

Modelos Conceituais de Dados

Banco de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

Motivação

- ◆ Objetivo da abordagem de BD:
 - oferecer abstração dos dados
 - separar aplicações dos usuários dos detalhes de hardware
 - ferramenta utilizada: [modelo de dados](#)
- ◆ Modelo de dados:
 - conjunto de ferramentas conceituais para a descrição dos dados e dos relacionamentos existentes entre os dados, da semântica e das restrições que atuam sobre estes

Categorias de Modelos de Dados

- ◆ Divisão baseada nos tipos de conceitos oferecidos para descrever a estrutura do BD
- ◆ Modelo de dados conceitual
 - modelo de alto nível
 - oferece conceitos próximos aos usuários
 - exemplo: [modelo entidade-relacionamento](#)

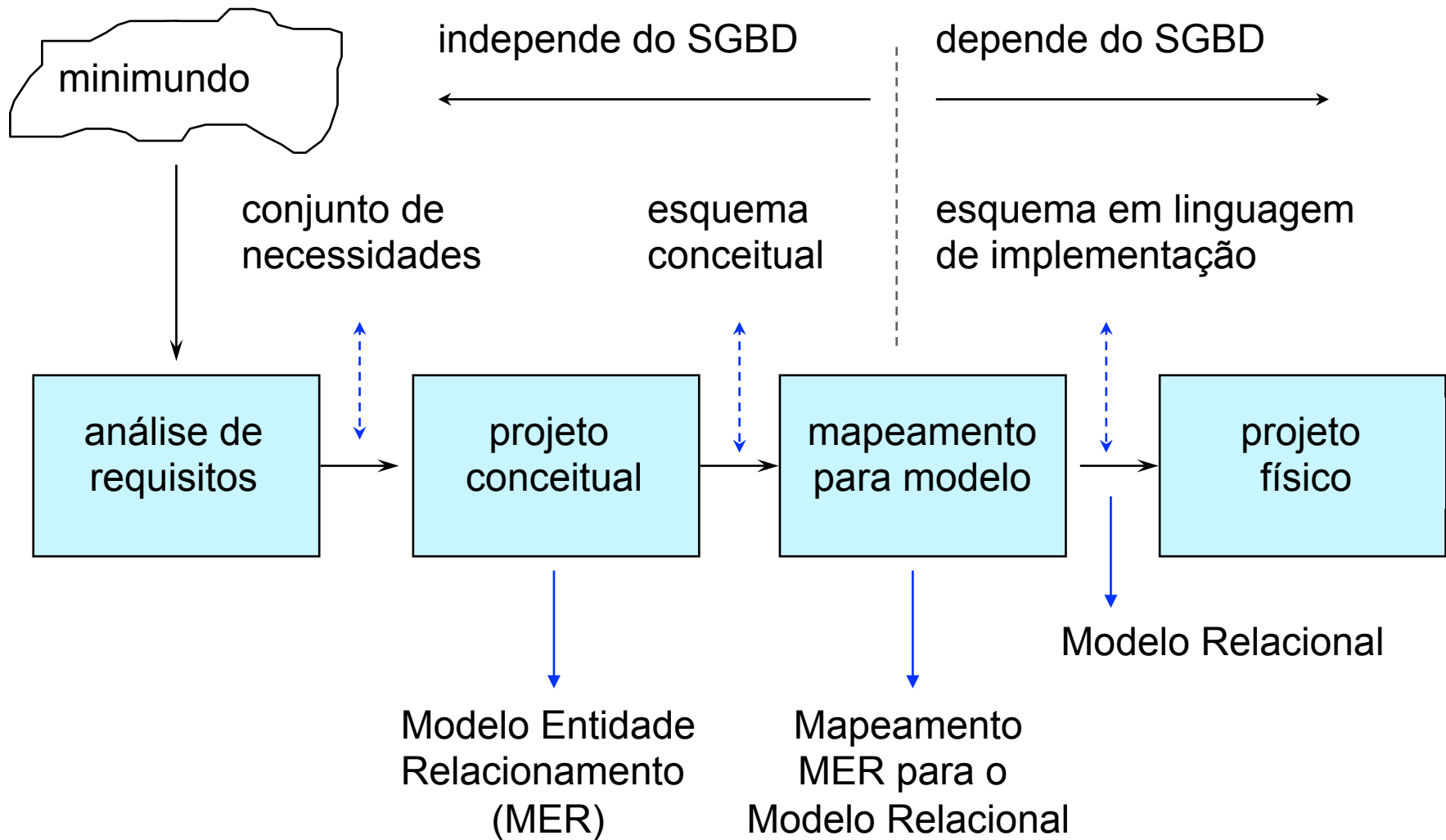
Categorias de Modelos de Dados

- ◆ Modelo de dados de implementação
 - oferece conceitos que
 - ◆ podem ser facilmente utilizados por usuários finais
 - ◆ não estão distantes da maneira na qual os dados estão organizados dentro do computador
 - é implementado de maneira direta
 - exemplo: **modelo relacional**

Categorias de Modelos de Dados

- ◆ Modelo de dados físico
 - modelo de baixo nível
 - descreve como os dados estão armazenados fisicamente no computador

