



iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA



emprego
digital

Módulo 5: Bases de Dados

Aula 2

Classes Associativas, Agregações, Composições e Generalizações



Recapitulando - Associações

Em UML existem os seguintes tipos de relações, que expressam diferentes semânticas de ligação entre objectos:

- Associação

Tem dois casos especiais:

- Agregação
 - Composição
- Generalização
- Relação de dependência

Classe Associativa

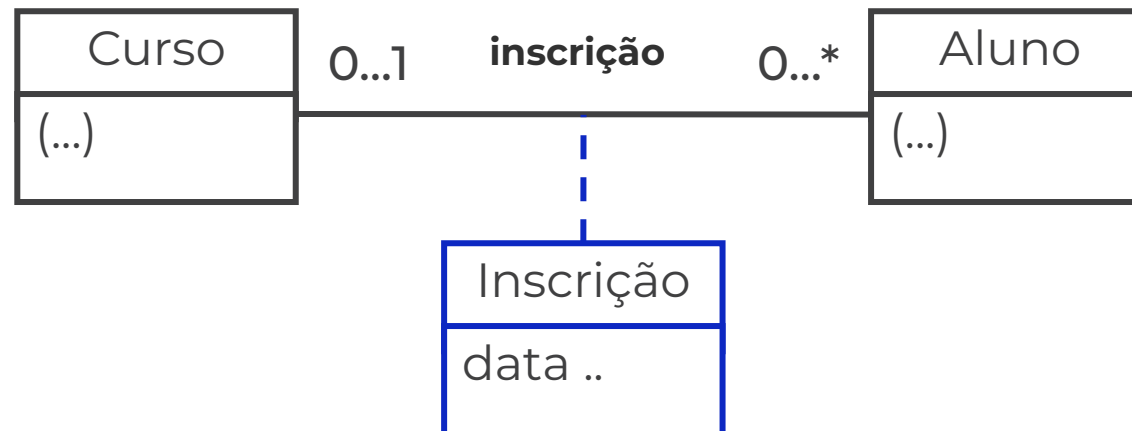
Por vezes, queremos acrescentar informação adicional a uma relação de associação...



Como fazer para acrescentar **o ano de inscrição**, por exemplo?

