “verbos” na descrição
  - Tal como para as entidades, são apenas candidatas a existirem na arquitectura da solução.
- Relação “é um(a)”
  - Explicita relação de tipagem
  - Representação posterior em OO
    - Herança - classe / sub-classe
    - Realização - classe / interface
    - Atributo na classe

v. 2017/18



## Outro exemplo

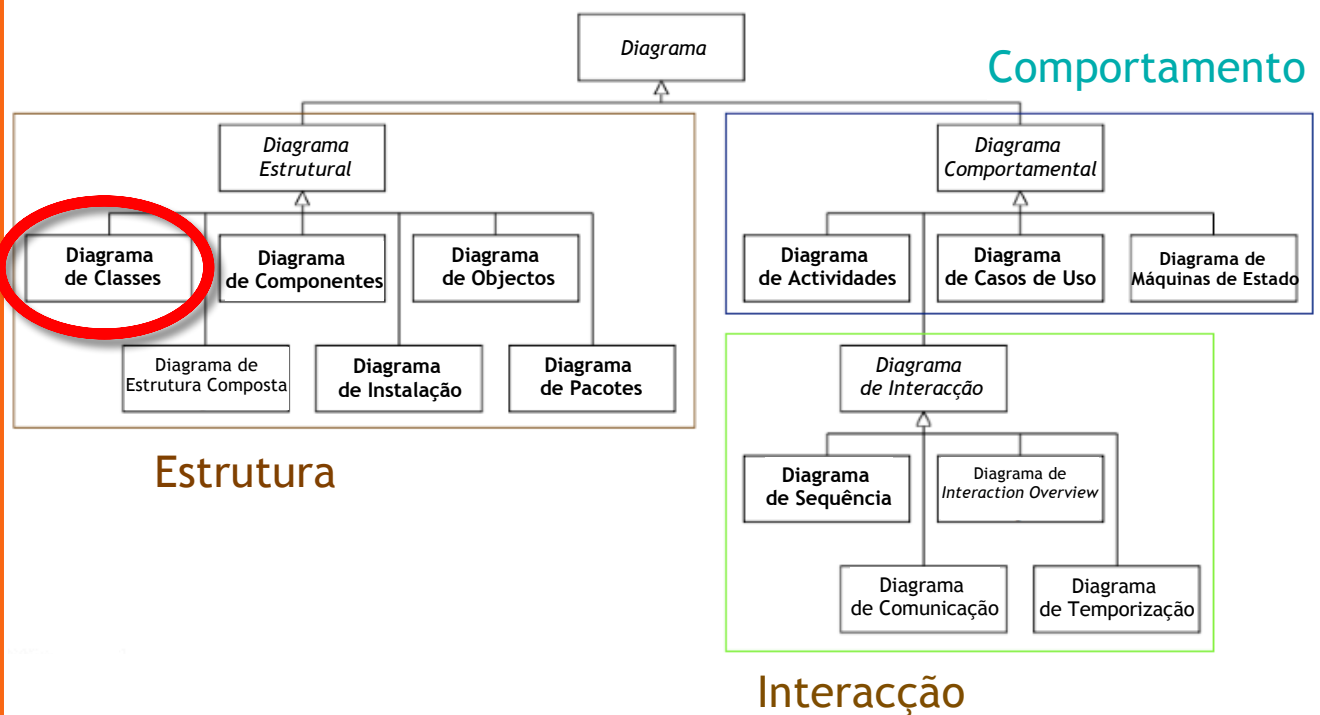


Um catálogo (con)tem documentos que são acedidos por leitores. Um documento pode ser audio, imagem, vídeo, ou texto e tem sempre em número de registo e uma classificação etária. Cada documento refere sempre um outro documento(!).

v. 2017/18



## Diagramas da UML 2.x



v. 2017/18



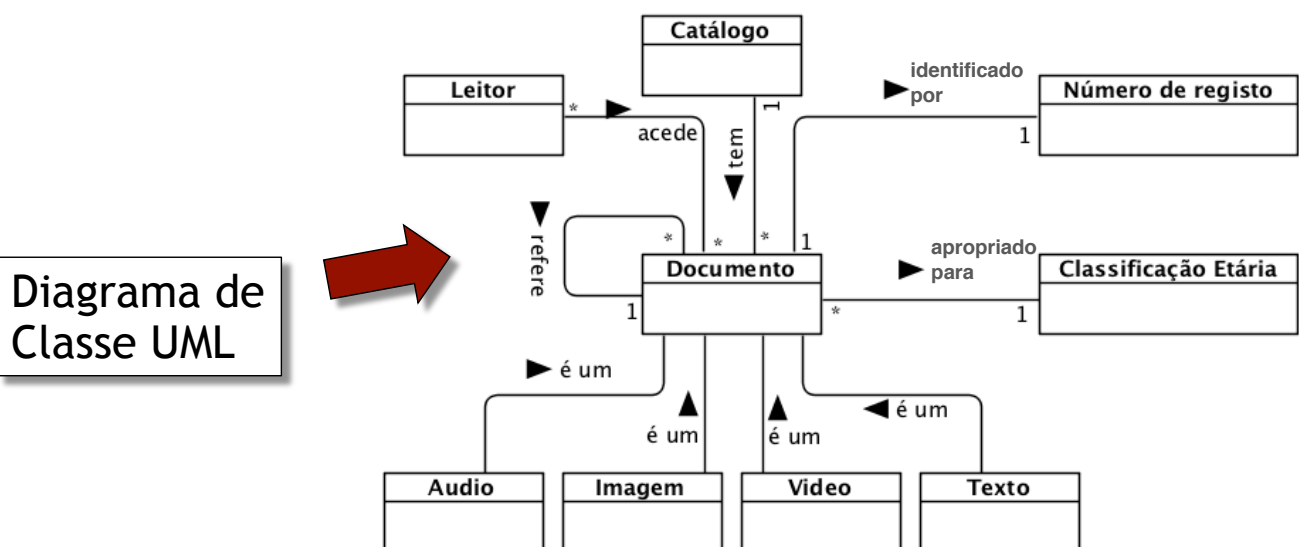
## Diagramas de Classe I - conceitos base