Projeto de BD



Modelo Entidade Relacionamento Básico

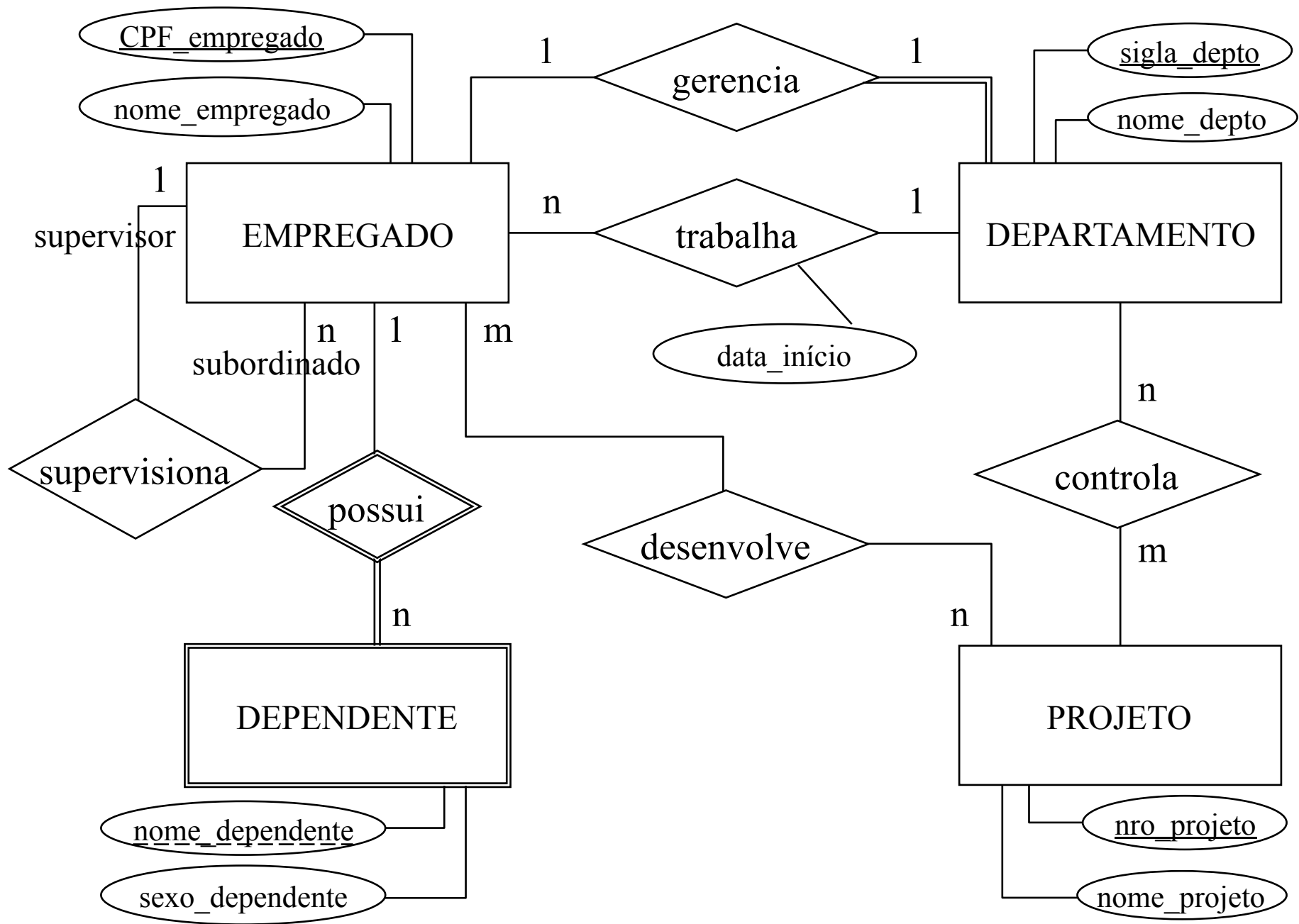
♦ Característica

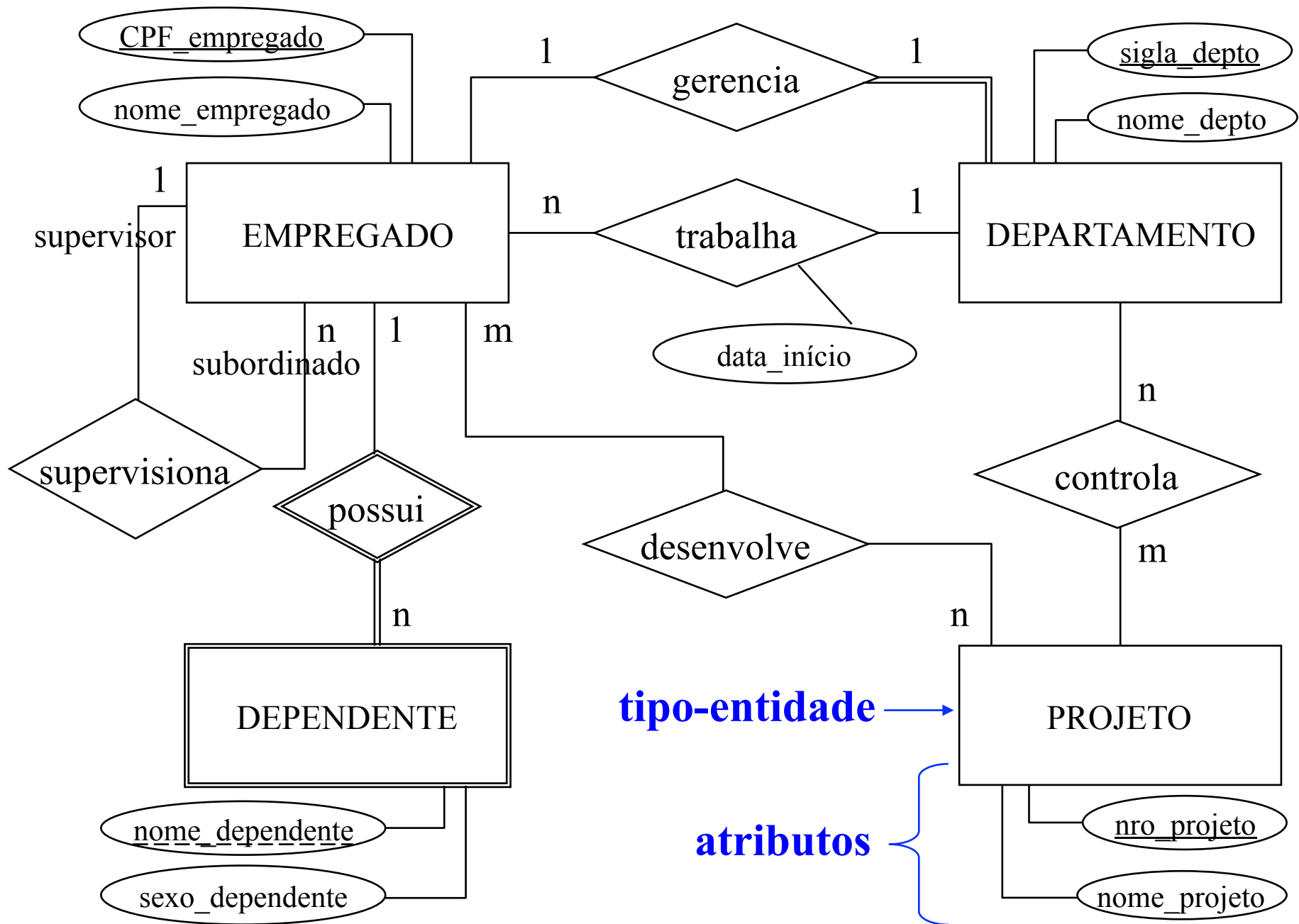
- proposto por Chen (1970)
- representa um problema como um conjunto de **entidades** e de **relacionamentos** entre estas entidades
- utilizado na modelagem de aplicações de BD tradicionais