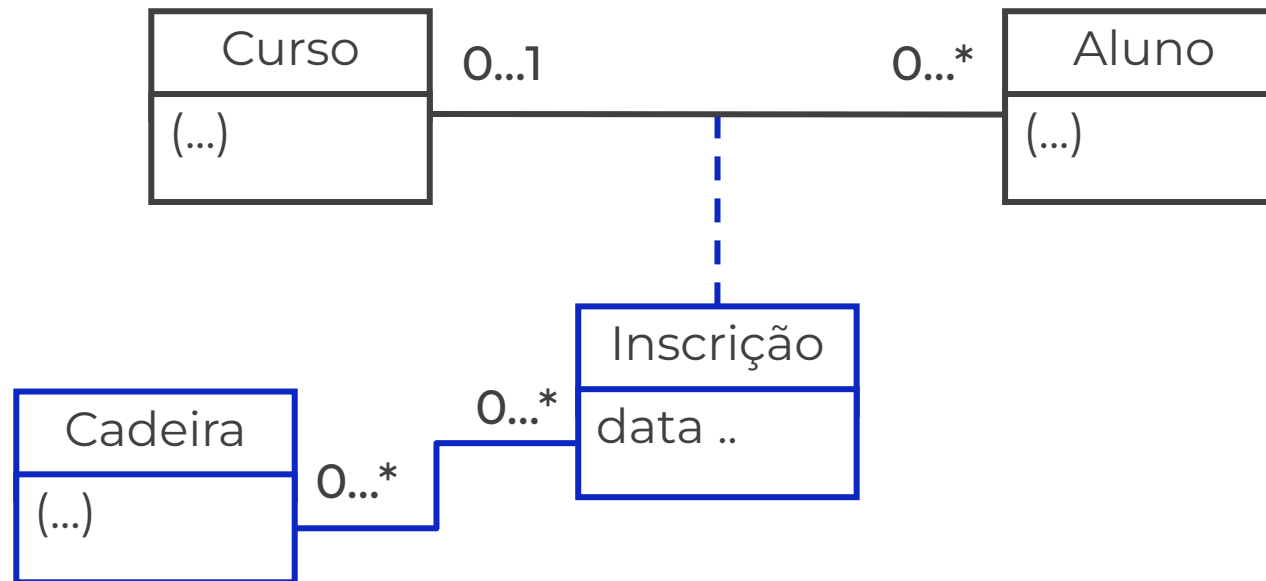
Classe Associativa

Usamos classes associativas quando é necessário colocar atributos na associação



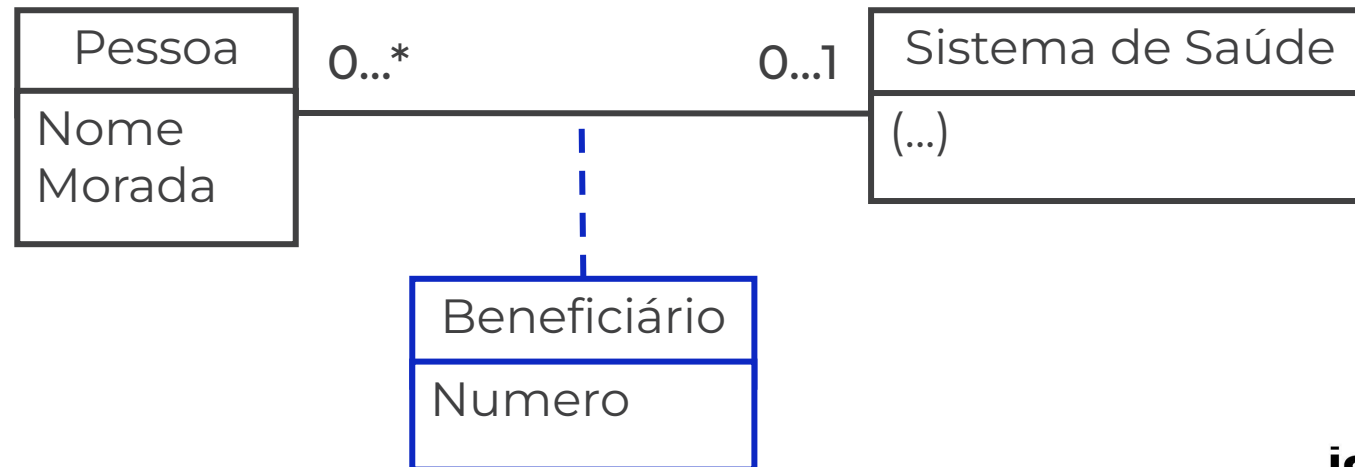
Classe Associativa

Também podemos usar classes associativas para **associar uma ligação entre objetos a outros de uma terceira classe**



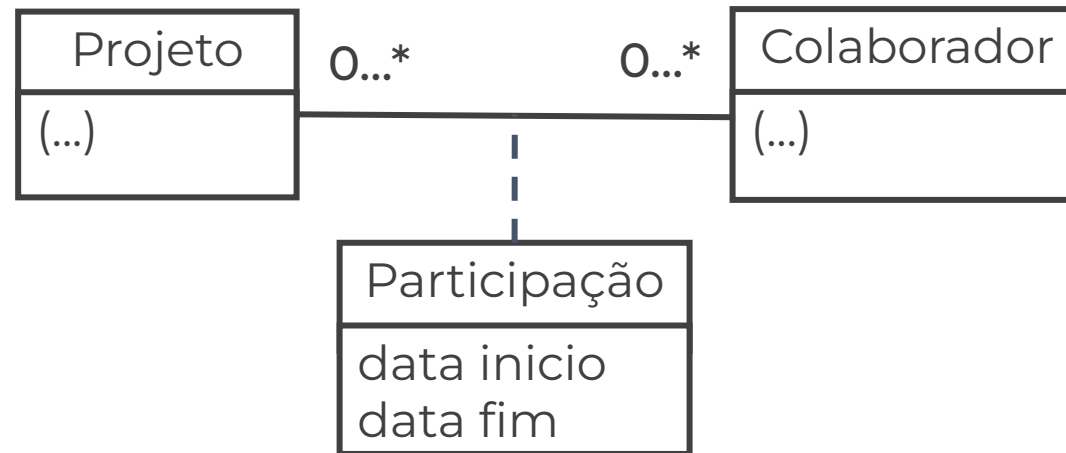
Classe Associativa

As classes associativas são mais frequentemente necessárias nas associações muitos para muitos, mas também podem fazer sentido em associações de outras cardinalidades:



Classe Associativa vs Duas Associações

Mas atenção! O facto de utilizarmos uma classe associativa para adicionar informação à associação, **não altera** a regra do muitos para muitos.



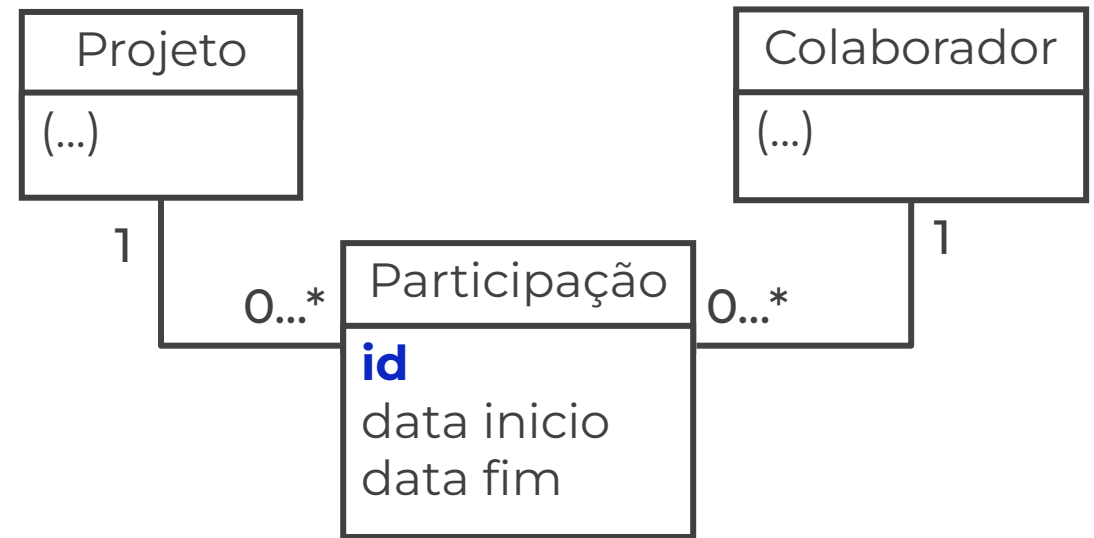
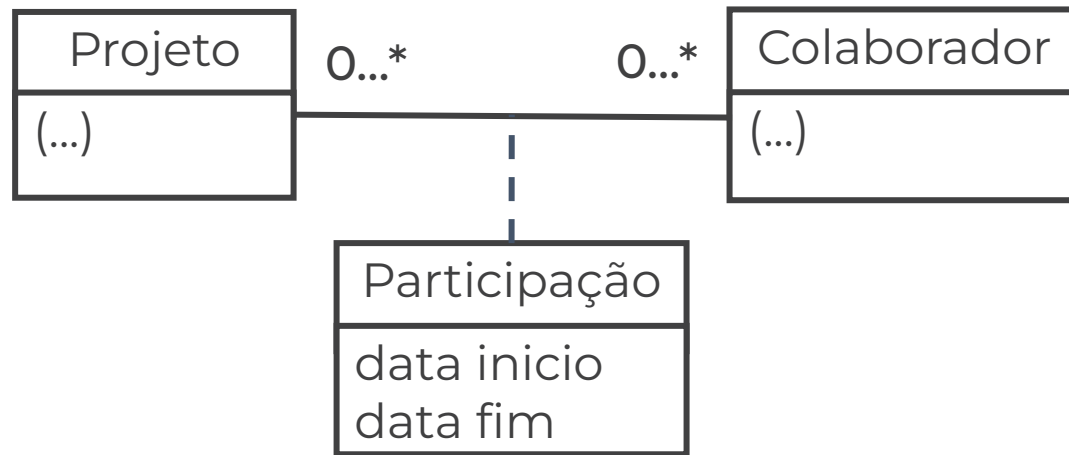
Continuamos a poder ter apenas uma associação entre o mesmo projeto e o mesmo colaborador, **mesmo com datas de participação diferentes.**