- A noção de classe é fundamental no paradigma OO
  - tipicamente uma classe representa uma abstração de uma entidade do mundo real.
- Cada classe descreve um conjunto de objectos com a mesma estrutura e comportamento:
  - Estrutura:
    - atributos
    - relações
  - Comportamento:
    - operações
- A organização do código em classes tem dois objectivos fundamentais:
  - facilitar a reutilização – através da reutilização de classes previamente desenvolvidas em novos sistemas;
  - facilitar a manutenção – o sistema deverá ser desenvolvido de forma a que a alteração de uma classe tenha o menor impacto possível no resto do sistema.



## Algumas notas sobre a notação

- Modelos de domínio são representados em **diagramas de classe** da UML
- Utilizam apenas um sub-conjunto da notação







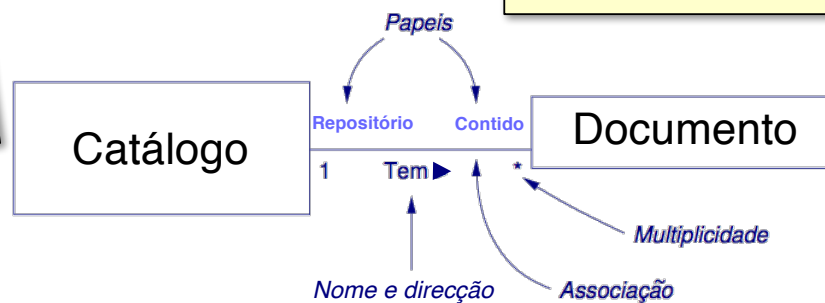
## Relações entre Entidades

- Três decorações possíveis:
  - nome** – descreve a natureza da relação (pode ter direcção)
  - papeis** – indica o papel que cada classe desempenha na relação definida pela associação (usualmente utilizado como alternativa ao nome)
  - multiplicidade** – quantos objectos participam na relação:
    - \* – zero ou mais objectos
    - $n$  –  $n$  objectos ( $n \geq 1$ )
    - $n..m$  – entre  $n$  e  $m$  objectos ( $n < m$ )

### Casos particulares:

- 1 – um objecto = objecto obrigatório
- 0..1 – zero ou um objectos = objecto opcional
- $n..*$  – mais de  $n$  objectos

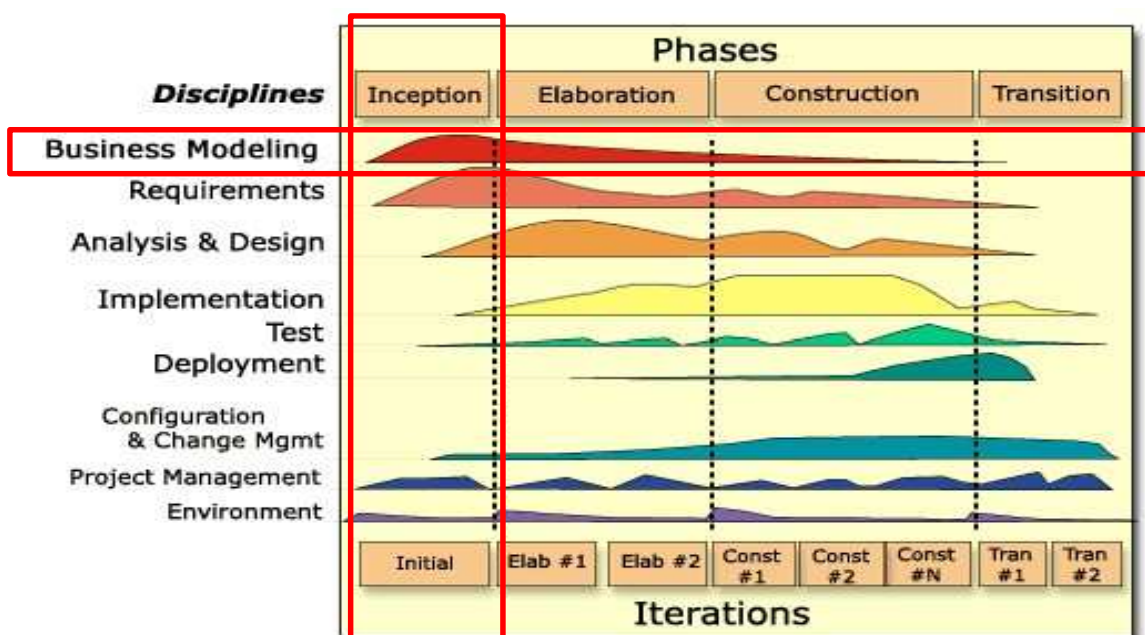
Um sub-conjunto da notação possível para os Diagramas de Classe UML!



v. 2017/18



## Próximos passos...



v. 2017/18





# Modelação do Domínio

## Sumário:

- O que é um Modelo de Domínio
- Identificação de Entidades
- Identificação de Relações
- Representação de Modelos de Domínio em UML - Diagramas de Classe
- Notação básica dos Diagramas de Classe (I)
  - Classes
  - Associações simples