

# Extensões do Modelo Entidade-Relacionamento

Profa. Maria Camila Nardini Barioni

[camila.barioni@ufu.br](mailto:camila.barioni@ufu.br)

Bloco B - sala 1B137

1º semestre de 2020

# Resolução: Exercício da Aula 3

O objetivo deste estudo de caso é construir um modelo ER para o banco de dados de uma empresa de locação de veículos. A empresa em questão aluga automóveis e camionetas. Para acelerar o atendimento, é importante conhecer os dados de **clientes** que já tenham usado a locadora no passado. Para cada cliente é necessário conhecer seu **nome**, **sexo**, **data de nascimento**, **endereço**, pelo menos um **telefone** de contato e **CPF**. Os clientes são identificados por um **código** interno à locadora.

# Resolução: Locadora Veículos

Cada **veículo** disponível para locação é identificado pela sua **placa**. Além disso, é necessário conhecer o **número do chassi**, o **número do motor**, o **tipo de veículo** e a **cor** de cada veículo.

O sistema em computador deverá registrar:

- **os veículos disponíveis** para locação na data corrente;
- **as reservas**, com previsão de que veículos estarão disponíveis em uma data futura. Cada **reserva** é identificada por um **código** único. Para cada reserva é necessário saber qual o **cliente efetuou a reserva**, a **data prevista para a retirada**, a **data prevista para a devolução** e o **local de retirada** do veículo;

# Resolução: Locadora Veículos

- **os veículos presentemente alugados**, a **data prevista de entrega** e **qual cliente efetuou cada locação**.

É importante notar que, depois que um veículo é devolvido não é mais necessário armazenar os dados referentes à locação do mesmo.

Os veículos são classificados por uma **tabela de tipos**. Por exemplo, P3 corresponde a automóveis pequenos, de quatro portas e com ar-condicionado e G4 a grandes automóveis de luxo. As **reservas não são feitas** para uma marca ou modelo de veículo, mas para um tipo de veículo. Portanto, cada reserva deve estar **vinculada** a um tipo de veículo específico.

# Resolução: Locadora Veículos

Para cada **tipo de veículo** os clientes desejam saber o **tamanho** (classificado como pequeno, médio e grande), o **número de passageiros**, o **número de portas**, o **valor da diária de locação**, o valor do quilometro rodado, bem como se possui os seguintes **acessórios**: ar-condicionado, rádio, CD, MP3, direção hidráulica e câmbio automático.

Para cada tipo de veículo, há um determinado **número de horas** necessário para limpeza e revisão de entrega, entre uma reserva e outra.

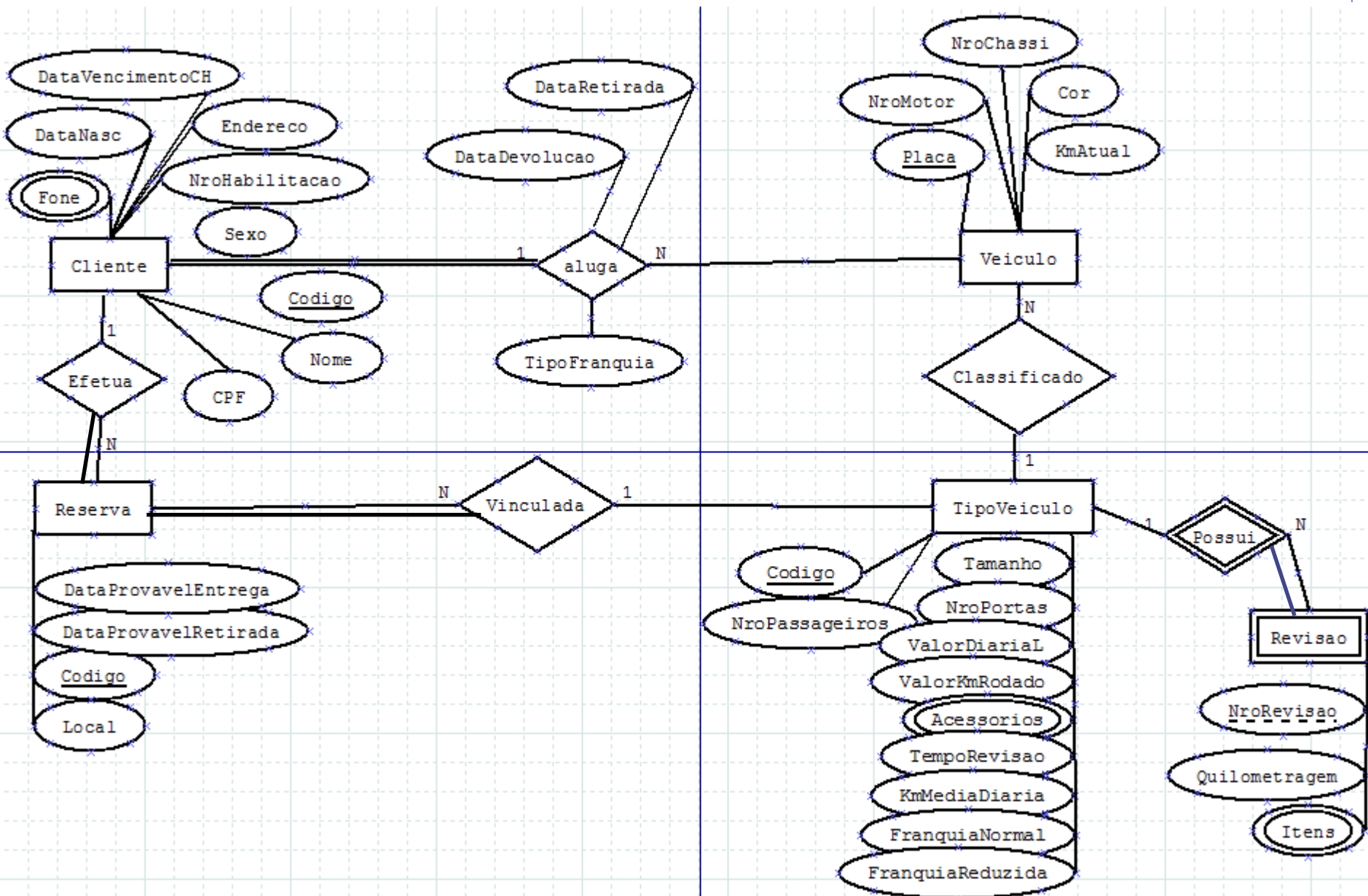
# Resolução: Locadora Veículos

Além disso, o sistema deve programar as **revisões** dos veículos, impedindo que sejam reservados quando há revisões pendentes. Esta programação é feita com base em um conjunto de parâmetros que são a **quilometragem atual** do veículo, a **quilometragem média** diária de um veículo do tipo, bem como em uma **tabela de revisões** programadas na qual **para cada tipo de veículo** há um conjunto de **itens** que devem ser revisados e a **quilometragem na qual ela deve acontecer**. Por exemplo, a primeira revisão dos veículos do tipo P3 acontece com 15.000 km rodados e dos veículos do tipo G4 acontece com 10.000 km.

# Resolução: Locadora Veículos

A seguradora que segura os veículos exige que, **para cada veículo alugado**, seja mantida a **identificação do motorista**, o **número de sua habilitação** e **data de vencimento da mesma**. A habilitação não pode vencer dentro do prazo da locação. Além disso, o cliente que aluga um veículo pode **optar por franquias** reduzida ou normal sendo que o valor de cada tipo de franquias varia de acordo com o tipo de veículo.

# Resolução: Locadora Veículos





# Modelo Entidade Relacionamento Estendido

## ◆ Características

- introduz semântica adicional ao modelo ER
- utilizado na modelagem de aplicações mais complexas, tais como CAD/CAM, BD gráficos, BD geográficos

## ◆ Conceitos

- subclasse, superclasse, hierarquia de herança
- generalização, especialização, e restrições
- agregação

# Subclasse/Superclasse

## ◆ Subclasse

- agrupamento das entidades de um subgrupo do tipo-entidade

## ◆ Exemplo

- superclasse: tipo-entidade empregado
- subclasses: secretário, engenheiro, técnico

cada entidade que é membro de qualquer uma das subclasses também é um empregado

# Herança

## ◆ de atributos

- atributos da superclasse são herdados pelas subclasses

## ◆ de relacionamentos

- instâncias de relacionamento da superclasse são herdadas pelas entidades das subclasses

## ◆ Observação

- qualquer entidade membro de uma subclasse deve ser também membro da superclasse
- qualquer entidade membro da superclasse pode ser opcionalmente incluída como membro de qualquer número de subclasses