E se quisermos ter várias participações do mesmo projeto com o mesmo colaborador?

Nesse caso, deveremos usar o **método de duas associações** em vez de uma classe associativa.

Duas Associações

Se quisermos ter mais do que um registo ou entrada associativa, usamos duas associações em vez de uma classe associativa



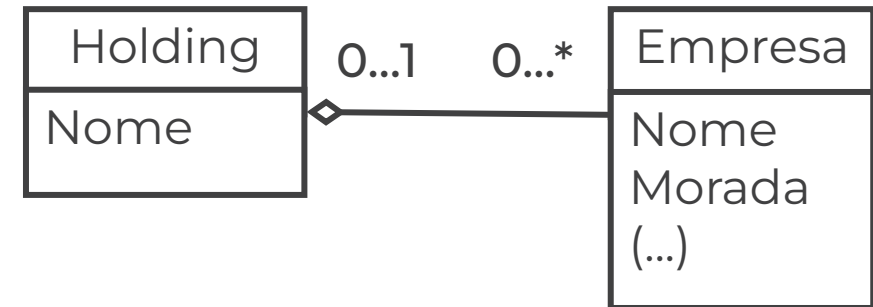
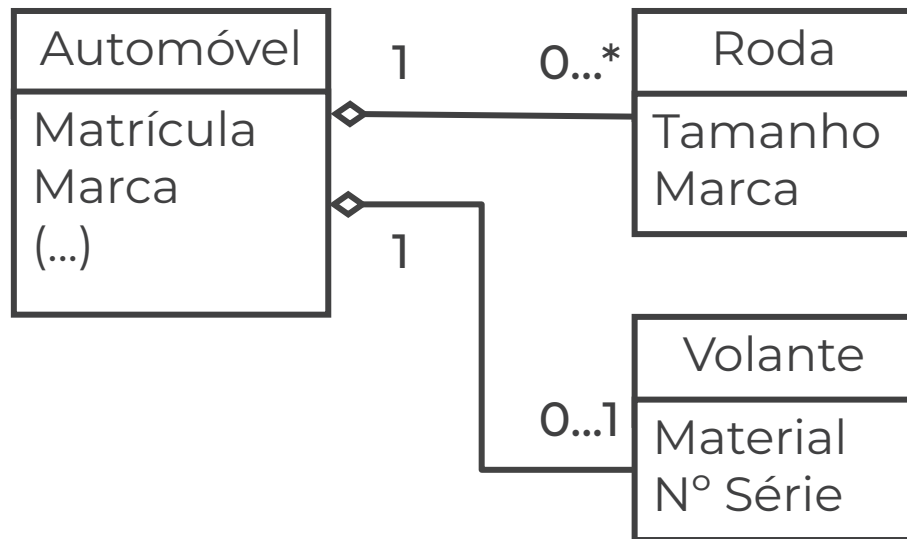
Agregações

As agregações são representadas graficamente por uma linha adornada com losango branco no extremo correspondente ao todo.



Agregações

As Agregações são associações que se utilizam quando se pretende representar a noção de Todo/Parte (um todo constituído por partes).



Composições

As composições são um caso especial de agregações: representam na mesma uma situação de todo e parte, **mas onde a parte apenas faz existe no contexto do todo, não fazendo qualquer sentido isoladamente.**



Composições: Identificação

Os componentes dependem do todo para efeitos de identificação: Os objectos componente **incluem no seu mecanismo de identificação o mecanismo de identificação do objecto composto (id é chave composta)**. Exemplo:

- Os nomes dos departamentos podem repetir-se entre empresas – se não juntarmos o nome da empresa não conseguiremos distinguir certos departamentos que possuam idêntica designação em empresas distintas
- Uma StatementLine só se pode identificar inequivocamente se também mencionarmos a Account a que diz respeito

Composições

Accounts diferentes mas *StatementLines* iguais. **Só é possível distinguir as duas statement lines pela conta associada.**

Account Info - 11-12-2020
Account 1234567890986 EUR ORDEM1 DraftAccount
Start Date: 03-11-2020
End Date: 03-11-2020

