

# 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

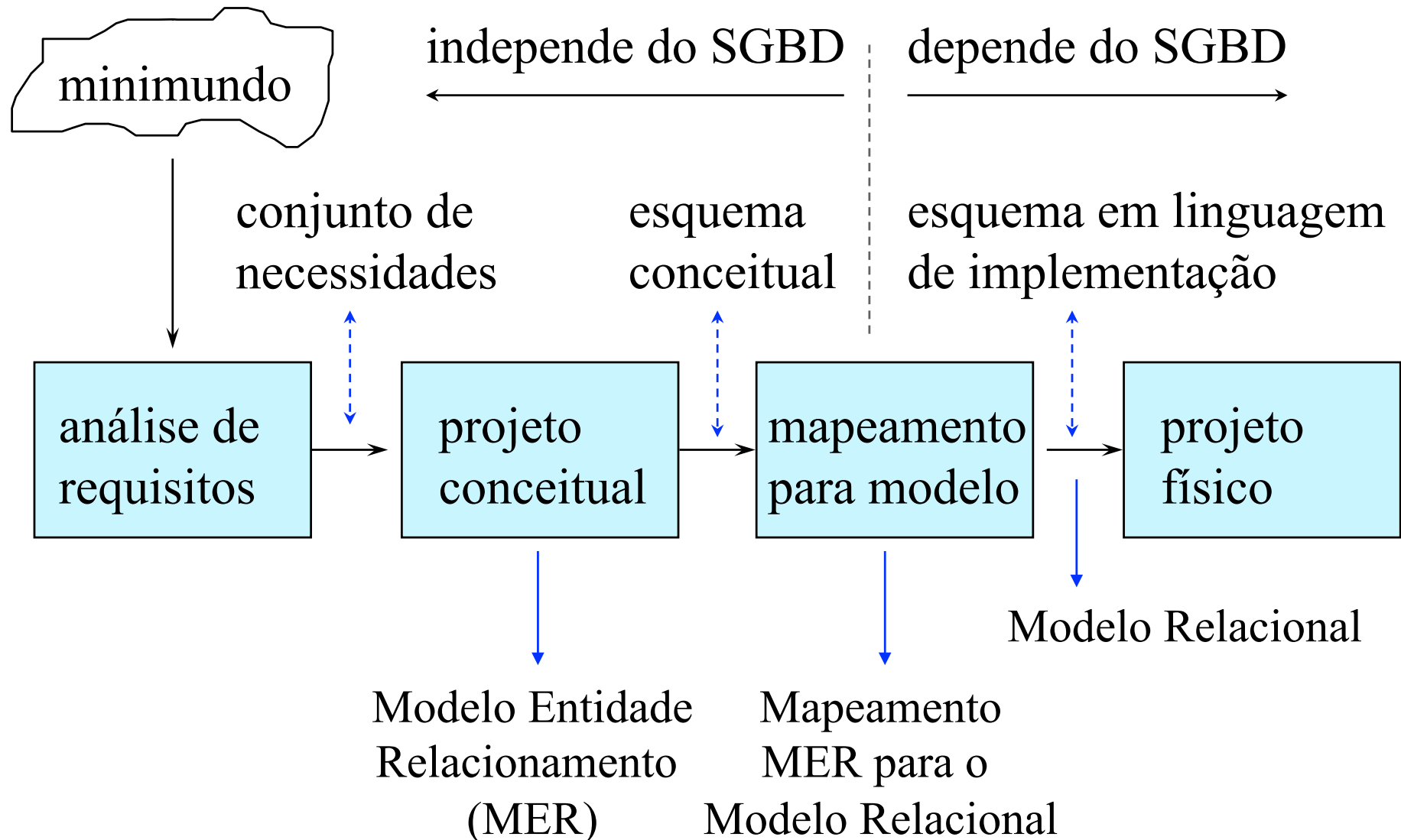
# Categorias de Modelos de Dados

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

# Modelo de Dados e o Projeto de BD



# Projeto de BD

- ◆ **Análise de requisitos**
  - entrevistas a usuários de BD
  - documentação do sistema
- ◆ **Projeto conceitual**
  - utiliza modelo de dados de alto nível
  - descreve de maneira concisa as necessidades dos usuários
  - inclui descrições detalhadas dos tipos de dados, relacionamentos e restrições
  - é fácil de ser entendido