# Generalização/Especialização

## ◆ Especialização

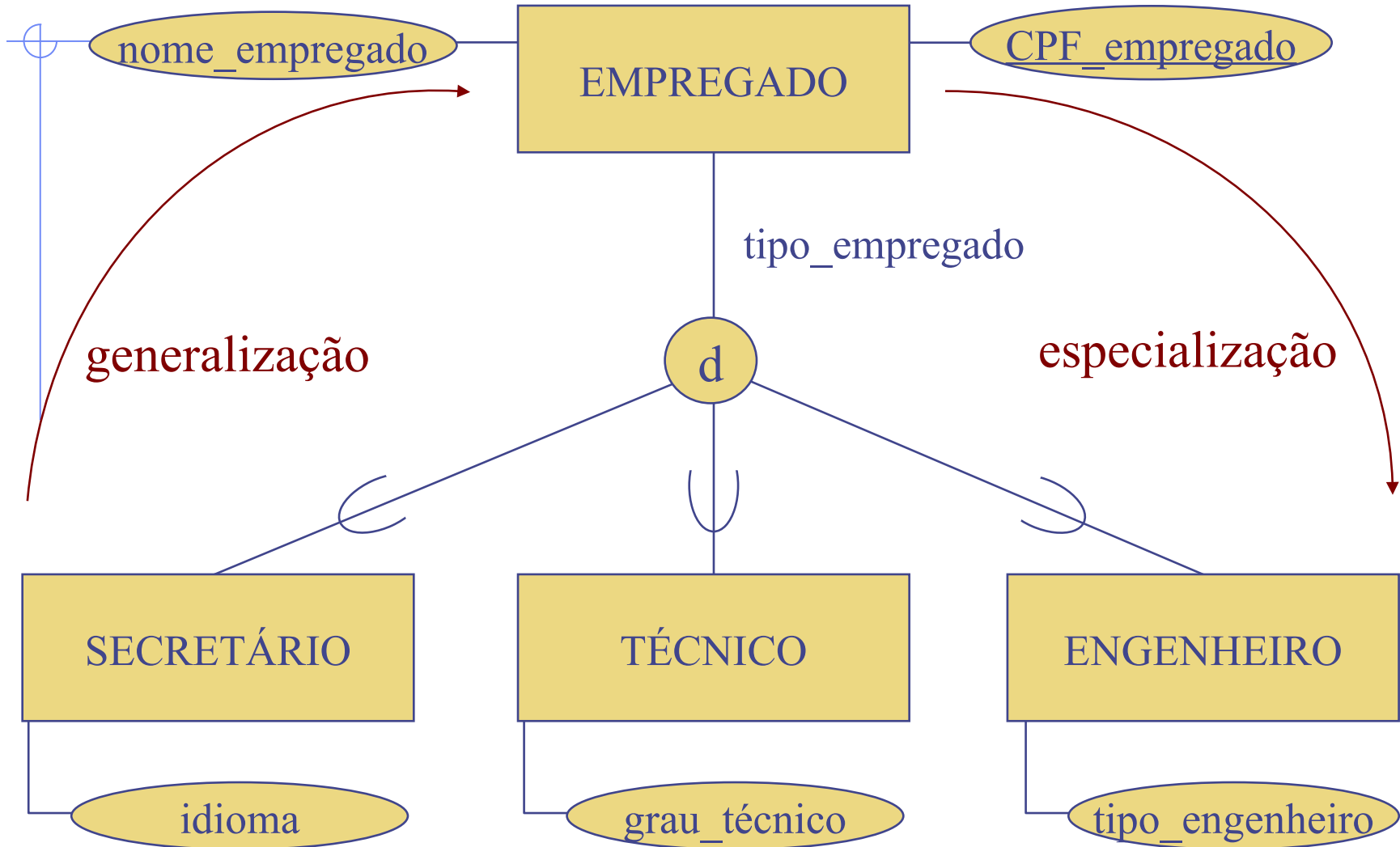
- resultado da separação de um tipo-entidade de nível mais alto (superclasse), formando vários tipos-entidade de nível mais baixo (subclasse)
- passos:
  - ◆ define-se um conjunto de subclasses de um tipo-entidade
  - ◆ associa-se atributos adicionais específicos às subclasses
  - ◆ estabelece-se tipos-relacionamento adicionais específicos às subclasses, caso necessário

# Generalização/Especialização

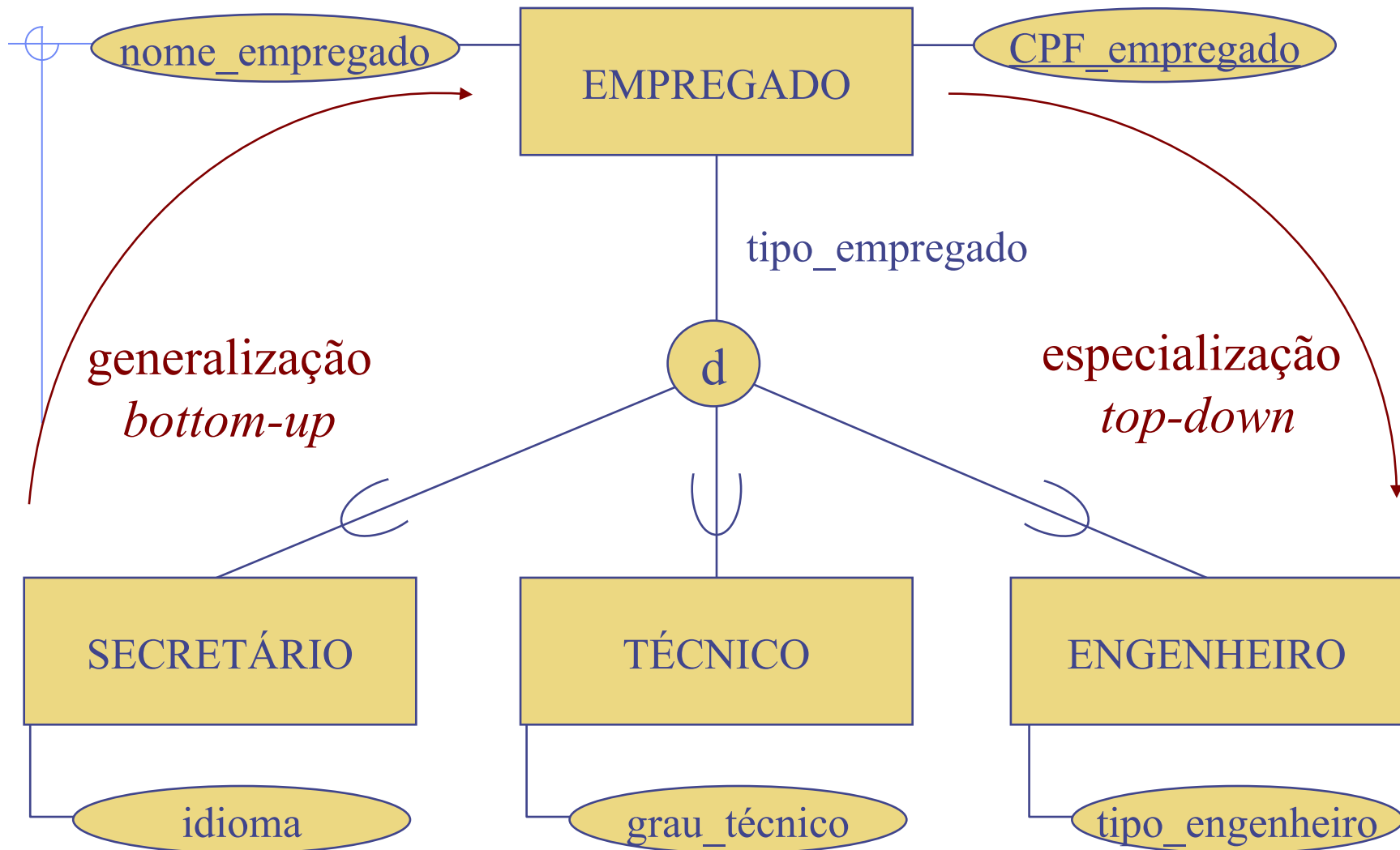
## ◆ Generalização

- resultado da união de dois ou mais tipos-entidade de nível mais baixo (subclasse), produzindo um tipo-entidade de nível mais alto (superclasse)
- é uma abstração de um conjunto de entidades
- passos:
  - ◆ suprime-se as diferenças entre os tipos-entidade
  - ◆ identifica-se os atributos em comum
  - ◆ generaliza-os em uma superclasse