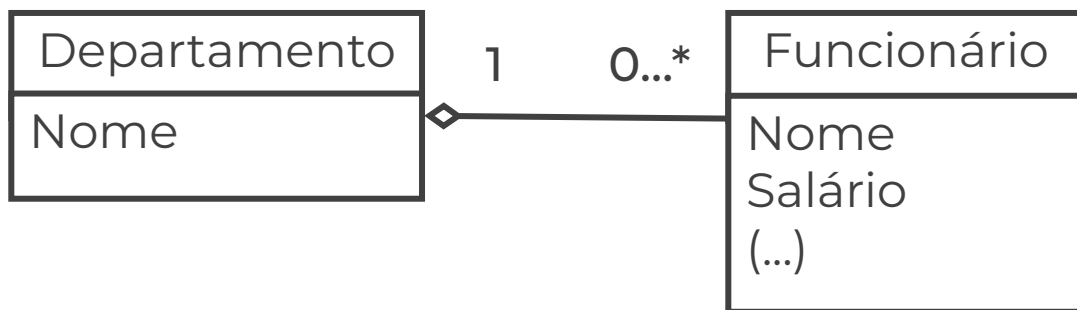
Date	Value Date	Description	Draft	Credit
31-08-2013	31-08-2013	SUMMARY	-3100,00	3200,00
30-09-2013	30-09-2013	SUMMARY	-3100,00	3200,00

Account Info - 11-12-2020
Account 1234567890987 EUR ORDEM DraftAccount
Start Date: 04-02-2014
End Date: 03-02-2014

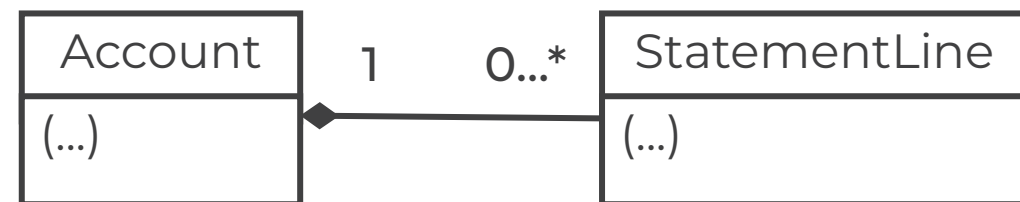
Date	Value Date	Description	Draft	Credit
31-08-2013	31-08-2013	SUMMARY	-3100,00	3200,00
31-12-2013	31-12-2013	SUMMARY	-1875,98	0,00

Agregações vs Composições

Apesar da obrigatoriedade existir em ambas as associações seguintes, são situações com semânticas distintas:



O Funcionário existe por si próprio e **não necessita** estar associado a um departamento para poder ser referenciado.



Uma StatementLine **só pode ser referida** se for indicada a Account respectiva.

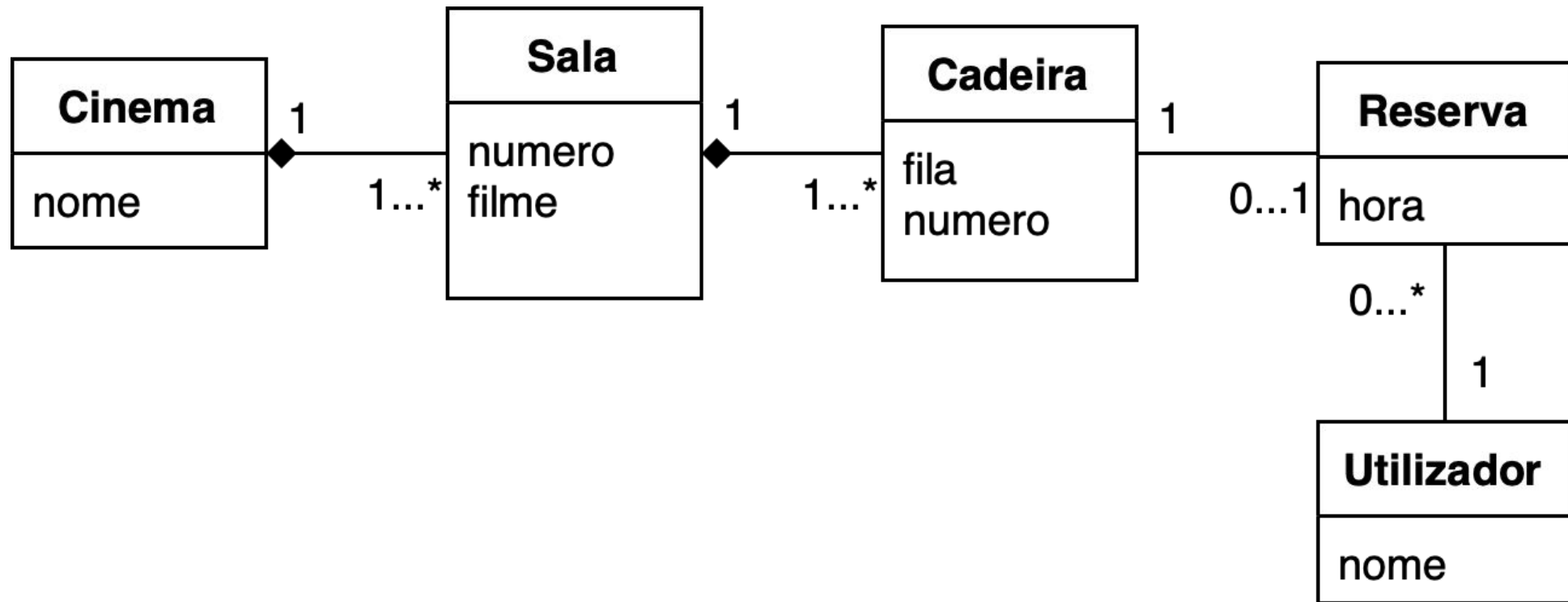
Exercício 1

Pretende desenvolver-se um sistema para gerir o *FrontTicket*. O descritivo abaixo descreve as necessidades do sistema.