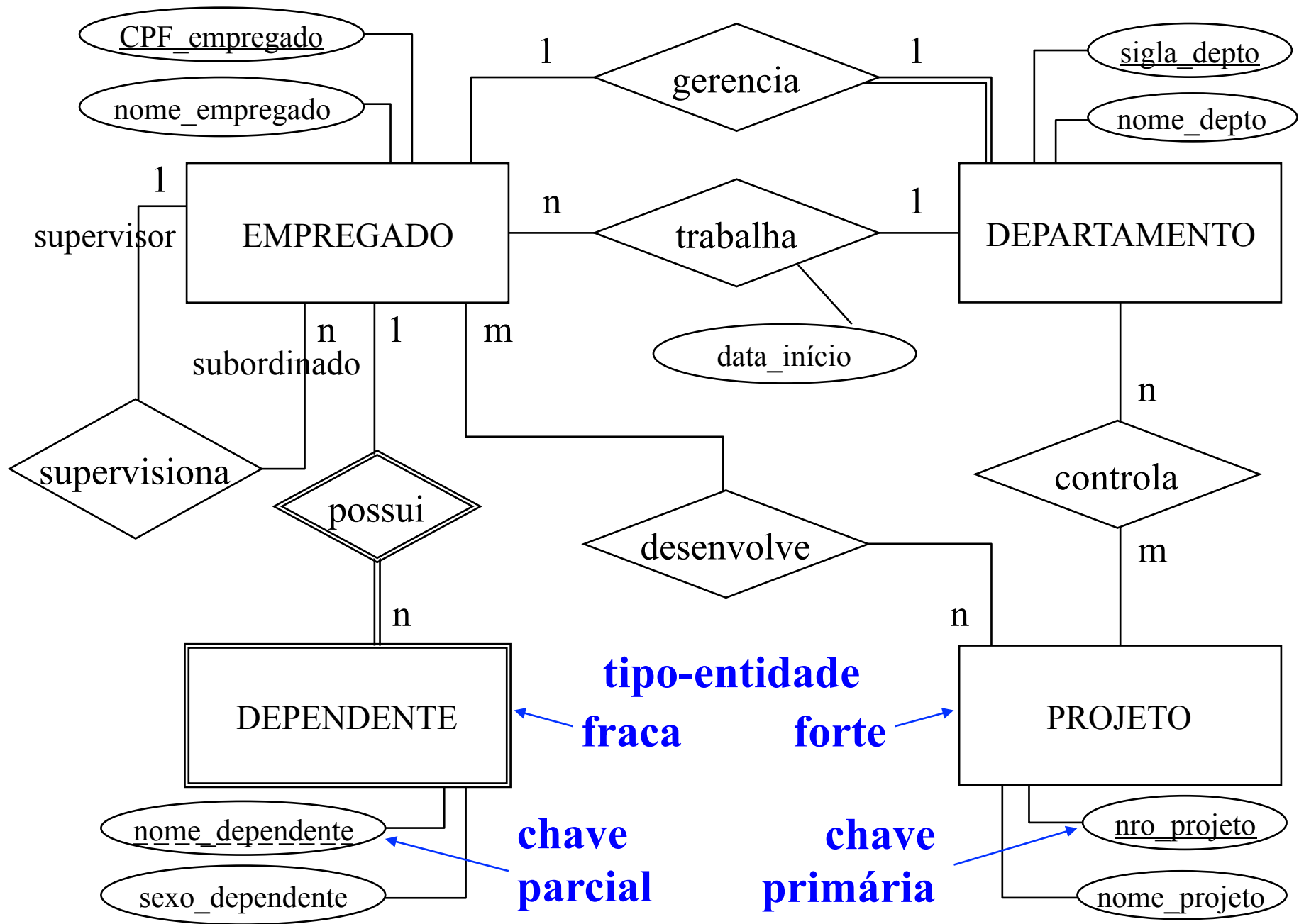
Modelo Entidade Relacionamento Básico

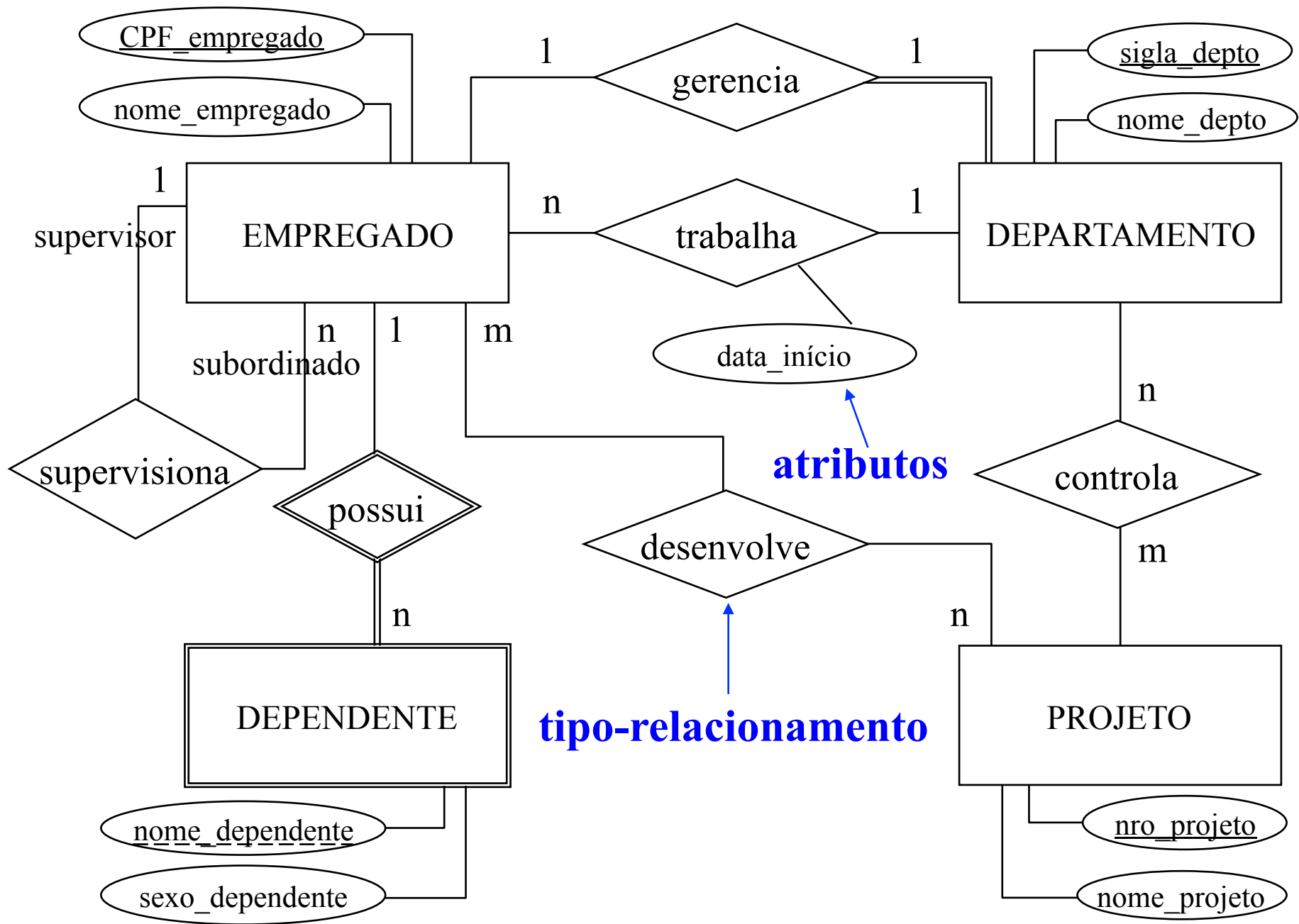
◆ Conceitos Básicos

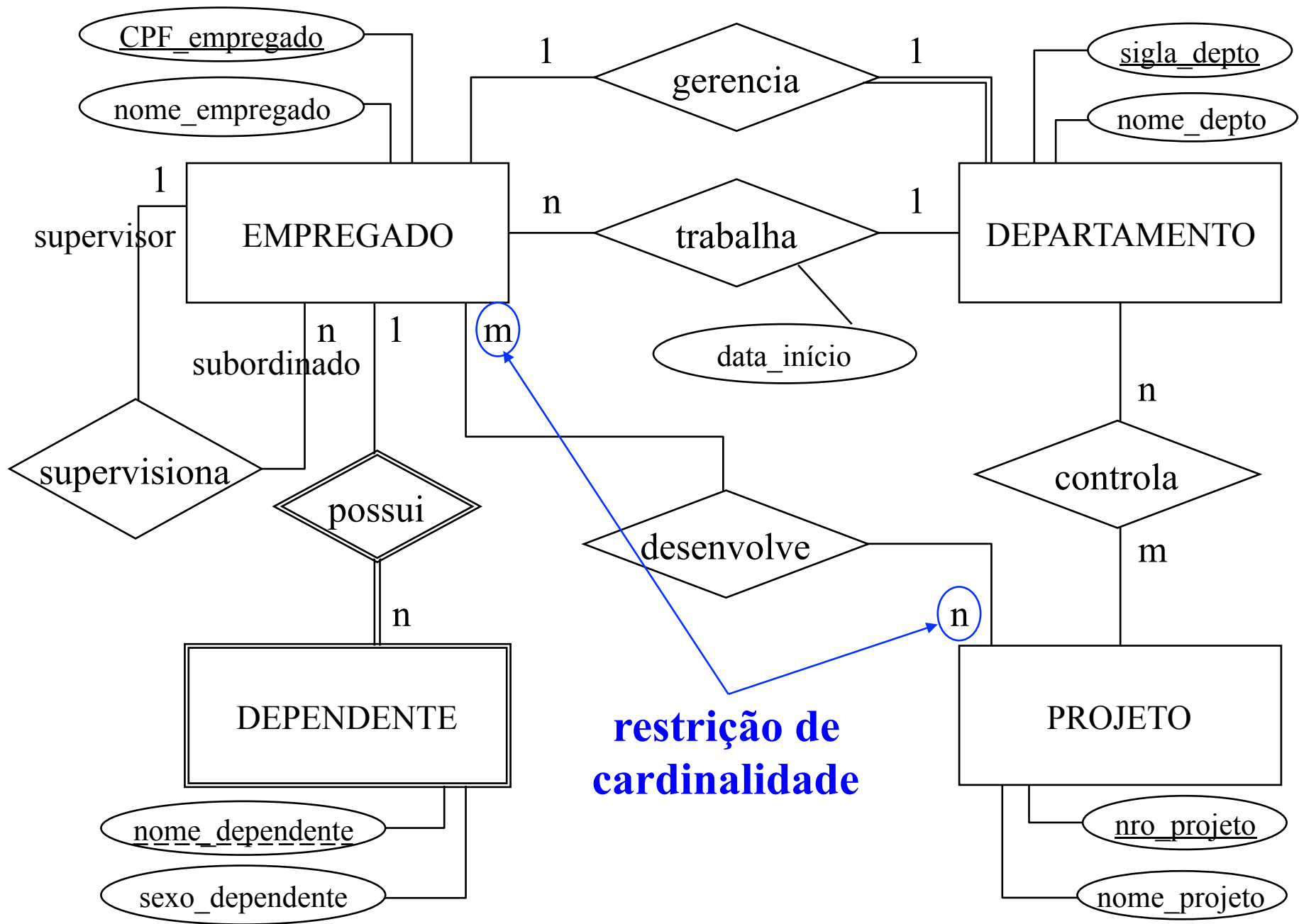
- entidade; atributo; tipo-entidade
- chave primária
- relacionamento; atributo; tipo-relacionamento unário, binário, ternário
- restrições de cardinalidade e de participação
- tipo-entidade fraca

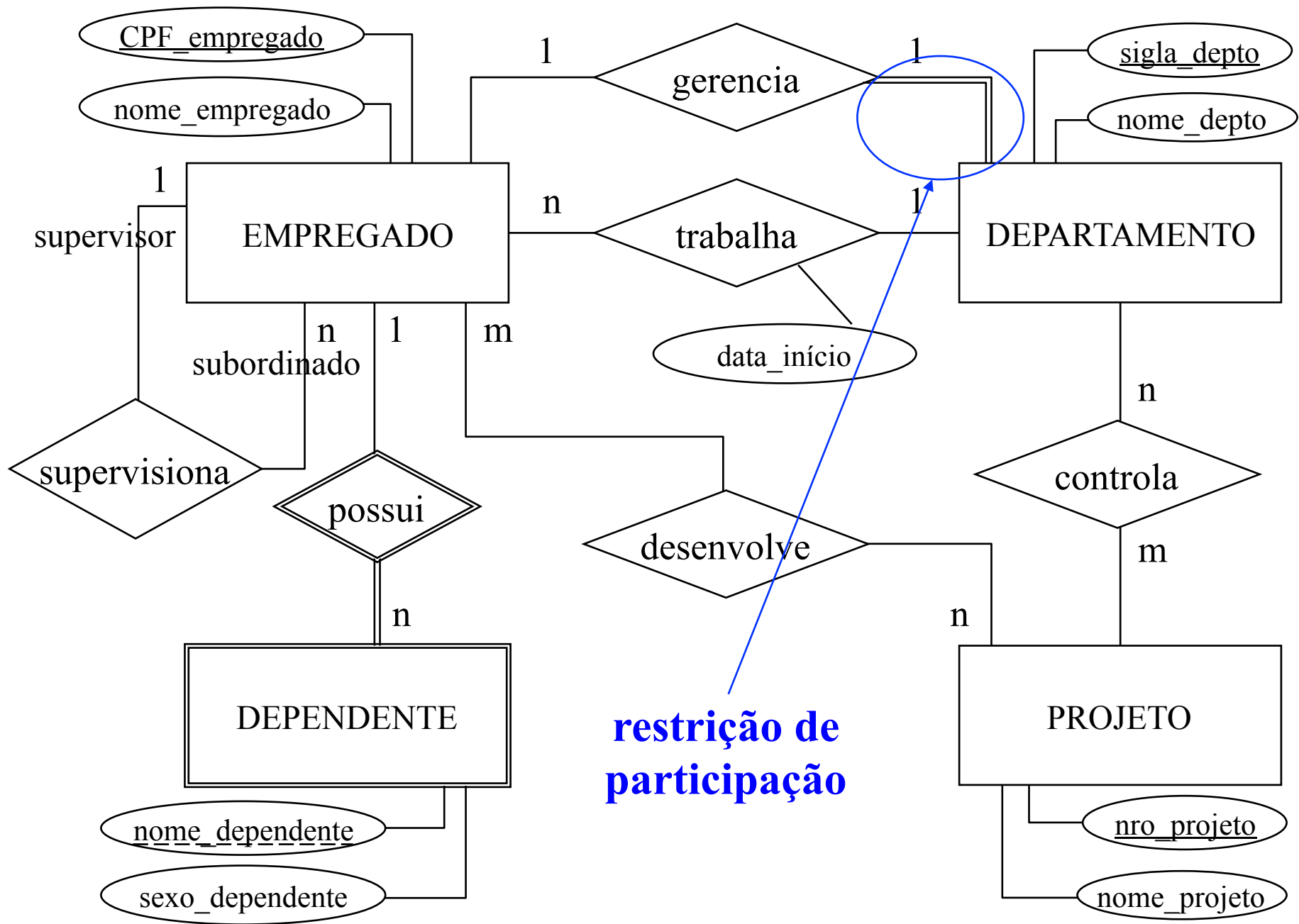




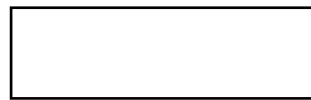








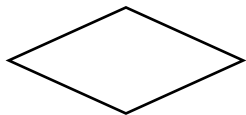
Resumo da Notação



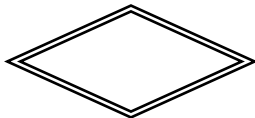
tipo entidade forte



tipo entidade fraca



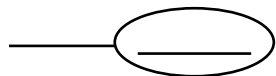
tipo relacionamento



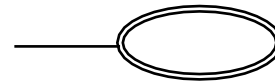
identificador de relacionamento



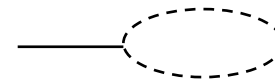
atributo



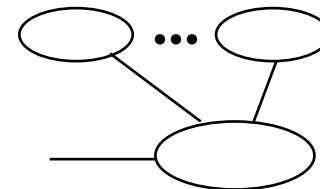
atributo chave



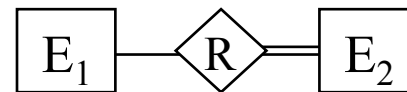
atributo multivalorado



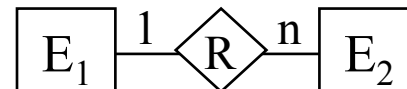
atributo derivado



atributo composto



participação total de E_2 em R



cardinalidade 1:N

Modelo Entidade Relacionamento Estendido

◆ Características

- introduz semântica adicional ao modelo ER
- utilizado na modelagem de aplicações mais complexas

◆ Conceitos

- generalização, especialização, e restrições
- agregação
- categoria

Subclasse/Superclasse

◆ Subclasse

- subagrupamento das entidades de um 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 herdados 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