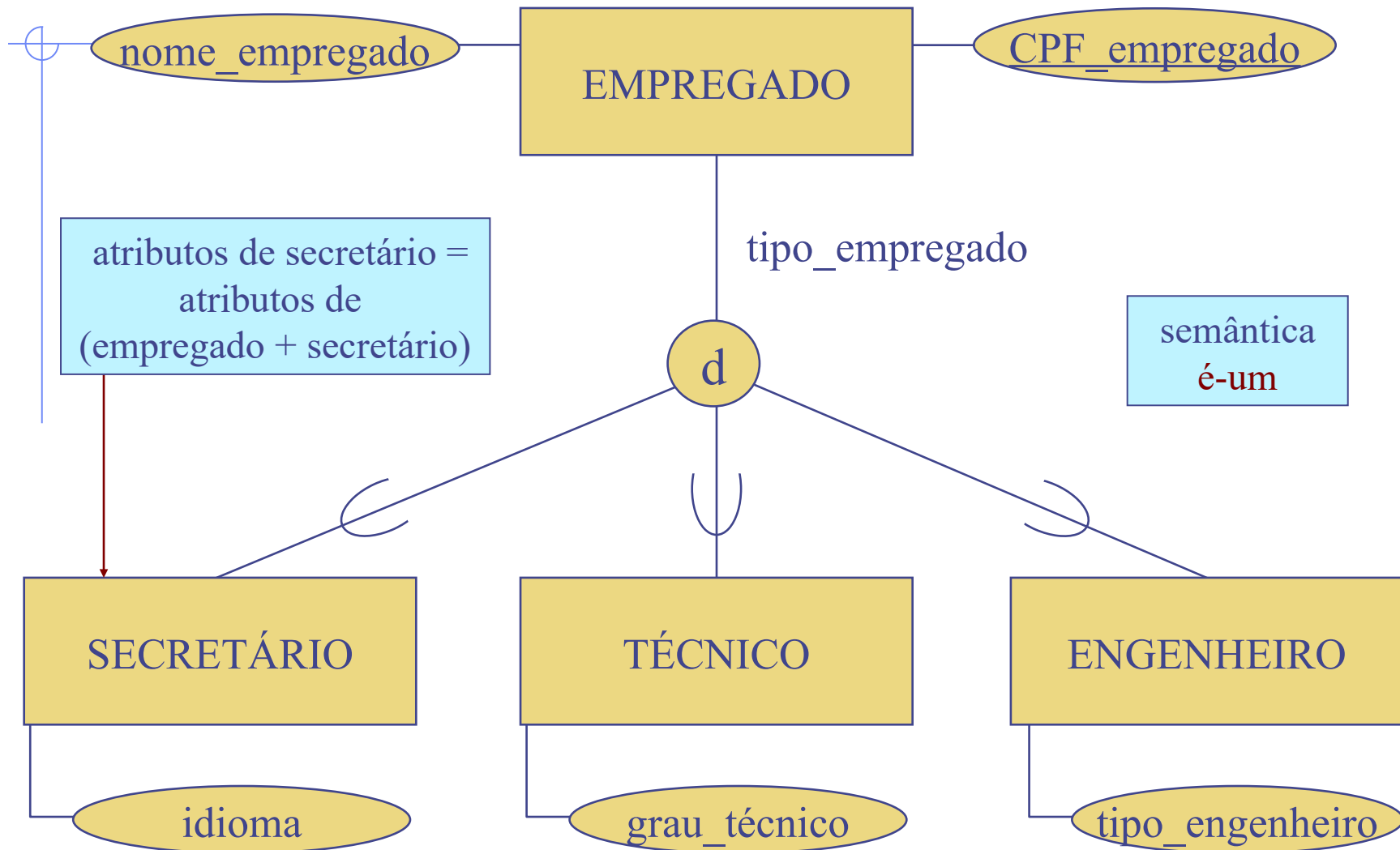
# Representação



# Representação



# Representação





# Chaves dos Tipos-Entidade

- ◆ Restrição de chave do ME-R: todos os tipos-entidade devem ter uma chave única
  - Restrição relaxada para o MER-X
    - ◆ subclasses não precisam ter chave explicitamente definida

# Restrições

## ◆ Especialização definida pelo atributo

- as subclasses que participam da hierarquia são determinadas por uma condição baseada em algum atributo da superclasse
- exemplo: atributo **tipo\_empregado**
- denominação:
  - ◆ subclasses definidas por predicado (ou definidas por condição)

# Restrições


## ◆ Especialização definida pelo usuário

- o membro da subclasse é determinado pelos usuários na operação que adicionar uma entidade à subclasse
  - ◆ um membro é especificado individualmente para cada entidade pelo usuário

# Restrição de Disjunção


## ◆ Subclasses mutuamente exclusivas

- uma entidade de uma superclasse deve ser membro, quando muito, de apenas uma subclasse

■ representação:  ← "d" (disjoint)

## ◆ Subclasses que se sobrepõem

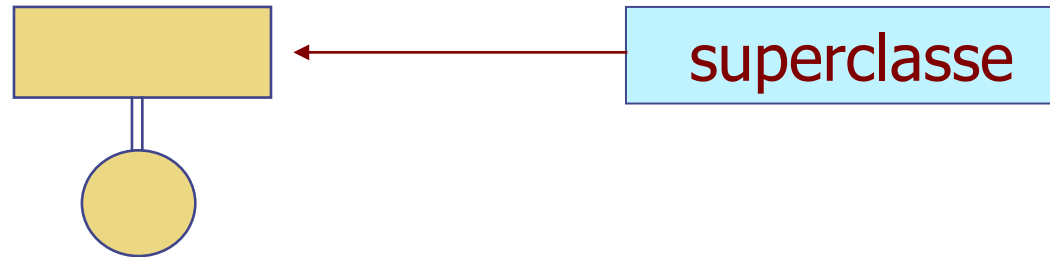
- uma entidade de uma superclasse pode ser membro de mais do que uma subclasse

■ representação:  ← "o" (overlap)

# Restrição de Completude

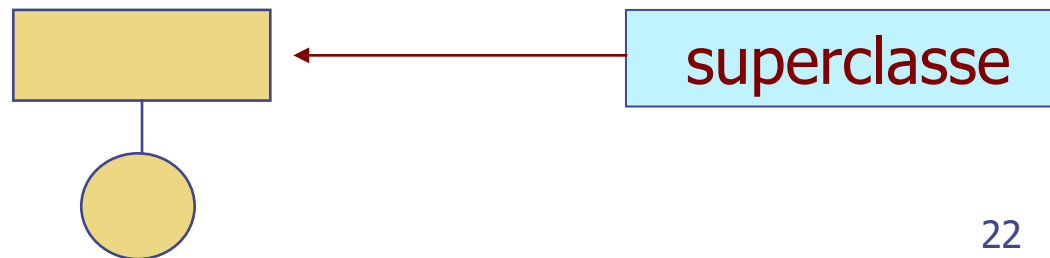
## ◆ Total

- cada entidade de uma superclasse deve ser membro de alguma subclasse na especialização
- representação:



## ◆ Parcial

- uma entidade de uma superclasse pode não pertencer a qualquer uma das subclasses
- representação:

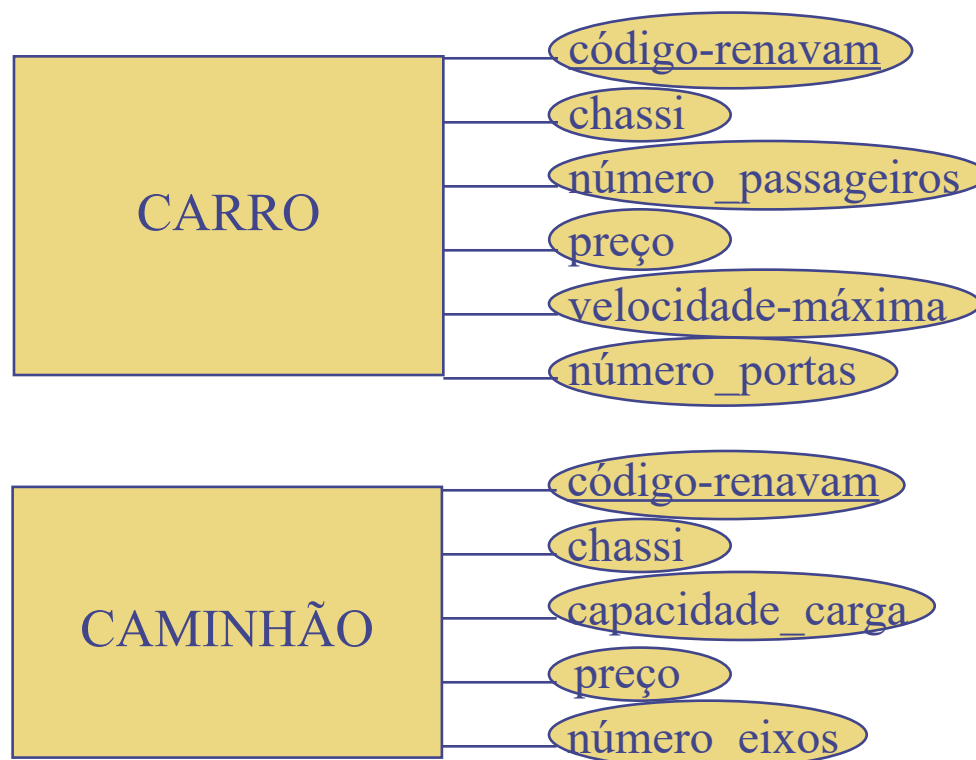


# Observações

- ◆ Restrições de disjunção e de completude são independentes
  - possibilidades de hierarquias
    - ◆ total disjunta
    - ◆ parcial disjunta
    - ◆ total com sobreposição
    - ◆ parcial com sobreposição