“O FrontTicket é uma plataforma de reserva de bilhetes para vários cinemas. Pretendemos implementar um sistema que suporte a funcionalidade da nossa plataforma. Ao efetuar uma reserva, o utilizador deverá indicar a hora da sessão, o Cinema, a Sala e a cadeira associada.”

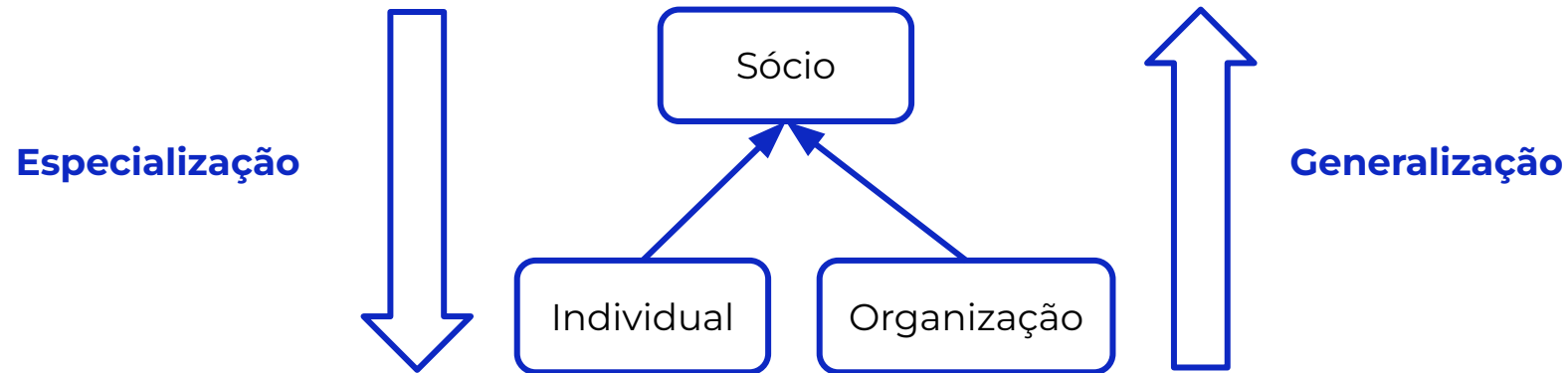
Elabore o diagrama de classes da base de dados necessária para suportar o sistema descrito.

Exercício 1 - Resolução



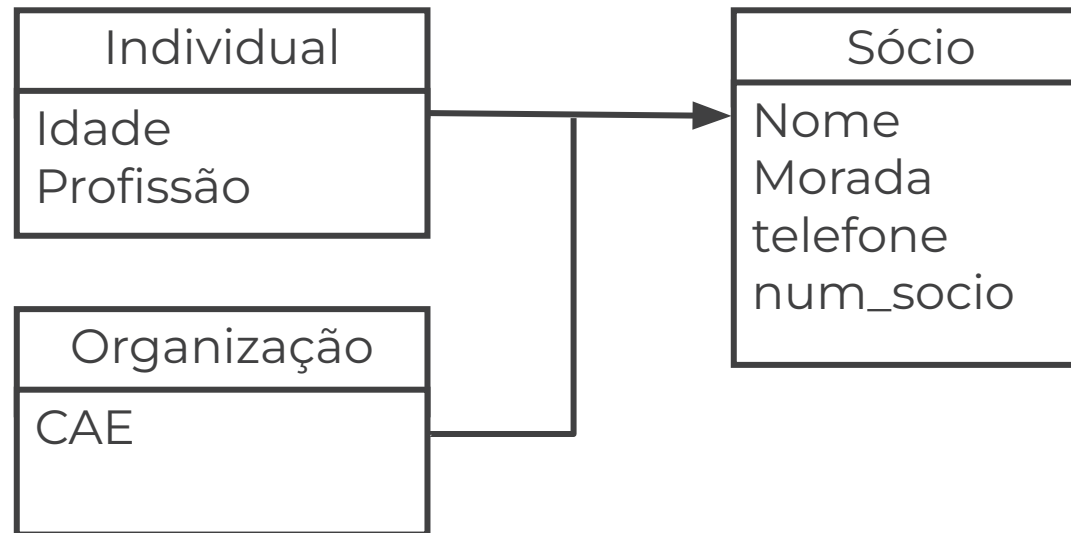
Generalização

A generalização é uma relação (“um para um”) que permite representar a noção de generalização/especialização nas classes.



Generalização

Parecido às hierarquias no Java, a generalização permite a uma classe herdar propriedades **(subclasse - superclasse)**



