



Lisbon School  
of Economics  
& Management  
Universidade de Lisboa



Carlos J. Costa

# **ANÁLISE E DESIGN DIAGRAMA DE CLASSES**



# Conceitos

**Diagrama de Classes**

**Classe**

**Associação**

**Generalização**

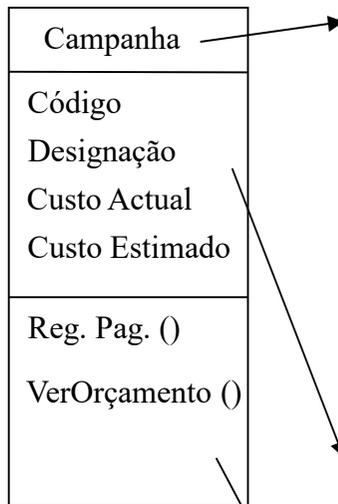
# Diagrama de Classes

- Diagramas que permitem especificar a estrutura estática de um sistema segundo a abordagem orientada por objectos.
- Utilizado para descrever o modelo de classes
- Elementos de um diagrama de classes:
  - Classes
  - Relações entre classes
    - Associações, Composições e Agregações
    - Generalizações

# Classe

## Identificação da Classe (Nome da Classe)

- Deve referir-se a objectos concretos mas abstrair da realidade
- Substantivos associados à descrição textual de um problema
- Escolher cuidadosamente os nomes
- Utilizar singular



## Atributos

- Valores que caracterizam os objectos de uma classe
- Tipos: Real, Inteiro, Texto, Booleano, Enumerado,...

## Operações

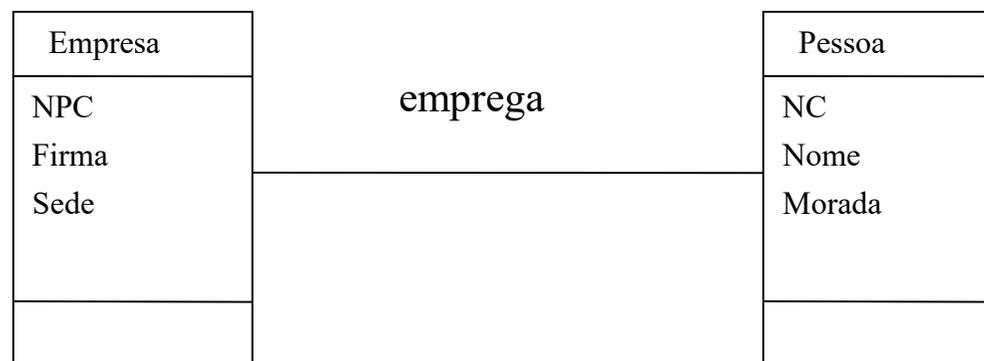
- Comportamentos da classe (serviço, método)

# Relações

- **Uma relação em UML estabelece a ligação entre elementos e é representada graficamente por um determinado tipo de linha. Na modelação orientada por objectos os três tipos de relações mais importantes são:**
  - **Associações,**
  - **Generalizações,**
  - **Dependências.**

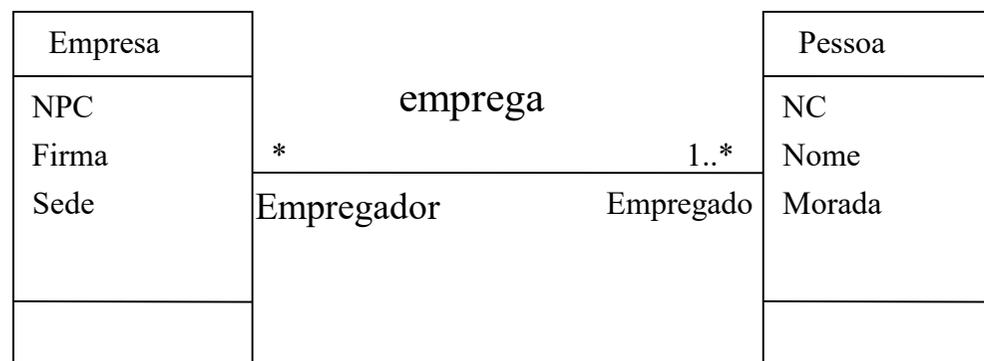
# Relações - Associações

- Uma relação de associação, ou simplesmente associação, é uma relação estrutural que especifica que objectos de uma classe estão ligados a objectos de outra.