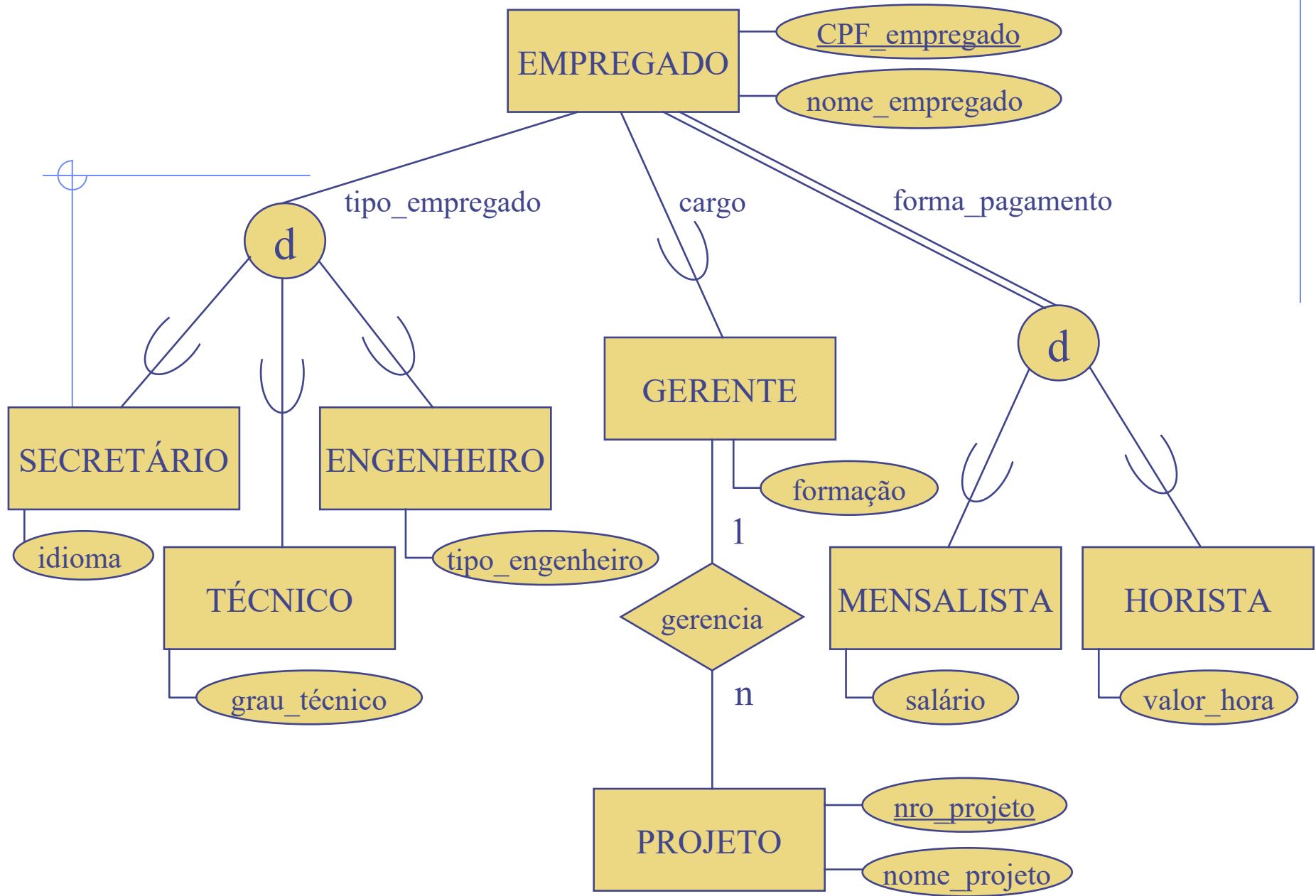
# Exercício

- ◆ Modele uma hierarquia de generalização/especialização para os tipos-entidade *carro* e *caminhão*. Defina as restrições de disjunção e de completude.



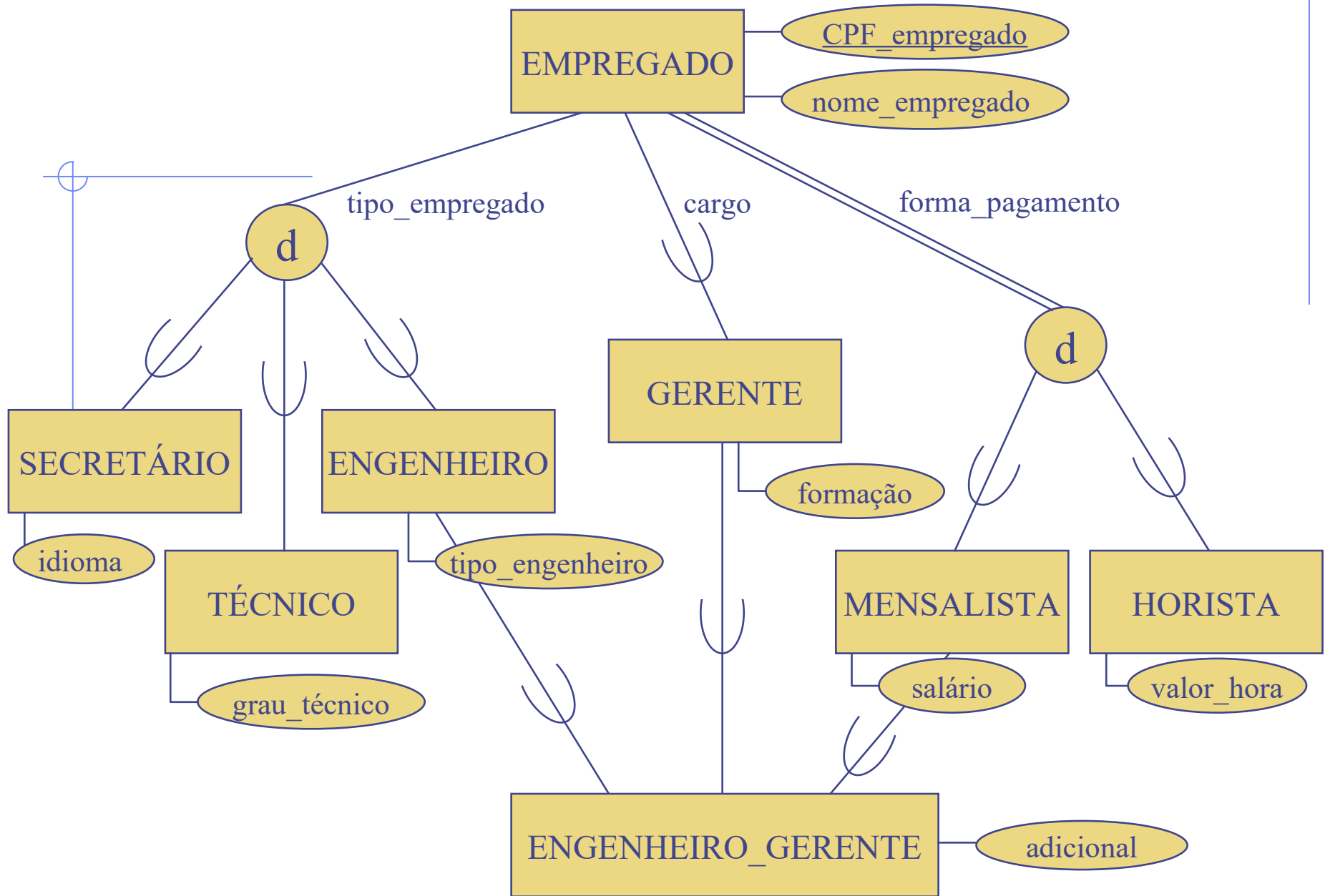


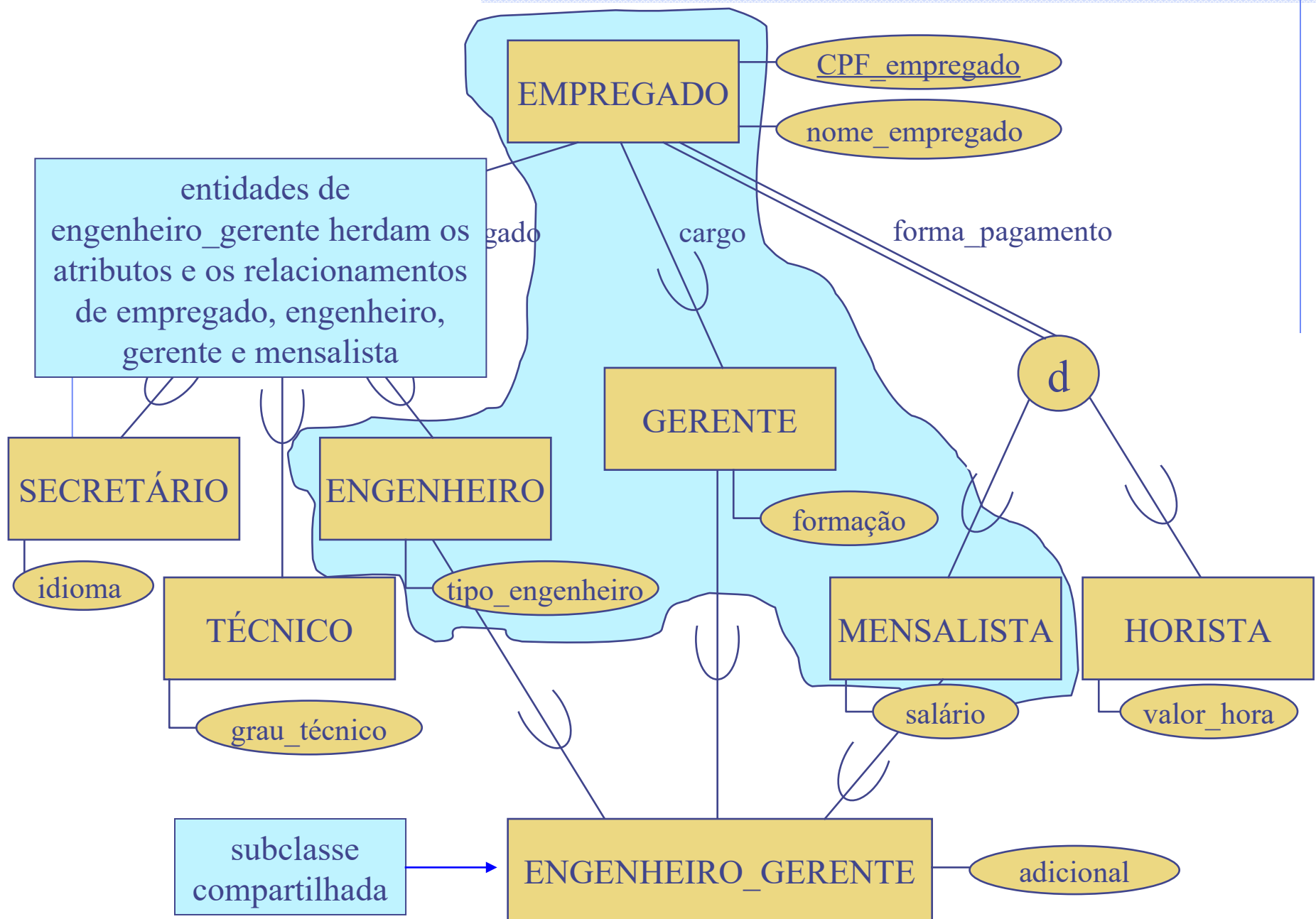




# Generalização/Especialização

- ◆ Uma subclasse pode possuir outras subclasses especificadas a partir dela
- ◆ Herança simples
  - cada subclasse participa como subclasse em **apenas um** relacionamento superclasse/subclasse
- ◆ Herança múltipla
  - cada subclasse pode participar como uma subclasse em **mais do que um** relacionamento superclasse/subclasse