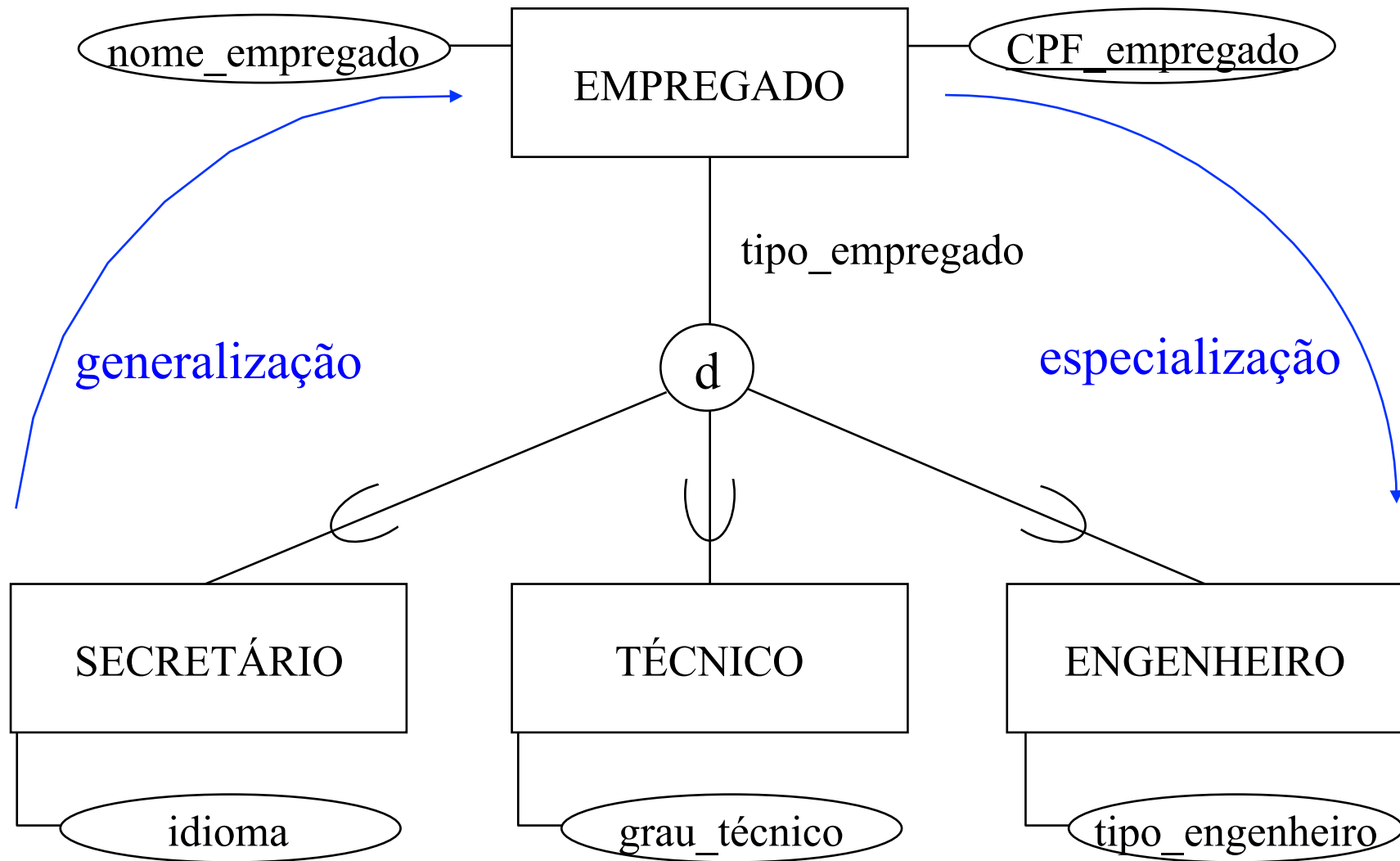
- resultado da separação de um tipo-entidade de nível mais alto (superclasse), formando vários tipos-entidades de nível mais baixo (subclasse)
- passos:
 - ◆ define-se um conjunto de subclasses de um tipo-entidade
 - ◆ associa-se atributos adicionais específicos às subclasses
 - ◆ estabelece-se tipos-relacionamentos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

Generalização/Especialização

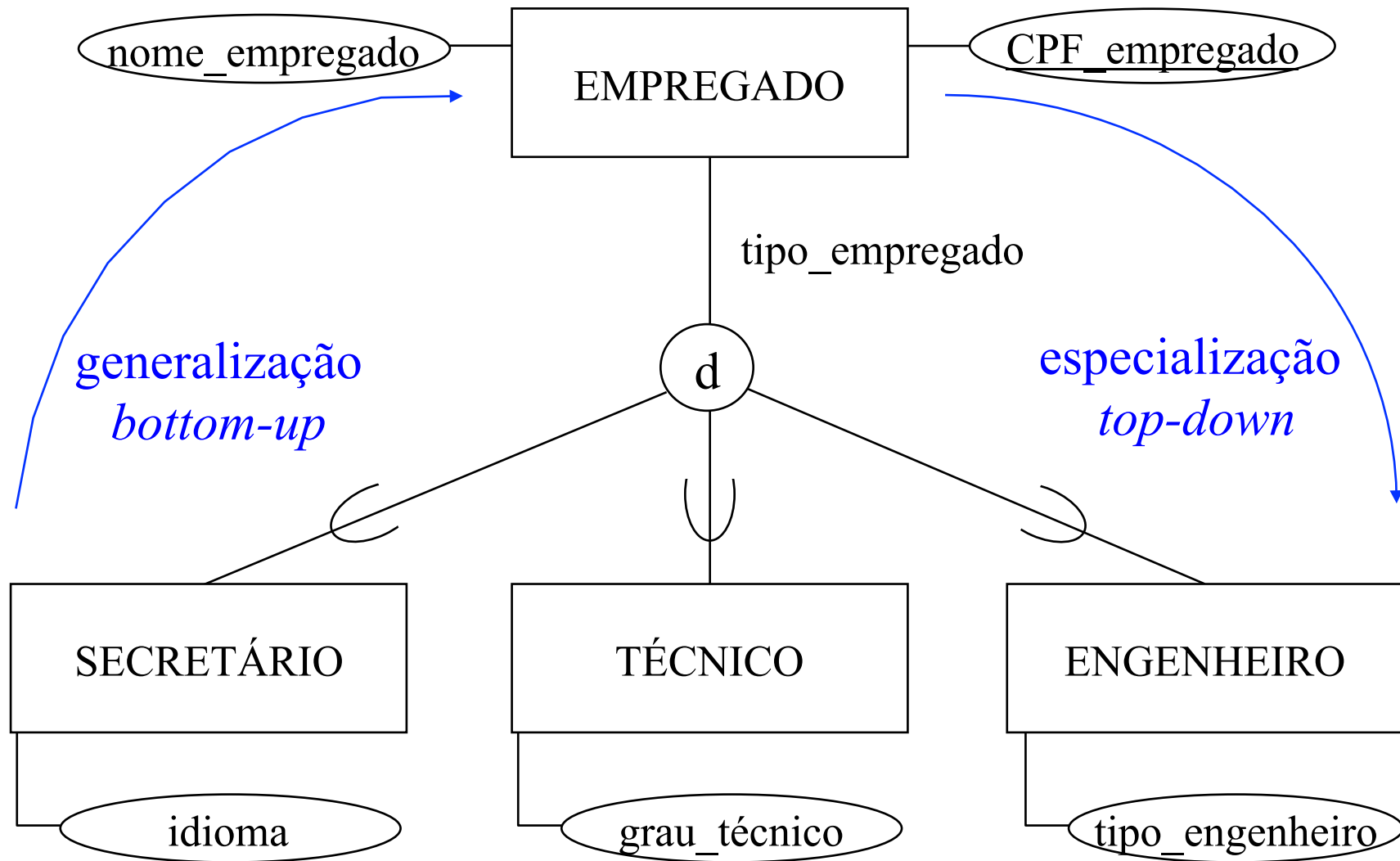
◆ Generalização

- resultado da união de dois ou mais tipos-entidades 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