# Projeto de BD

- ◆ Mapeamento para modelo
  - tradução do esquema conceitual em uma linguagem de alto nível para uma linguagem de implementação
- ◆ Projeto físico
  - especificação de estruturas internas de armazenamento
  - especificação das formas de organização de arquivos para BD

# Modelo Entidade-Relacionamento



# Modelo Entidade Relacionamento

## ◆ Características

- foi desenvolvido para facilitar o projeto lógico do BD
- permite a representação da estrutura lógica global do BD
- é um dos modelos de dados com maior capacidade semântica
- representa um problema como um conjunto de **entidades** e **relacionamentos** entre estas entidades

# Entidade

- ◆ Qualquer coisa do mundo real envolvida no problema
- ◆ Possui existência independente
- ◆ Pode ser um objeto com:
  - existência física: uma pessoa, um carro
  - existência conceitual: uma companhia, um emprego, um curso
- ◆ Descrita por propriedades particulares:  
**atributos**

# Atributos

- ◆ Caracterizam uma entidade ou um relacionamento

- exemplo: tipo-entidade `cliente`

- atributos: `nome_cliente`

- `endereço_cliente`

- `data_nascimento`

- ◆ Domínio de um atributo

- conjunto de valores possíveis para o atributo

- pode assumir valor nulo (i.e., null)

- exemplos: `nome_cliente: varchar(50)`

- `data_nascimento: date`

# Exemplos

Tipo-entidade cliente *atributos*: nome\_cliente  
endereço\_cliente  
data\_nascimento

– entidade  $e_1$ :

- ◆ nome\_cliente: Márcia
- ◆ endereço\_cliente: Rua X, 1
- ◆ data\_nascimento: 12/03/1970

– entidade  $e_2$ :

- ◆ nome\_cliente: Romualdo
- ◆ endereço\_cliente: Rua Floriano Peixoto, 10
- ◆ data\_nascimento: 10/10/1982

# Classificação dos Atributos

## ◆ *Simple versus* Compostos

### – atributo simples ou atômico

- ◆ não pode ser decomposto (dividido) em atributos mais básicos
- ◆ exemplo: **sexo**