# Herança Múltipla

## ◆ Regra

- se um mesmo atributo ou relacionamento for herdado mais do que uma vez por diferentes relacionamentos superclasse/subclasse então o atributo ou o relacionamento deve ser incluído apenas uma vez na subclasse

# Agregação

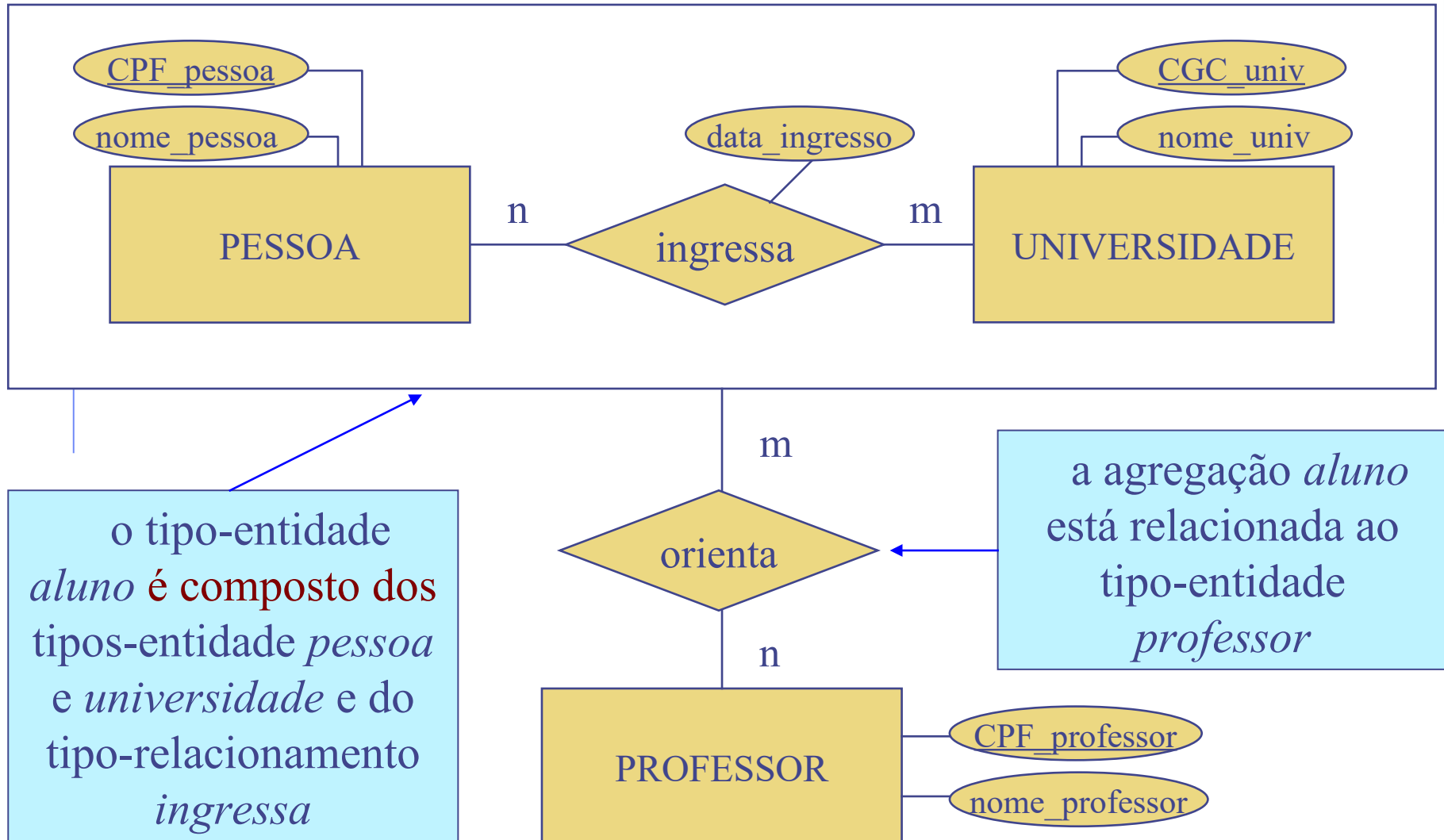
- ◆ É um conceito para a construção de objetos compostos a partir de seus objetos componentes
  - Idéia: elementos de modelagem podem associar-se, formando outros elementos que representam essa associação
- ◆ Pode assumir diversas formas:
  - Agregando atributos em Tipos-Entidade e Tipos-Relacionamento
    - ◆ os valores dos atributos compõem a entidade
  - Agregando Tipos-Entidade e Tipos-Relacionamentos
    - ◆ combinar entidades que estão relacionadas por uma instância de relacionamento em uma entidade agregada de alto nível

# Agregação

- ◆ Tipos-entidades agregados são representados como tipos-entidades comuns
- ◆ Engloba
  - dois tipos-entidades e um tipo-relacionamento

# Representação

ALUNO





# Agregação

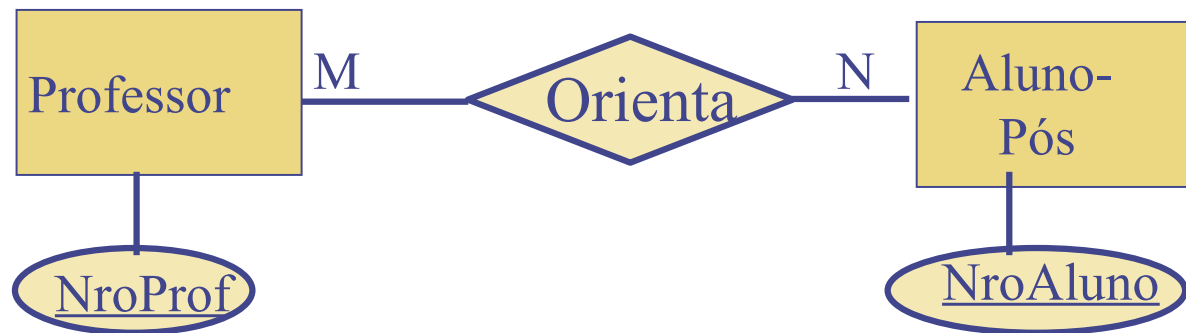
◆ Situações que indicam a necessidade de agregação:

- 1) Quando é necessário identificar cada relacionamento (o relacionamento tem chave)
- 2) Quando é necessário mais de um relacionamento envolvendo as mesmas entidades
- 3) Quando existe a necessidade de associar dois relacionamentos

# Agregação

◆ **1o. Caso:** O tipo relacionamento tem um identificador próprio:

- Nesse caso, embora seja possível identificar a entidade agregação por um identificador próprio, ela também pode ser identificada pelo relacionamento entre as entidades que participam do relacionamento:

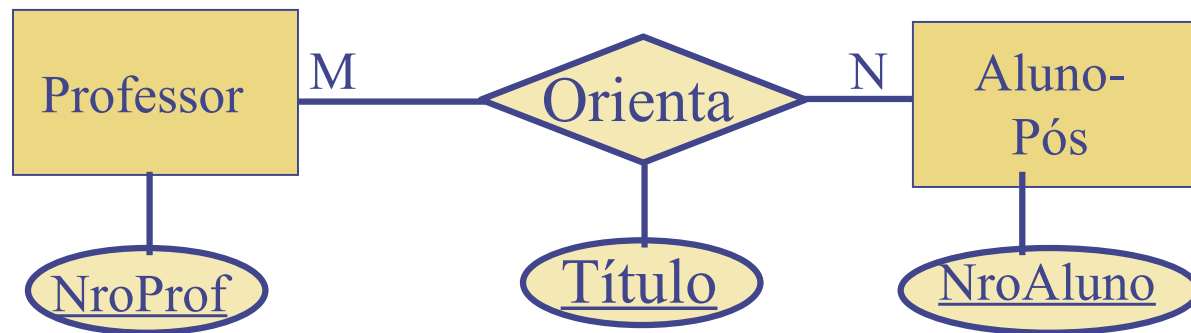


Título único em todo o sistema

# Agregação

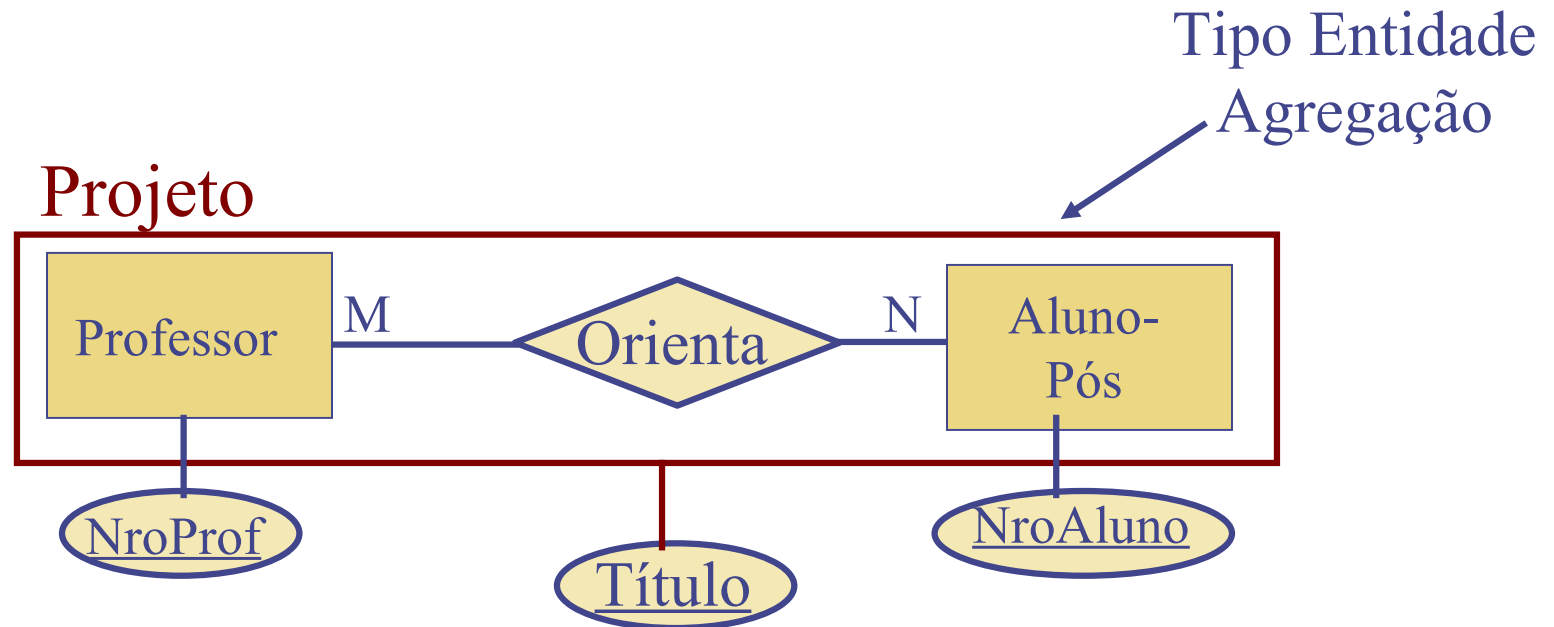
◆ **1o. Caso:** O tipo relacionamento tem um identificador próprio:

- Nesse caso, embora seja possível identificar a entidade agregação por um identificador próprio, ela também pode ser identificada pelo relacionamento entre as entidades que participam do relacionamento:



**O Tipo Relacionamento não tem identificador!!**

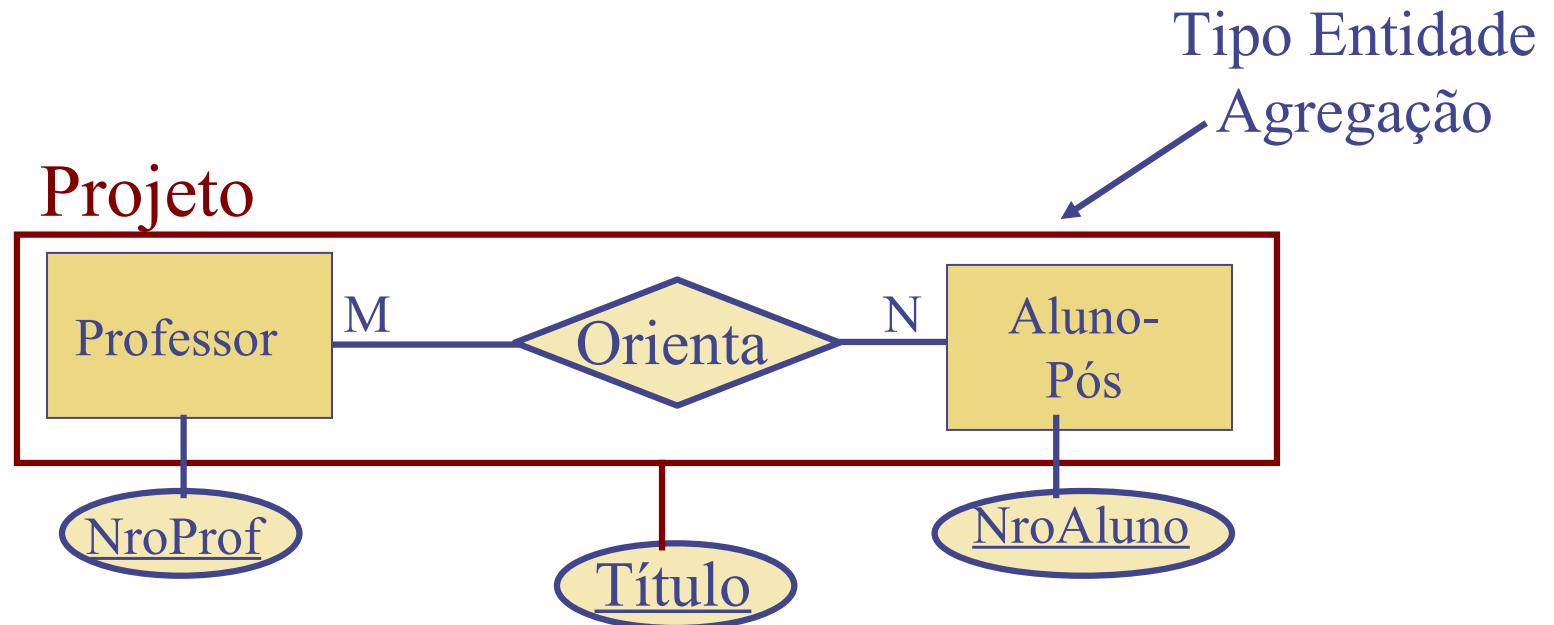
# Agregação



Chave:

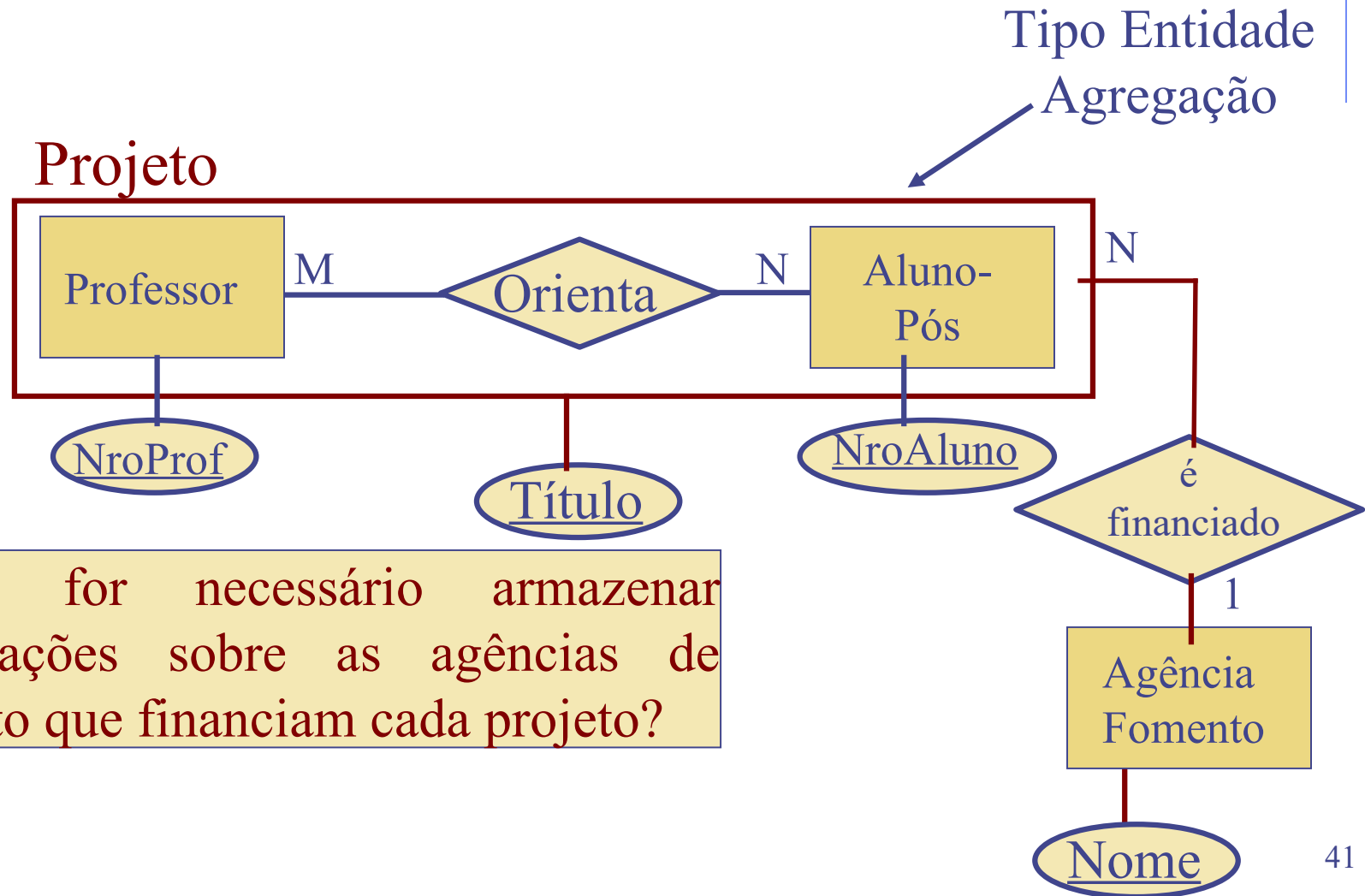
- NroProfessor + NroAluno
- ou
- Título

# Agregação



E se for necessário armazenar informações sobre as agências de fomento que financiam cada projeto?