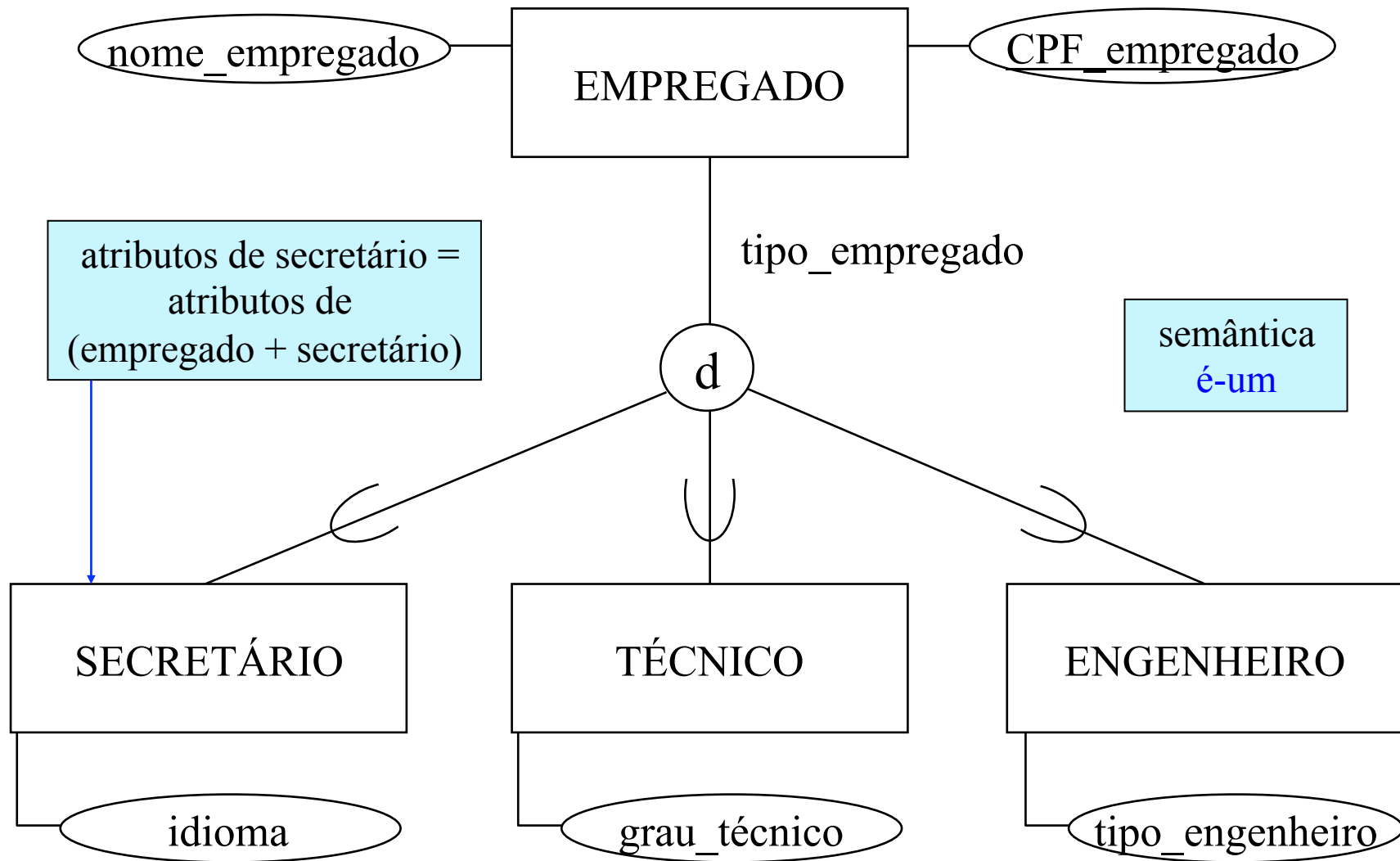
Representação



Representação



Representação

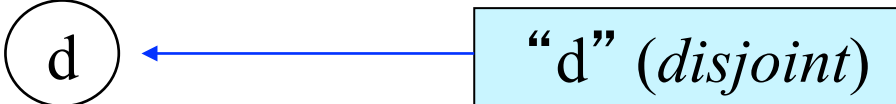


Restrições

- ◆ Especialização definida pelo atributo
 - as subclasses que participam da hierarquia são determinadas por uma condição baseada em algum atributo da superclasse
 - exemplo: `tipo_empregado`
- ◆ Denominações
 - subclasses definidas pelo predicado
 - subclasses definidas pela condição

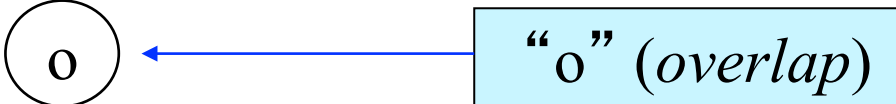
Restrição de Disjunção

- ◆ Subclasses mutuamente exclusivas
 - uma entidade de uma superclasse deve ser membro, quando muito, de apenas uma única subclasse

– representação: 

- ◆ Subclasses que se sobrepõem

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

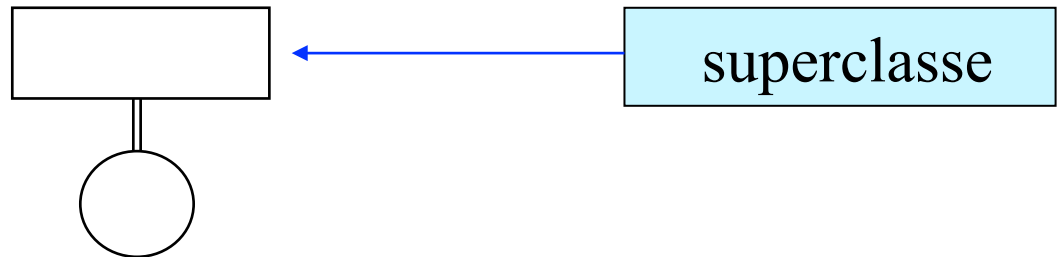
– representação: 

Restrição de Completude

◆ Total

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

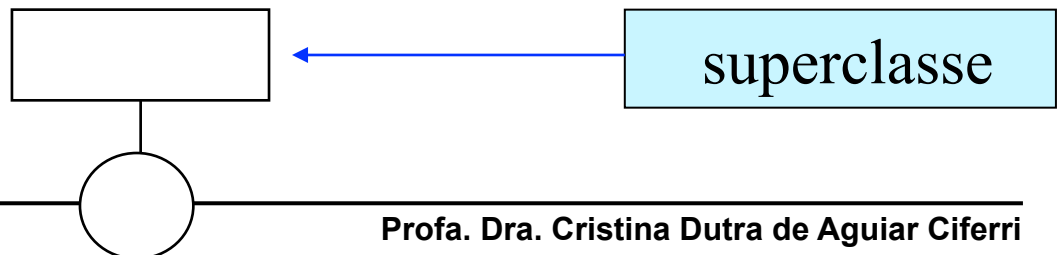
- representação:



◆ Parcial

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

- representação:



Observações

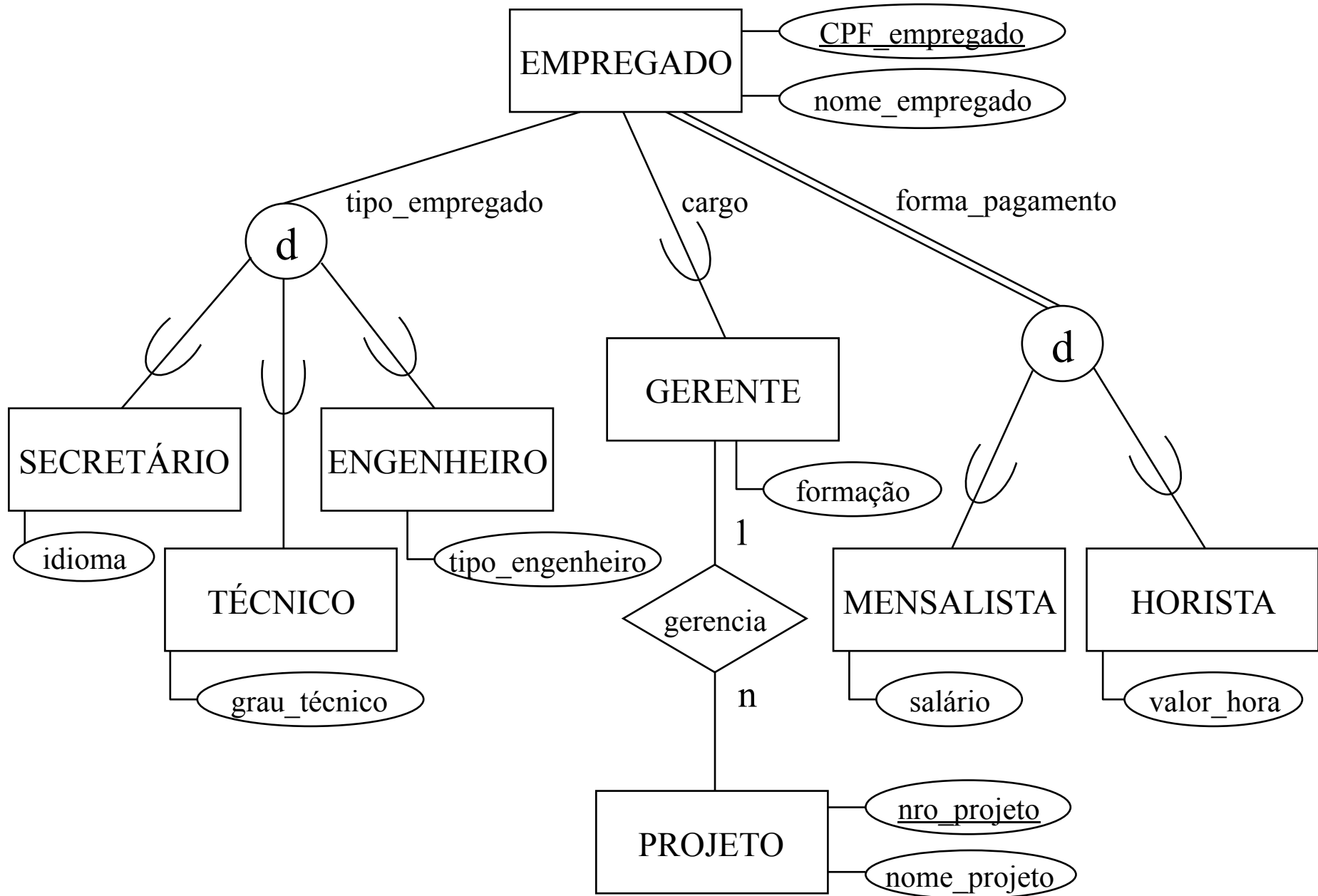
- ◆ Restrições de disjunção e de completude são independentes
 - possibilidades de hierarquias
 - ◆ total disjunta
 - ◆ parcial disjunta
 - ◆ total com sobreposição
 - ◆ parcial com sobreposição

Observações

- ◆ Regras de inserção e remoção
 - se uma entidade de uma superclasse for removida, então ela deve ser automaticamente removida de todas as subclasses a que pertence
 - se uma entidade for inserida em uma superclasse, então ela deve ser necessariamente inserida em todas as subclasses definidas pelo atributo, quando este for satisfeito