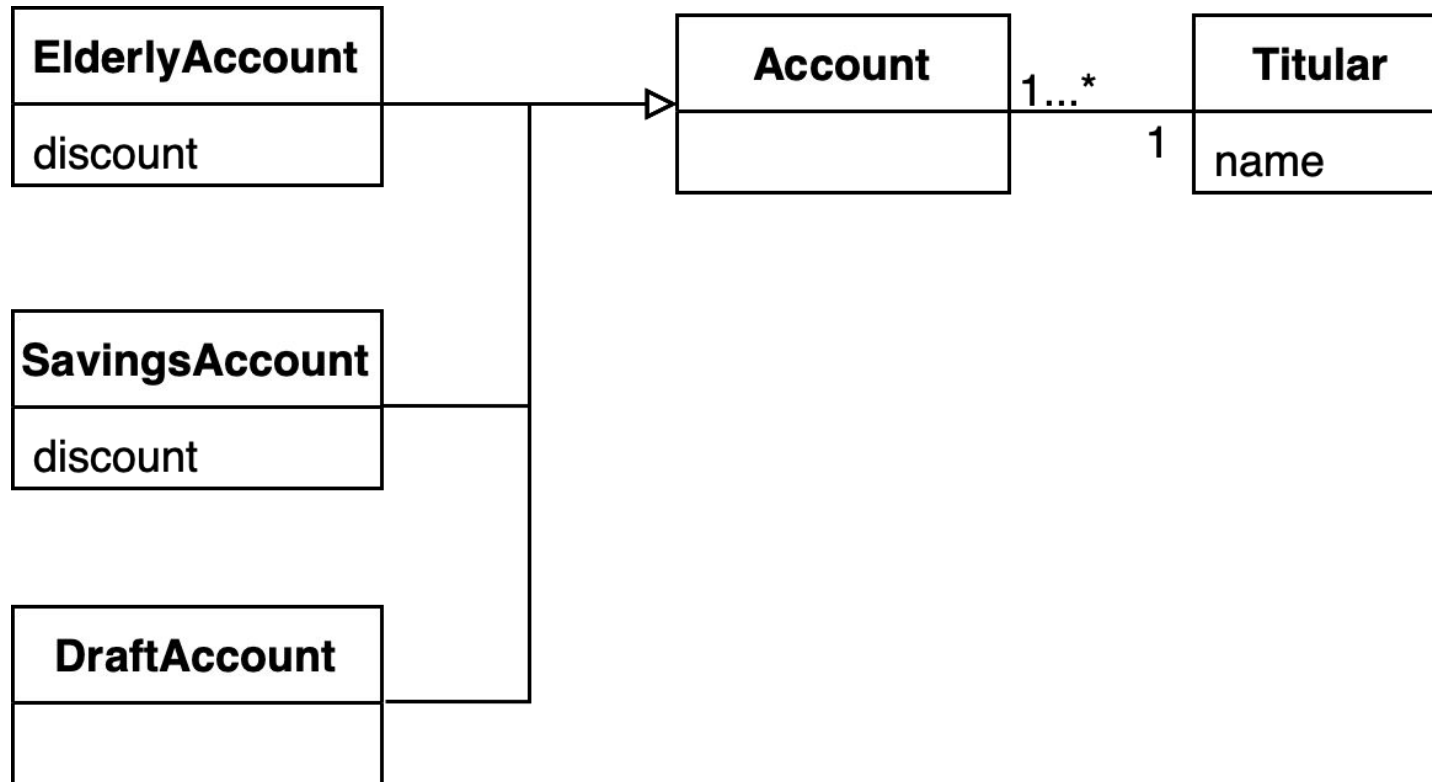
Exercício 2

Pretende desenvolver-se um sistema para gerir o *MeuBanco*. O descritivo abaixo descreve as necessidades do sistema.

“Pretende-se um sistema que permita gerir uma parte do nosso banco. Pretende-se representar os vários tipos de conta no nosso sistema: ElderlyAccount, SavingsAccount e DraftAccount. As duas primeiras têm um discount associado, com o tipo de desconto em percentagem. O sistema deve também ter informação do titular da conta (uma conta apenas pode ter um titular).”

Elabore o diagrama de classes da base de dados necessária para suportar o sistema descrito.