# Agregação

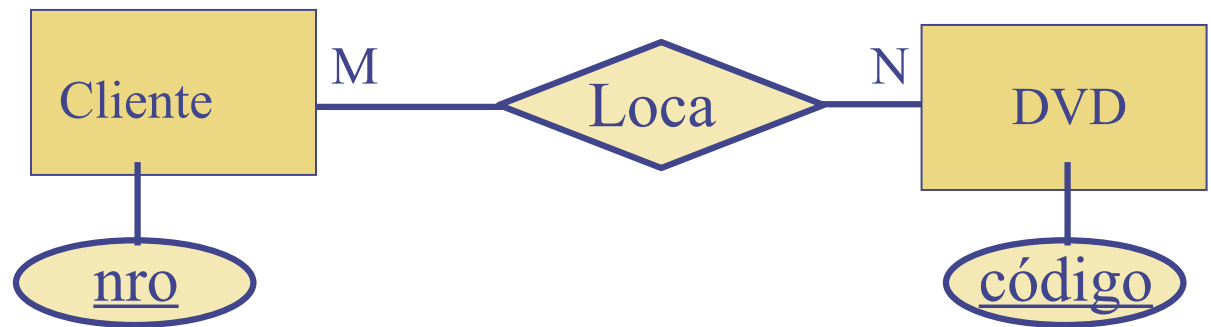


E se for necessário armazenar informações sobre as agências de fomento que financiam cada projeto?

# Agregação

◆ **2o. Caso:** Pode haver mais de um relacionamento envolvendo as mesmas entidades

- Como identificar cada locação?



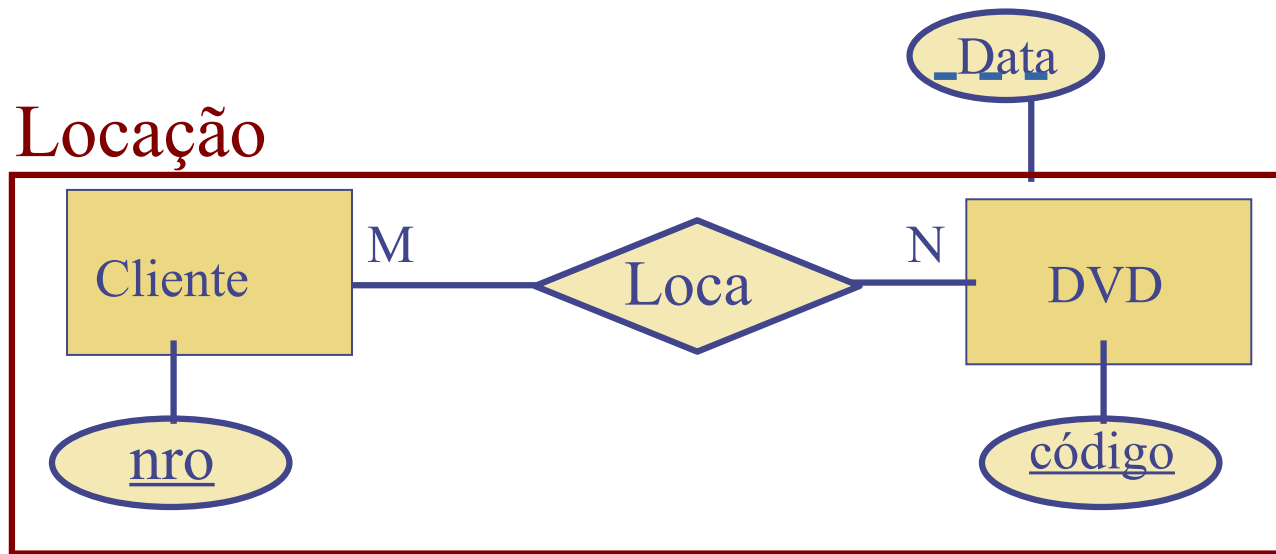
# Agregação

◆ **2o. Caso:** Pode haver mais de um relacionamento envolvendo as mesmas entidades

- Como identificar cada locação?

Chave:

- nro +
- código +
- data





# Agregação

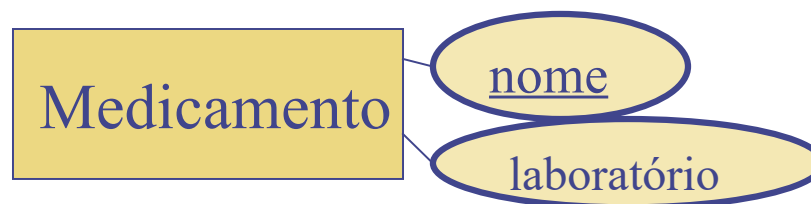
- ◆ **3o. Caso:** Pode haver a necessidade de associar dois tipos relacionamentos



- Suponha que seja necessário alterar esse modelo do seguinte modo: é necessário saber que medicamentos existem e que medicamentos foram prescritos em cada consulta.

# Agregação

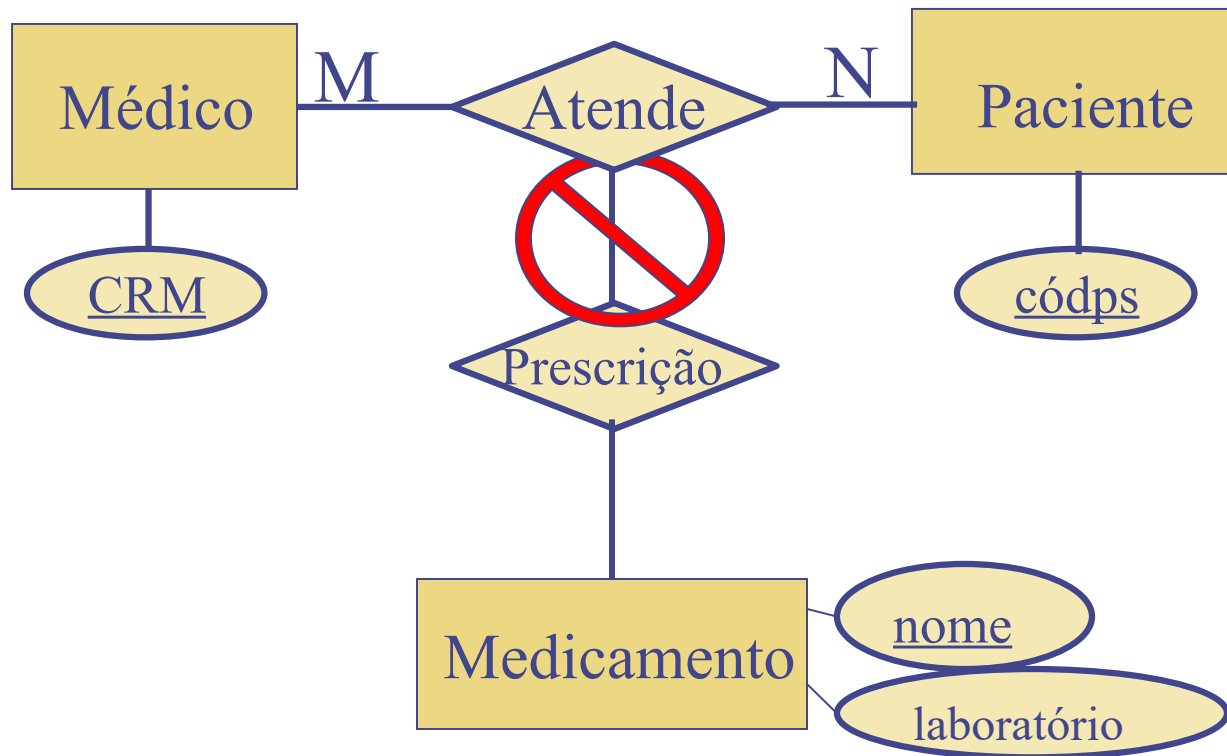
◆ **3o. Caso:** Pode haver a necessidade de associar dois tipos relacionamentos



- A questão agora é:  
com que tipo  
entidade existente  
o novo tipo  
entidade deve ser  
associado?