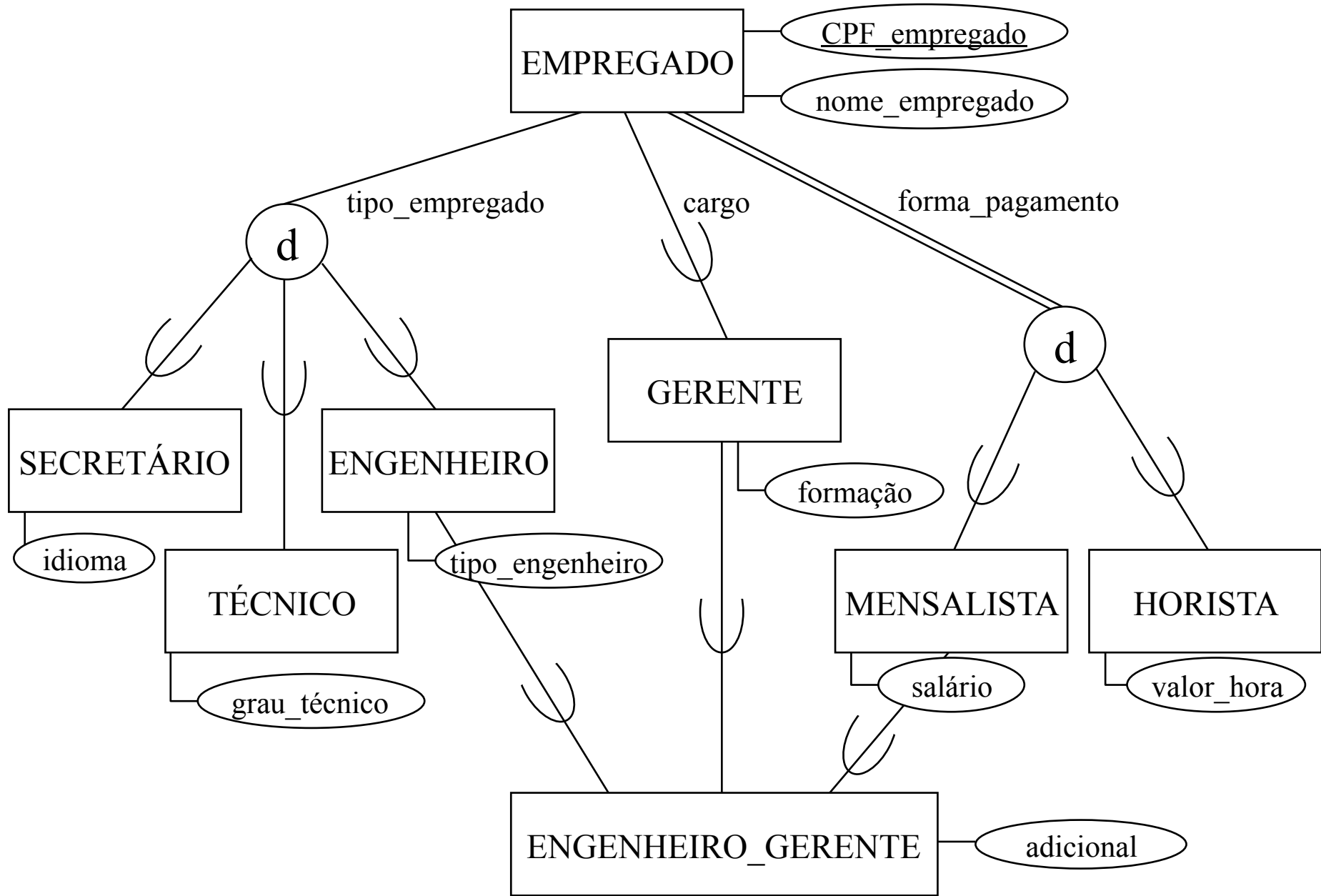
Observações

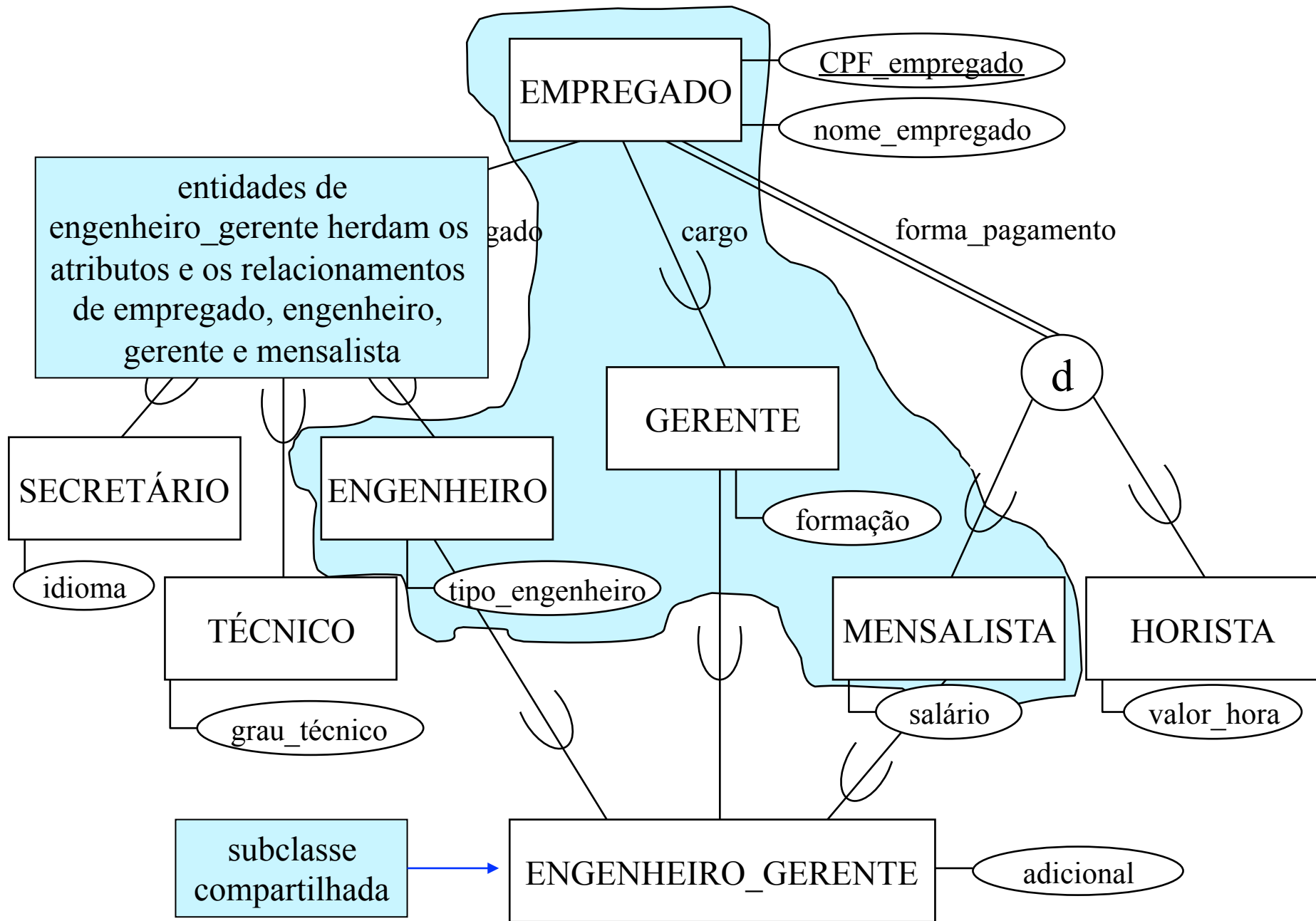
- ◆ Regras de inserção e remoção
 - se uma entidade for inserida em uma superclasse com especialização total, então ela deve ser necessariamente inserida em pelo menos uma das subclasses da especialização
 - ...



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

◆ Restrições

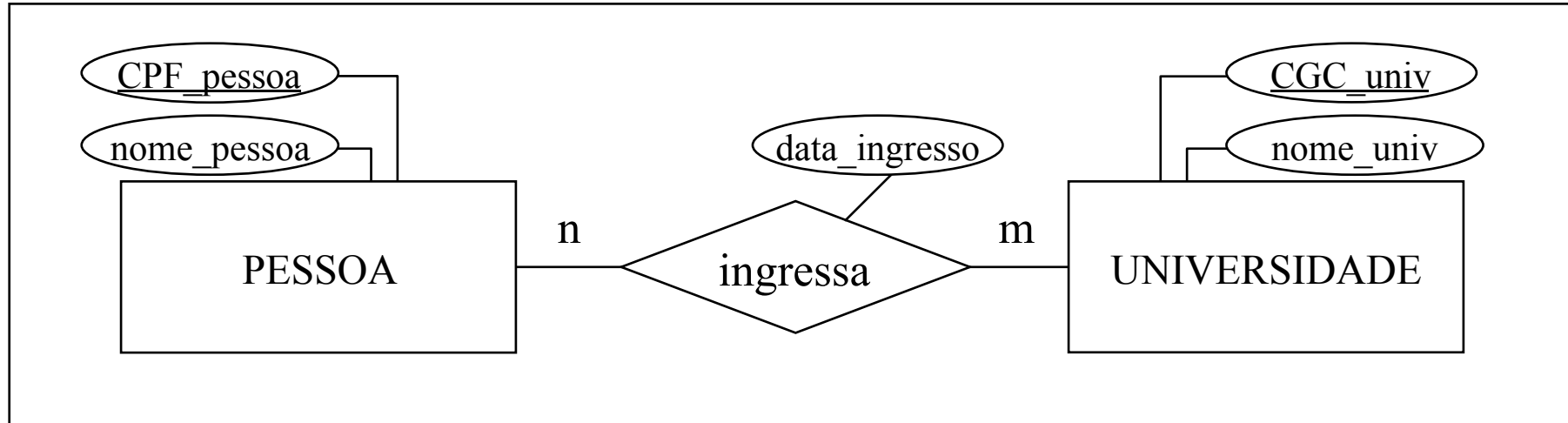
- alguns mecanismos de herança
 - ◆ não permitem herança múltipla
 - ◆ não permitem a especificação conjunta de herança múltipla e de diferentes predicados