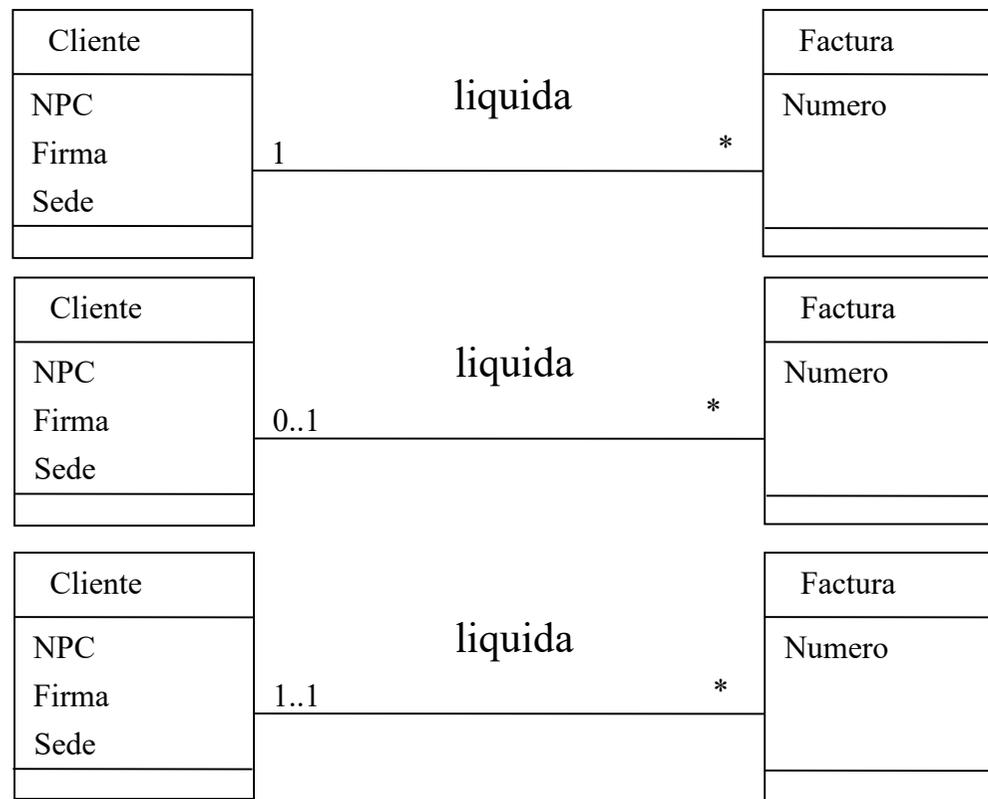
# Relações - Associações

- Uma associação é representada em UML por uma linha a cheio complementada por um conjunto de adornos que especificam diferentes informações, tais como:
  - O nome
  - O papel de cada participante na associação
  - A multiplicidade de cada participante na associação
  - O tipo de agregação



# Relações - Associações

- **Multiplicidade**



Ex.: venda a dinheiro ou factura/recibo

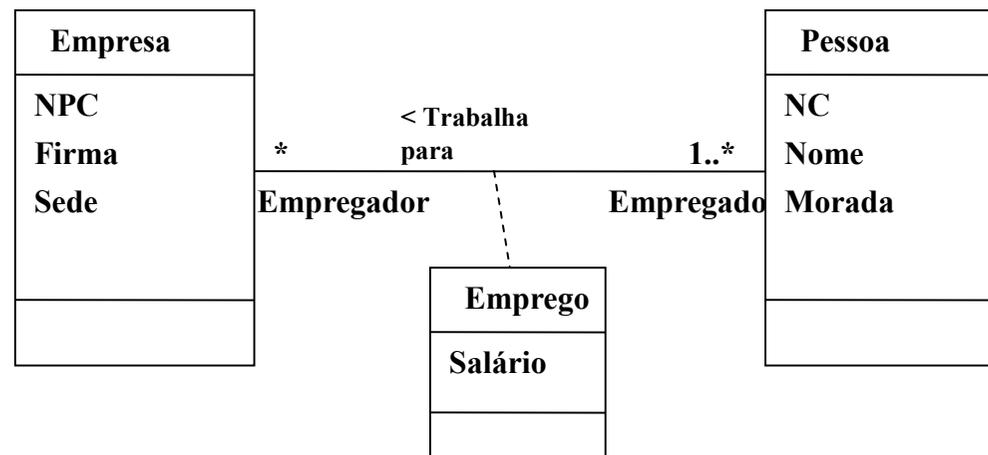
←

# Relações - Associações

- O papel é parte da associação, não da classe
- A multiplicidade traduz o número de instâncias de uma classe que se podem relacionar (através da associação) com uma única instância da(s) outra(s) classe(s) participante(s). Pode-se especificar em UML qualquer tipo de multiplicidade. Por exemplo,
  - multiplicidade muitos (\*),
  - um ou mais (1..\*),
  - exactamente um (1),
  - zero ou um (0..1),
  - um determinado número (e.g., 3),
  - uma determinada gama (e.g., 2..6),
  - multiplicidade mais complexa especificada através de listas (e.g., 0..3, 5..7, 10..\* para representar “qualquer número de objectos excepto 4, 8 ou 9”).