Exercício 2 - Resolução

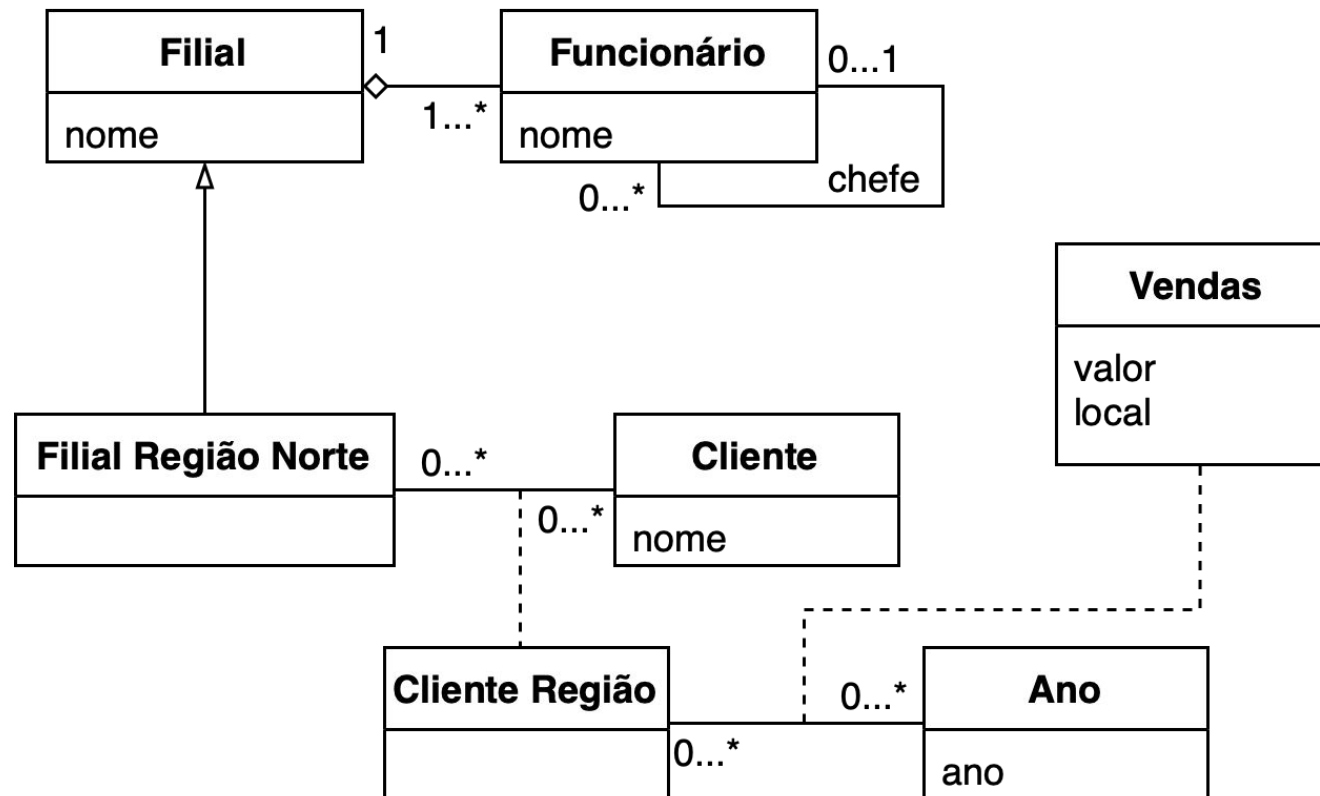


Exercício 3

Considere o diagrama de classes que se apresenta de seguida e tente validar se as seguintes afirmações são verdadeiras ou falsas:

- Um funcionário pode ser chefe dele próprio?
- Para um determinado ano, podem ser registados vários valores de vendas de um mesmo cliente?
- Um funcionário pode ter como chefe um funcionário de outra filial?

Exercício 3 - Diagrama de Classes



Exercício 3 - Resolução

- Um funcionário pode ser chefe dele próprio? **Verdadeiro**
- Para um determinado ano, podem ser registados vários valores de vendas de um mesmo cliente? **Verdadeiro, desde que as regiões sejam diferentes.**
- Um funcionário pode ter como chefe um funcionário de outra filial? **Verdadeiro**

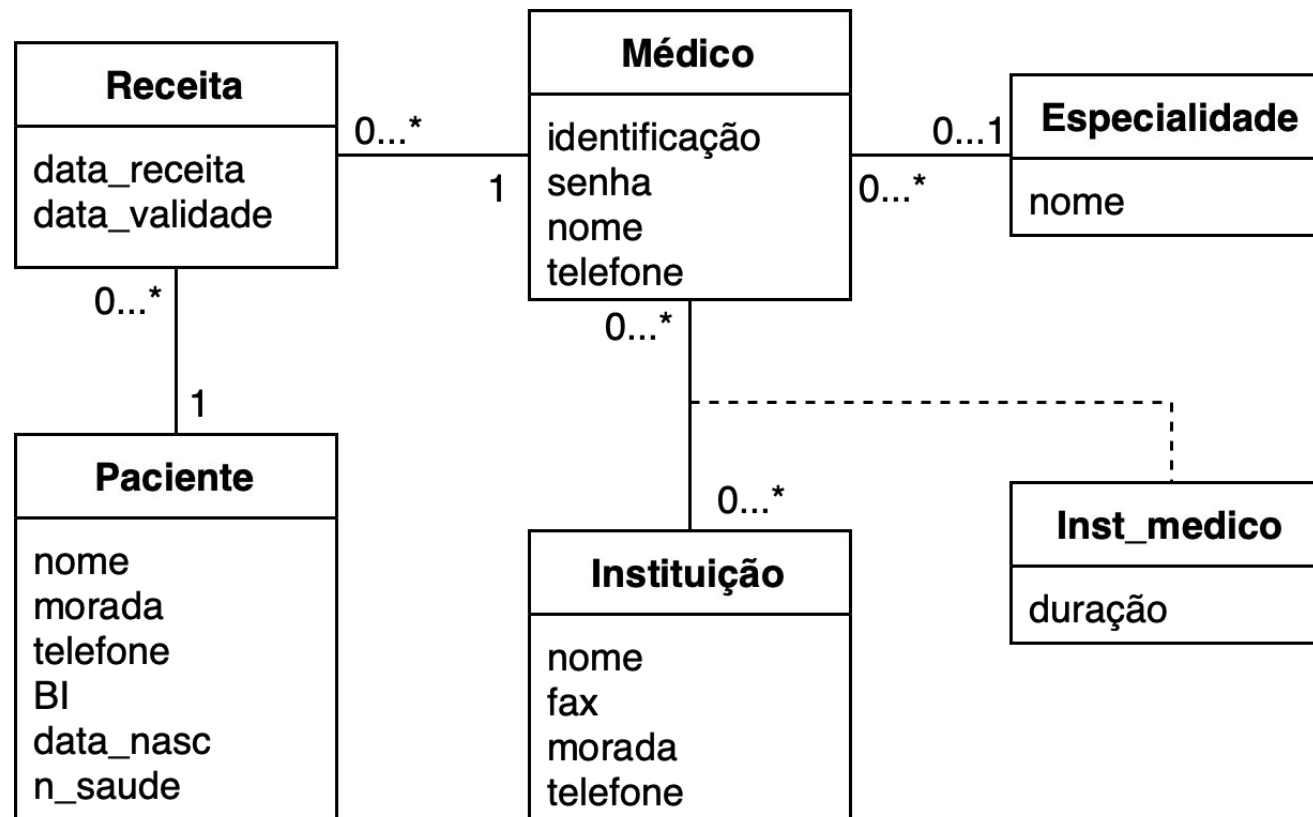
Exercício 4

Pretende desenvolver-se um sistema simples para o Ministério da Saúde de gestão de receitas. O descritivo abaixo descreve as necessidades do sistema.