# Agregação

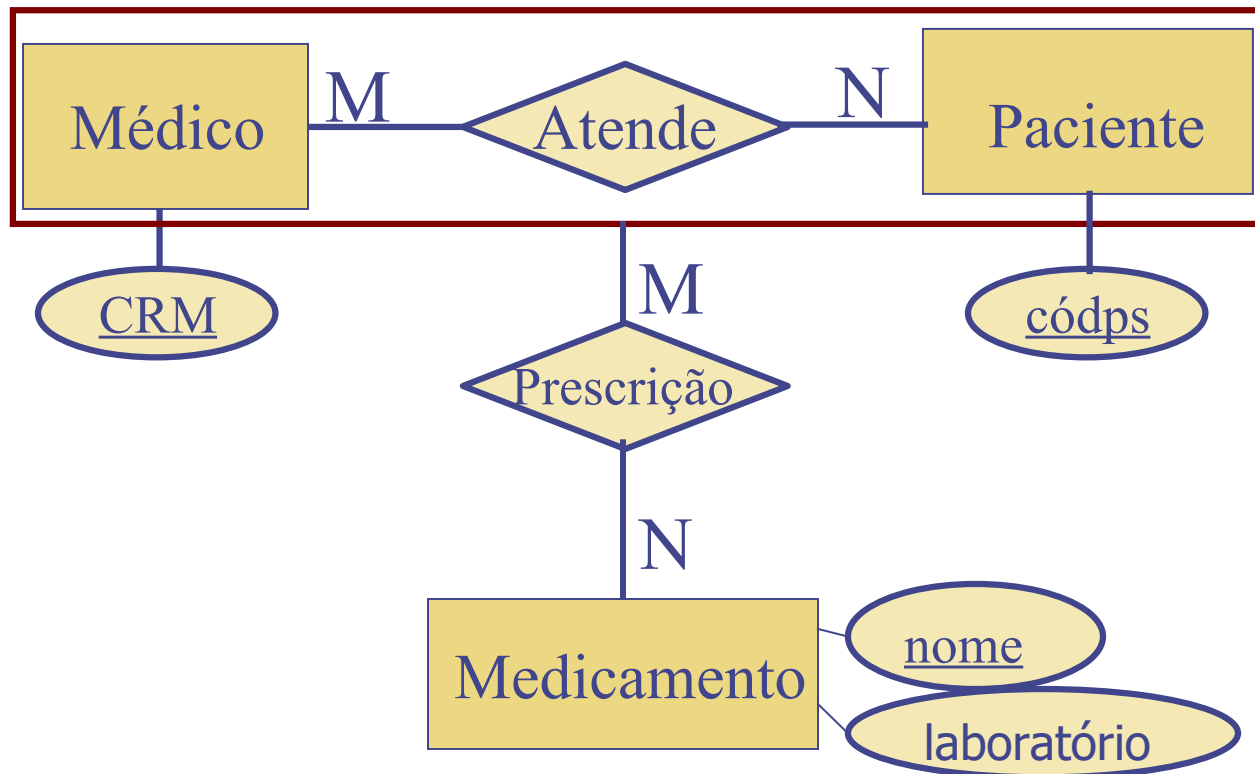
- ◆ **3o. Caso:** Pode haver a necessidade de associar dois tipos relacionamentos



# Agregação

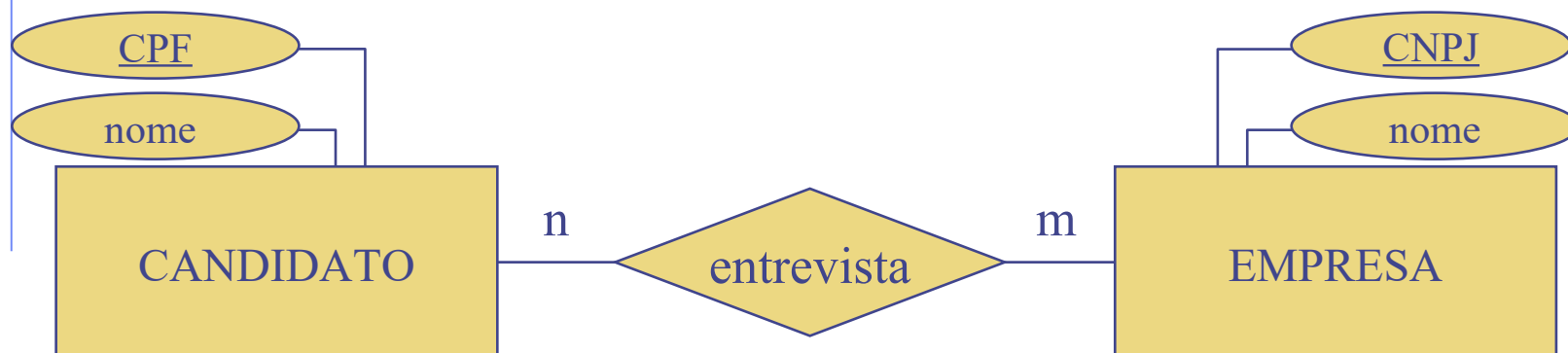
- ◆ **3o. Caso:** Pode haver a necessidade de associar dois tipos relacionamentos

## Consulta



# Exercício

◆ Considere o esquema ER abaixo:

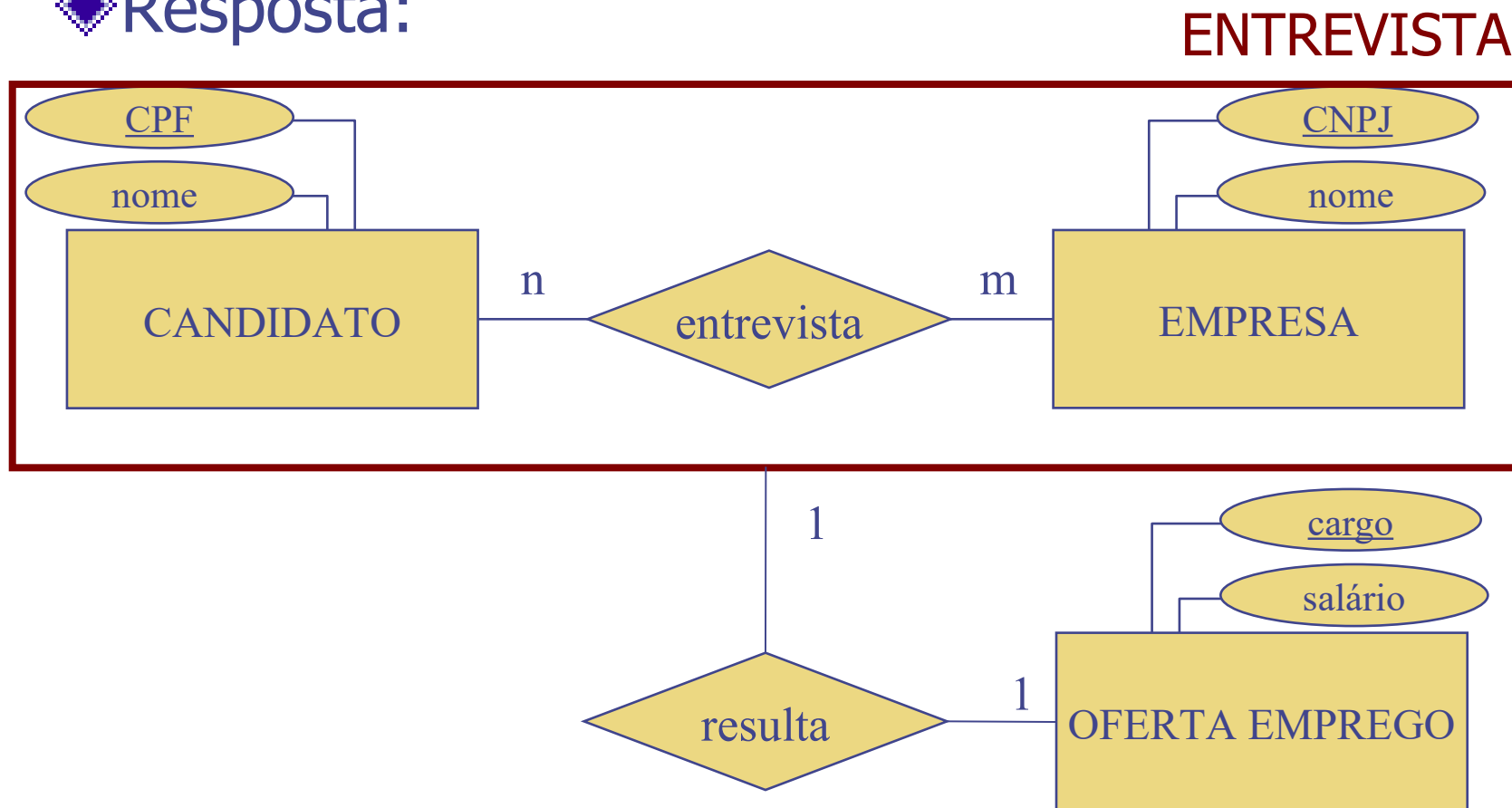


- Como modelar a situação em que algumas entrevistas resultam em uma oferta de emprego (com cargo e salário inicial) e outras não?

# Exercício

Existe uma outra maneira de modelar  
essa mesma situação?  
Fica para pensar em casa...

◆ Resposta:



# Projeto Lógico de BD

## ◆ Classificar tipos-entidades e atributos

- tipos-entidade possuem informações descritivas, atributos não
- atributos devem ser mantidos de forma atômica
- atributos devem ser relacionados às entidades que eles descrevem

## ◆ Identificar chaves primárias

# Projeto Lógico de BD

- ◆ Identificar tipos-relacionamentos e seus atributos
  - determinar o grau dos tipos-relacionamentos
    - ◆ definir tipos-relacionamento ternários cuidadosamente
  - identificar as restrições que se aplicam sobre cada tipo-relacionamento
    - ◆ cardinalidade
    - ◆ participação
  - Caso necessário, definir os papéis
- ◆ Identificar tipo-entidade forte e tipo-entidade fraca



# Projeto Lógico de BD

- ◆ Verificar os requisitos de operações
  - se eles se referirem a dados que não estão modelados, repetir os passos anteriores.
- ◆ Modelar hierarquias de generalização
  - identificar atributos e relacionamentos comuns
  - determinar as restrições de disjunção e de completude
- ◆ Modelar agregações

*"Podemos notar que um projetista de BD necessita de um bom conhecimento do minimundo que está sendo modelado para que possa tomar essas decisões!"*

# Bibliografia

- ◆ Elmasri, Ramez; Navathe, Shamkant B. **Sistemas de banco de dados**. 4 ed. São Paulo: Addison Wesley, 2005, 724 p. Bibliografia: p. [690]-714.

# Material indicado para estudo complementar para casa

- ◆ Capítulo 8 do livro: Elmasri, Ramez; Navathe, Shamkant B. Sistemas de banco de dados. 6ª edição.
  - Lista de exercícios “Modelo Entidade Relacionamento Estendido”
  
- ◆ Haverá uma atividade sobre o Modelo Entidade Relacionamento Estendido no dia 16/03. Essa atividade será assíncrona.