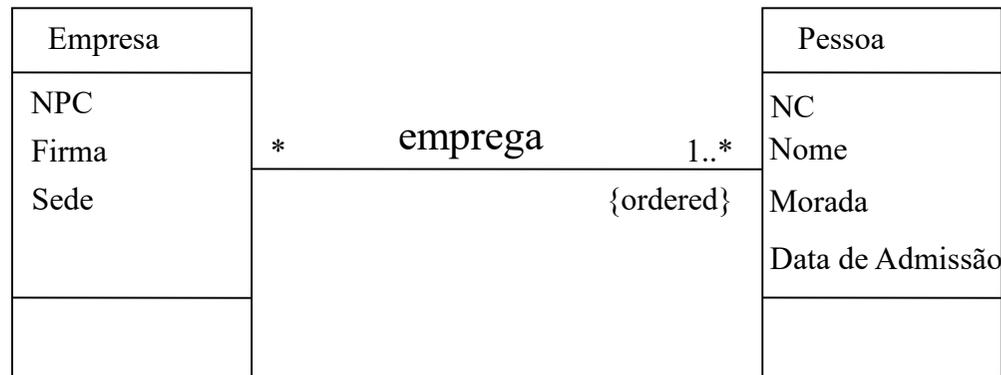
# Relações - Associações

- A associação pode ser suficientemente complexa para ela própria ser traduzida por uma classe (classe associativa)



# Relações - Associações

- Associações Ordenadas
  - {ordered} indica que os objectos da classe *Pessoa* estão ordenados segundo uma determinada ordem, e.g. *Ordered by increasing time*
  - Por omissão as associações não têm ordem

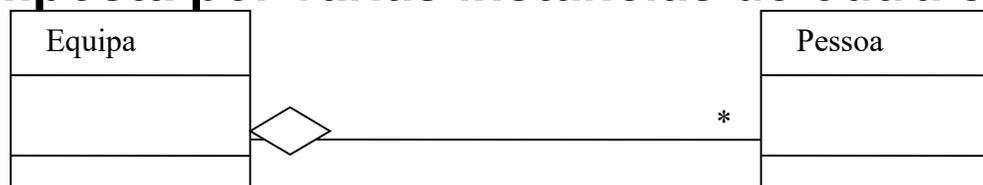


# Relações - Associações

- **Associações OR**
  - **Especifica que os objectos de uma classe podem participar no máximo numa das associações num determinado instante**

# Relações - Associações

- A associação entre classes sem agregação reflecte que ambas as classes se encontram no mesmo nível conceptual (ex. Uma Pessoa compra um Produto)
- Relação de associação com agregação
  - Traduz que existe uma relação do tipo “*is-part-of*”, “*has-a*”, “*consiste em...*”, “*contém...*” ou “*é parte de...*”
  - Uma instância de determinada classe possui ou é composta por várias instâncias de outra classe.



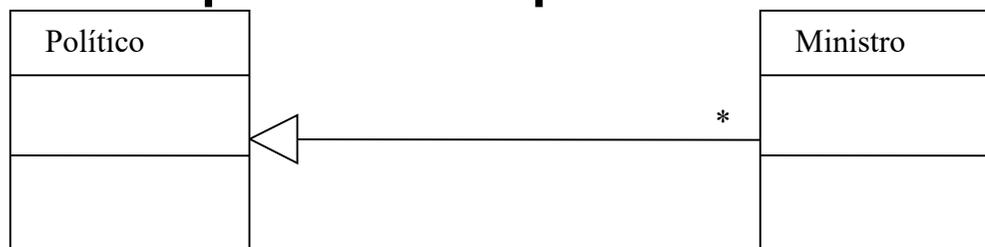
# Relações - Associações

- **Composição, ou agregação composta** - é uma variante à agregação simples, em que é adicionada a seguinte semântica:
  - forte pertença do “todo” em relação à “parte”.
  - tempo de vida delimitado (as “partes” não podem existir sem o “todo”).
  - o “todo” é responsável pela disposição das suas “partes”, ou seja, “o todo” é responsável pela criação e destruição das suas “partes”.



# Relações - Generalização

- Uma relação entre um elemento geral (e.g., superclasse, super-use case, super-actor) e um elemento mais específico (e.g., subclasse, sub-use case, sub-actor).
- Geralmente conhecida como uma relação do tipo “*is-a*”, “*is-a-kind-of*” ou “*é*”:
  - “Um gestor é um empregado”
  - “Um ministro é um membro do governo”
  - “Um deputado é um político”



# Relações - Dependência

- **Uma relação de dependência, ou simplesmente dependência, indica que a alteração na especificação de um elemento pode afectar outro elemento que a usa, mas não necessariamente o oposto.**

# Bibliografia

- Bennet, S. McRobb, S & Farmer, R., *Object Oriented Systems Analysis and Design using UML*, MacGarw-Hill, 1999.
- Booch, G., Rumbaugh, J. & Jacobson, I, *The Unified Modeling Language User Guide*. Addison Wesley, 1999 (tradução portuguesa brasileira \_\_\_\_\_; *UML Guia do Usuário*; Campus, 2000).
- Costa, C. *Desenvolvimento para Web*, ITML Press, 2007
- Nunes, M & O'Neill, H. *Fundamental de UML*, FCA, 2001
- Silva, A & Videira, C., *UML, Metodologias e Ferramentas CASE*, Edições Centro Atlântico, 2001
- Terry, Q. *Visual Modeling With Rational Rose 2000 and UML*, Addison-Wesley. 2000.
- *Oxford Dictionary of Computing*, Oxford University Press.