“O sistema deve permitir aos médicos a introdução das receitas dos pacientes. O médico é o único responsável pela inserção da receita no sistema, e, para tal deverá identificar-se através de uma identificação e de uma senha, previamente atribuídas. A receita será inserida por um médico e refere-se apenas a um paciente previamente inserido no sistema. Sobre cada médico é necessário saber qual o seu nome, número de telefone de contacto, qual a sua principal especialidade, as várias instituições onde este trabalha ou trabalhou e quanto tempo aconteceu. A data de validade da receita será definida pelo médico e também existe a data da receita que consiste no dia em que a receita foi passada pelo médico. O sistema armazena alguma informação sobre os pacientes, tais como o seu nome, número de bilhete de identidade, data de nascimento e a sua identificação (número) no Sistema de Saúde.”

Elabore o diagrama de classes da base de dados necessária para suportar o sistema.

Exercício 4 - Resolução



Exercício 5

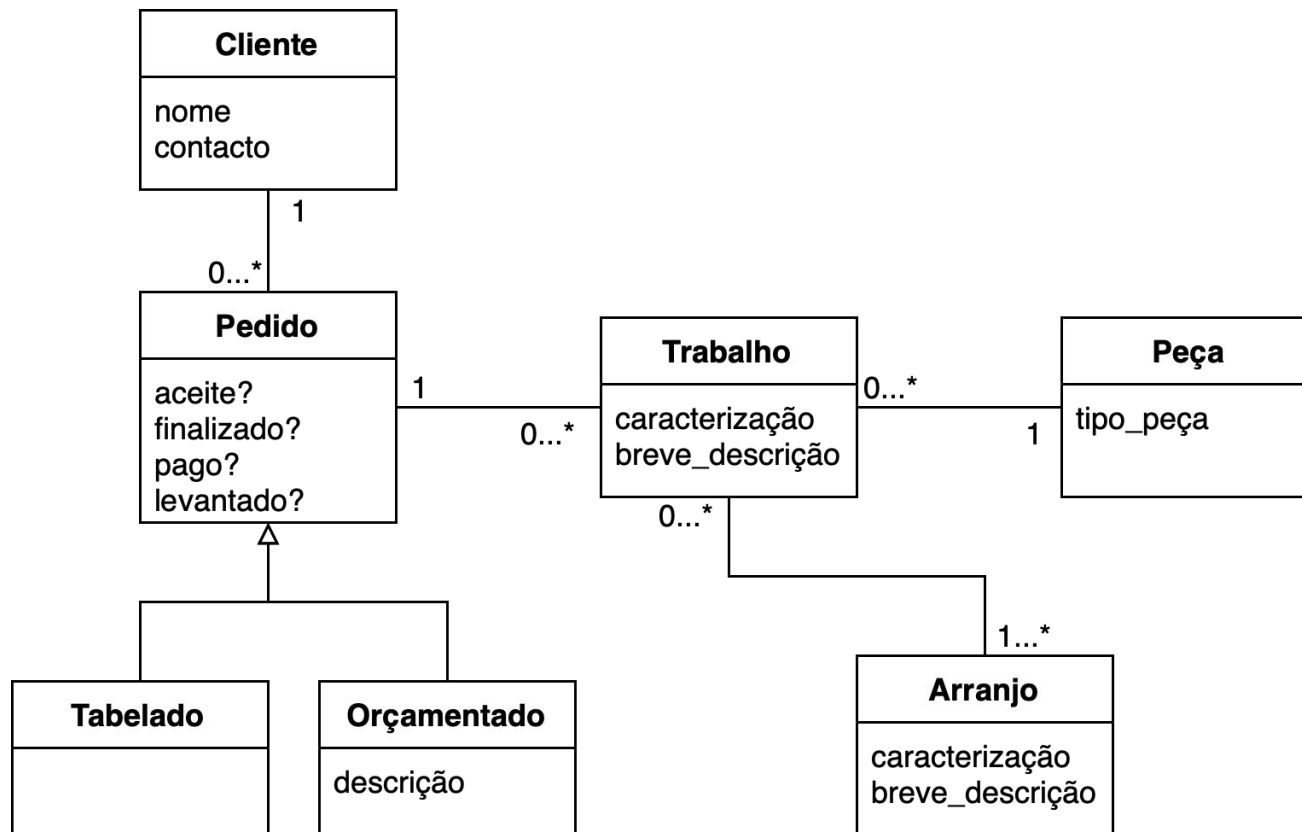
Pretende desenvolver-se um sistema simples para uma cadeia de retrosarias para a caracterização dos pedidos. O descritivo abaixo descreve as necessidades do sistema.

“Os pedidos solicitados pelos clientes resultam de combinações de duas variáveis: peça e arranjo. Exemplos de tipos de peças são: Camisa, Calças, Casaco, etc. Exemplos de tipo de arranjos são: Coser, Bainhas, Colocar Fechos, etc. Cada solicitação de execução de um trabalho é descrita através da especificação do tipo de arranjo e da peça de vestuário, sendo necessário fazer a caracterização da peça (cor, tamanho, etc.) e ainda, em alguns casos, uma breve descrição sobre o arranjo a executar. Cada pedido de execução pode conter um ou vários arranjos.

Os vários pedidos arranjos de peças têm o custo tabelado, podendo no entanto surgir situações não habituais em que é necessário elaborar um orçamento. Para o caso do orçamento é necessário introduzir uma descrição detalhada sobre o arranjo a executar. O trabalho apenas é iniciado depois de, telefonicamente, o cliente aceitar o orçamento. O sistema deve avisar os clientes através de uma SMS, de que o pedido está finalizado, sendo necessário garantir que o cliente possui um contacto telefónico associado. Após receber a mensagem o cliente pode levantar as peças”.

Elabore o diagrama de classes da base de dados necessária para suportar o sistema descrito.

Exercício 5 - Resolução



O futuro profissional começa aqui

iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA



emprego
digital



UPskill