Agregação

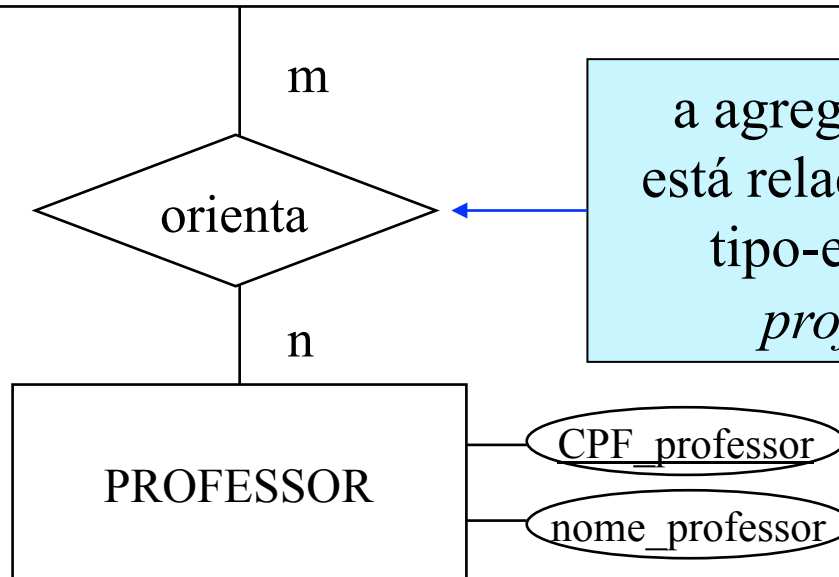
- ◆ Tipos-entidades agregados são representados como tipos-entidades comuns
- ◆ Pode englobar
 - dois tipos-entidades e um tipo-relacionamento
- ◆ Dados vistos em um nível mais baixo
 - atributos dos tipos-relacionamentos
 - chaves primárias dos tipos-entidades

Representação

ALUNO

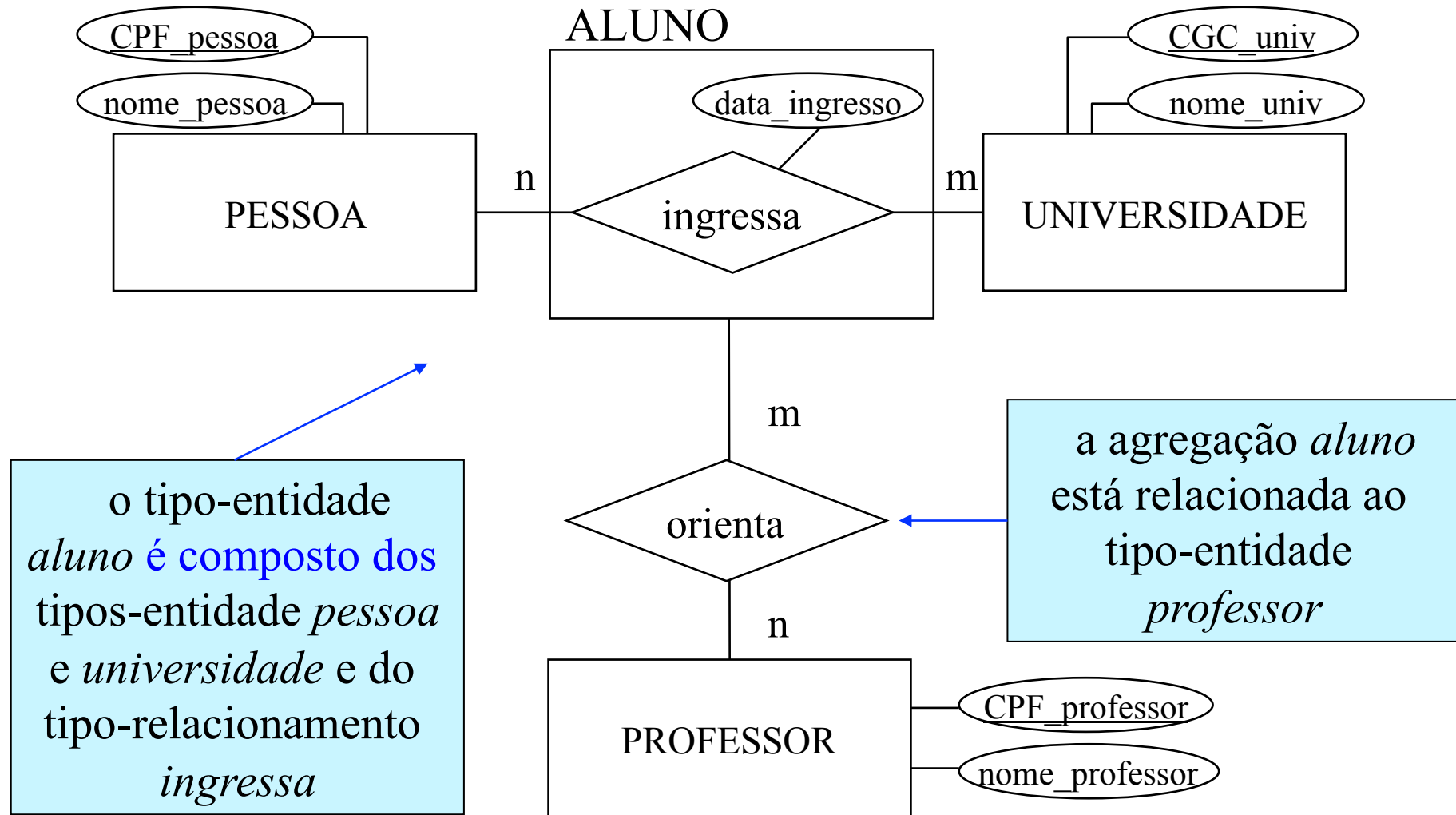


o tipo-entidade *aluno* é composto dos tipos-entidade *pessoa* e *universidade* e do tipo-relacionamento *ingressa*

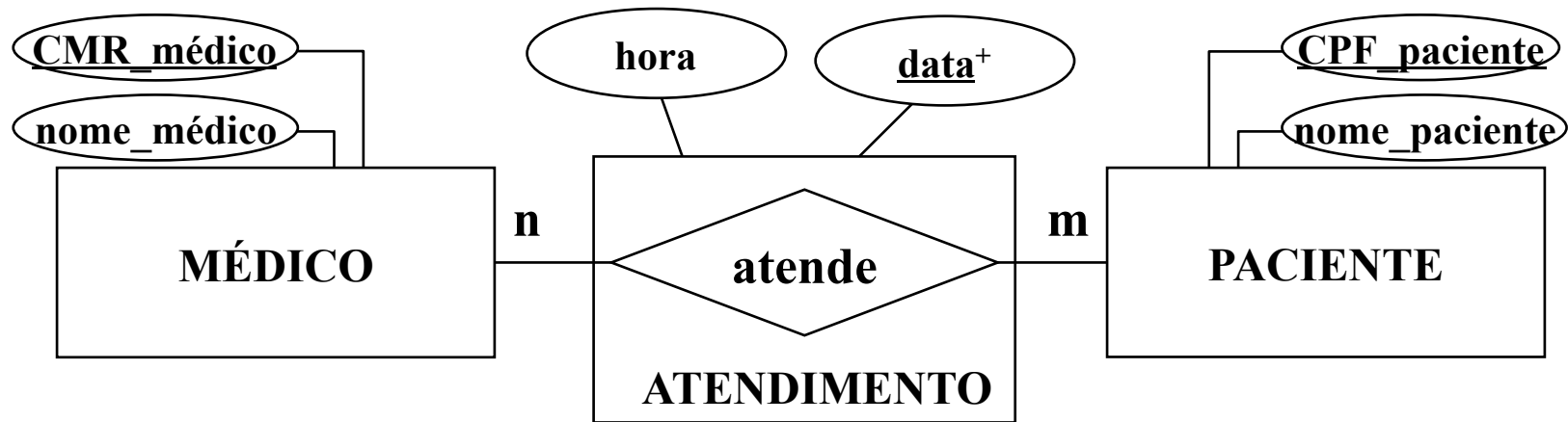


a agregação *aluno* está relacionada ao tipo-entidade *professor*

Representação Simplificada

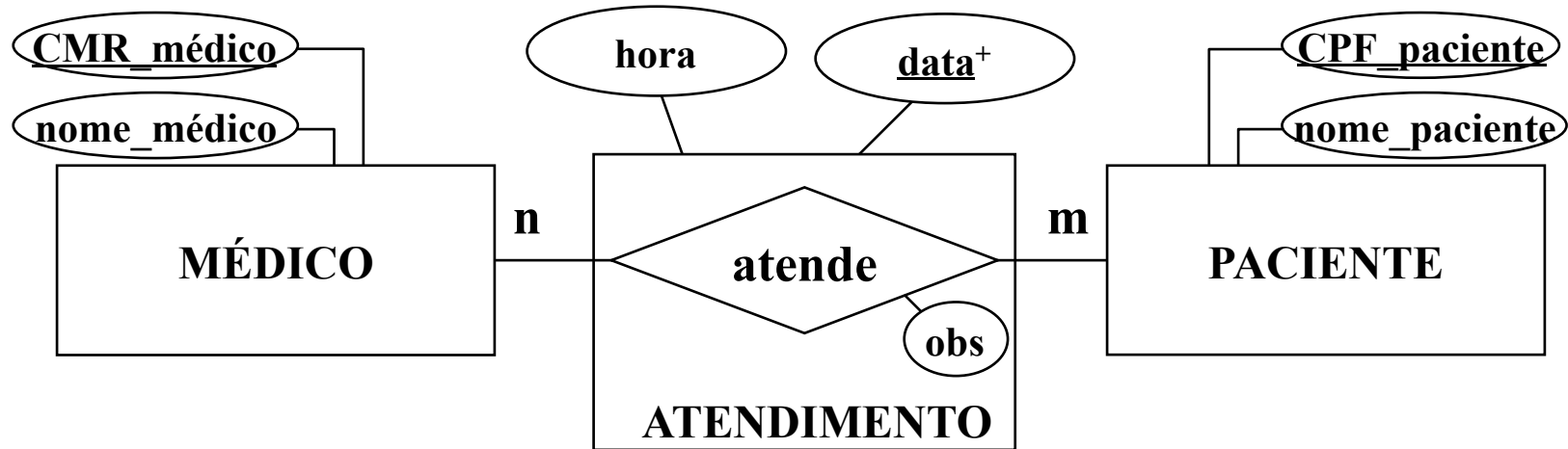


Agregação



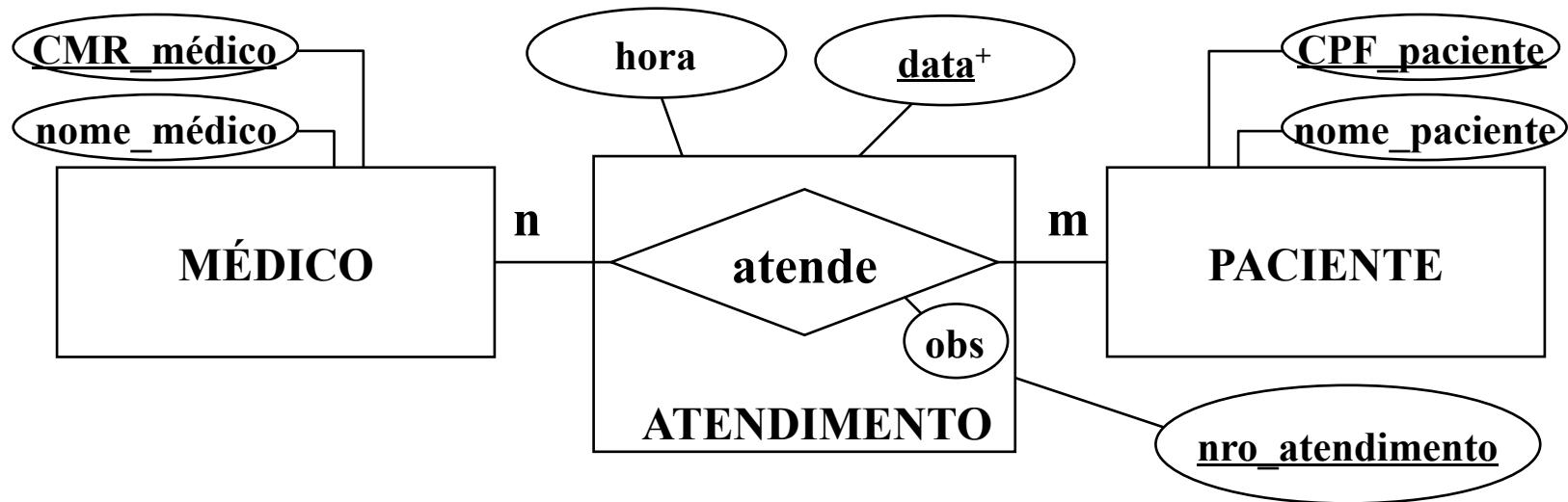
Enfocando a possibilidade do mesmo médico atender o mesmo paciente em diferentes datas

Agregação



Adicionando atributos também ao tipo-relacionamento

Agregação



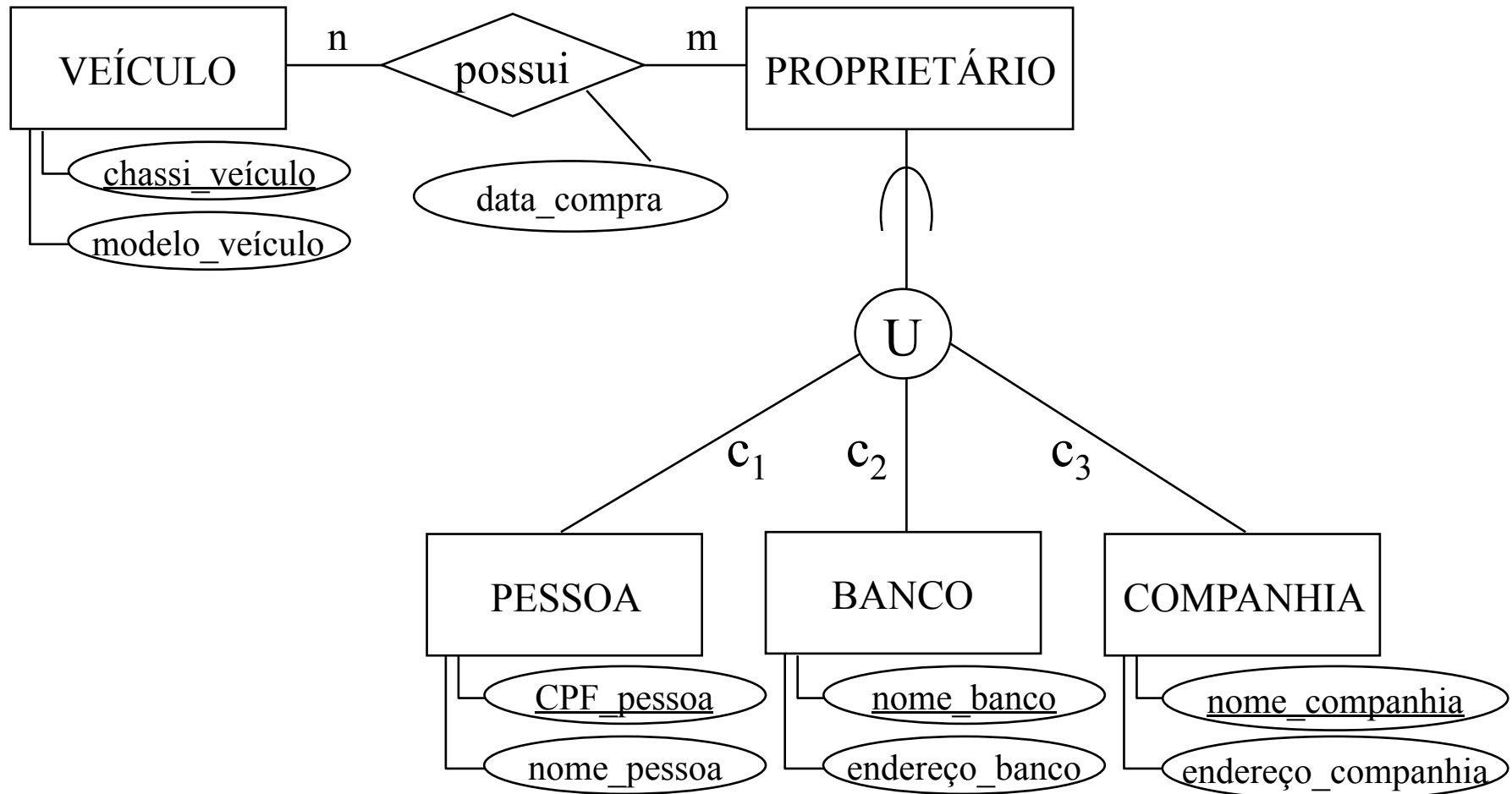
Adicionando uma chave primária à agregação

Categoria

- ◆ Representa uma coleção de objetos que é a **união** de objetos de diferentes tipos entidades
- ◆ Exemplo
 - superclasses: **pessoa**, **banco**, **companhia**
 - categoria: **proprietário**

subclasse da união de
pessoa, banco e companhia

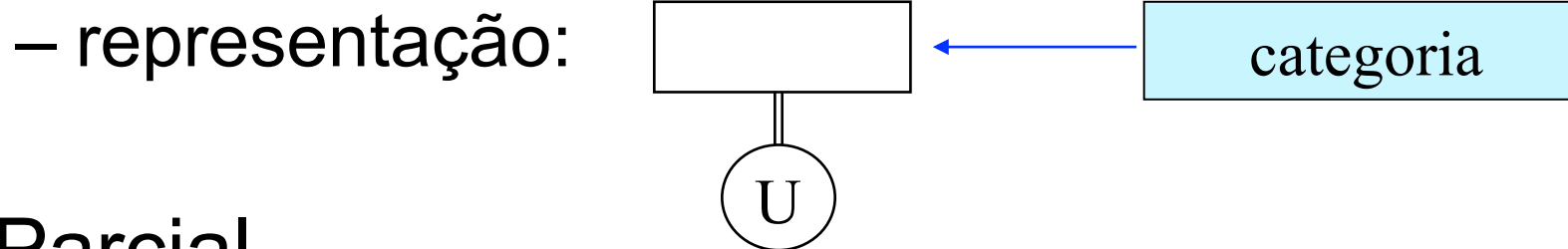
Representação



Restrição de Completude

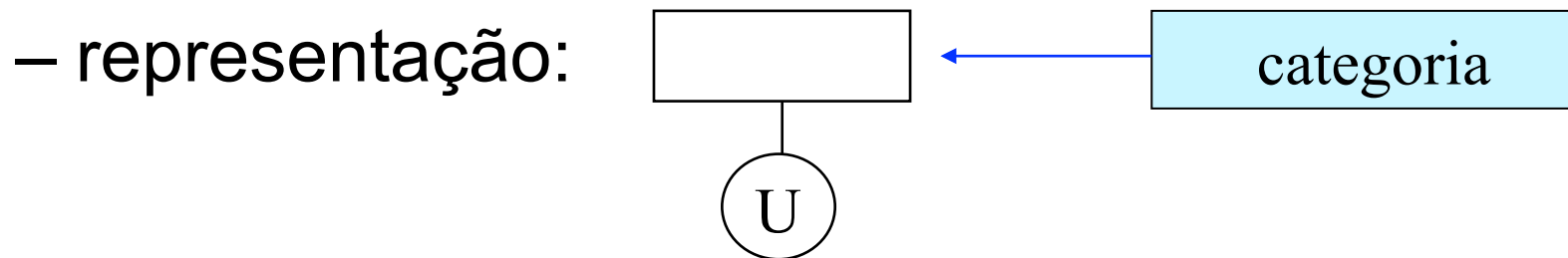
◆ Total

- cada entidade de cada superclasse deve ser membro da categoria



◆ Parcial

- uma entidade de uma superclasse pode não pertencer à categoria



Predicado

- ◆ Categoria definida pelo predicado
 - indica que as entidades das superclasses que são membro da categoria são determinadas por condições específicas
 - especificada apenas com a restrição de completude parcial
 - exemplos:
 - ◆ c_1 : idade > 18 anos
 - ◆ c_2 : fundos > R\$ 1.000.000,00
 - ◆ c_3 : ano_fundação > 1997

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
 - identificar as restrições que se aplicam sobre cada tipo-relacionamento
 - ◆ cardinalidade
 - ◆ participação
- ◆ Identificar tipo-entidade forte e tipo-entidade fraca

Projeto Lógico de BD

- ◆ Identificar agregações
 - cardinalidade
 - unicidade das tuplas

- ◆ Modelar hierarquias de generalização
 - identificar atributos e relacionamentos comuns
 - determinar as restrições de disjunção e de completude

Projeto Lógico de BD

- ◆ Modelar categorias
 - determinar restrições de completude
 - determinar os predicados, quando necessário

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