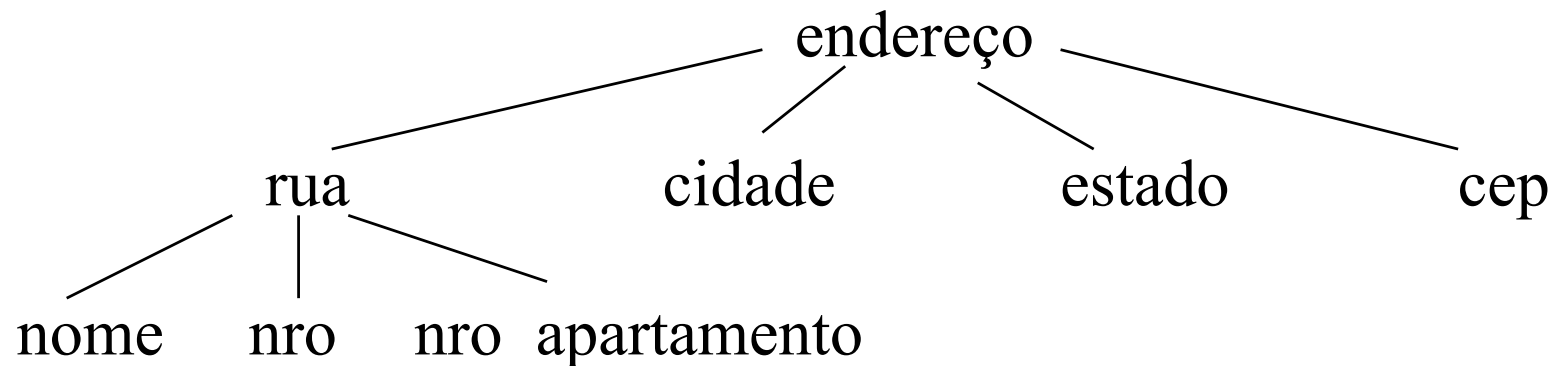
### – atributo composto

- ◆ pode ser decomposto (dividido) em vários outros atributos mais básicos
- ◆ possui como valor a concatenação dos valores dos atributos simples que o formam
- ◆ exemplo: atributo **endereço**, composto de *nome\_ rua*, *nro\_casa*, *complemento*, *nome\_bairro*, ...

# Atributos Simples x Compostos

## ◆ Atributos Compostos

- podem formar hierarquias



## ◆ Observação

- se nenhuma consulta será realizada sobre os atributos mais básicos de um atributo composto, então o atributo composto pode ser armazenado no BD como um atributo simples

# Classificação dos Atributos

## ◆ Monovalorados *versus* Multivalorados

### – atributo monovalorado

- ◆ possui um único valor para cada entidade
- ◆ exemplo: [idade](#)

### – atributo multivalorado

- ◆ possui múltiplos valores para cada entidade
- ◆ exemplo: atributo [telefone](#)

valores: *222-0000, 222-0001, ...*

- ◆ pode possuir limites inferior/superior com relação à multiplicidade dos valores assumidos
- ◆ exemplo:  $nro\_min = 0, nro\_max = 3$

# Classificação dos Atributos

## ◆ Armazenados *versus* Derivados

– atributo armazenado

◆ está realmente armazenado no BD

– atributo derivado

◆ pode ser determinado através de outros atributos ou através de entidades relacionadas

◆ exemplos: **idade** = data\_atual - data\_nascimento

**nro\_empregados** = soma das entidades  
empregado para um  
departamento

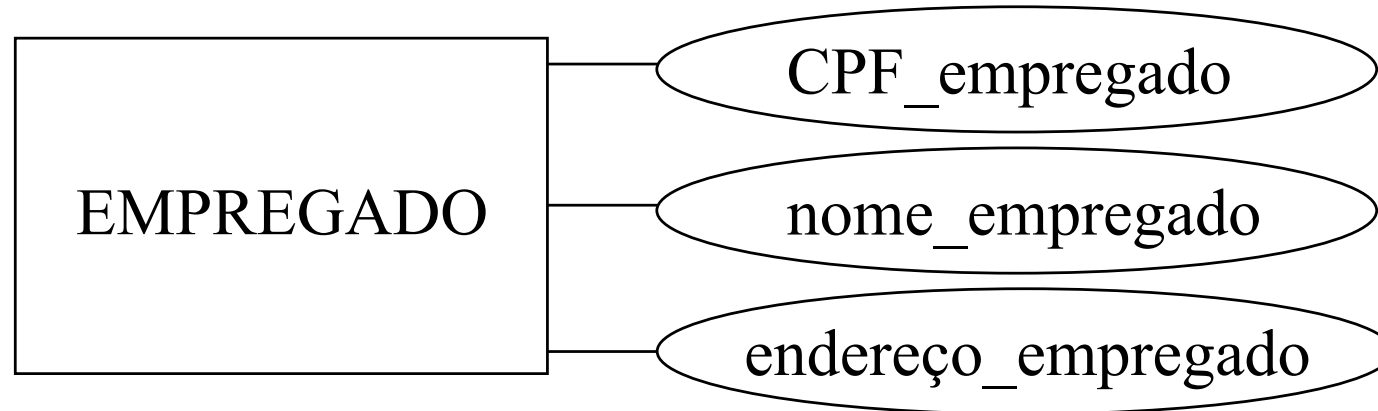
◆ pode ou não ser armazenado no BD



# Tipo-Entidade

- ◆ Conjunto de entidades do mesmo tipo
- ◆ Descrito por um nome e uma lista de atributos
- ◆ Entidades de um tipo-entidade
  - compartilham os mesmos atributos
  - possuem seus próprios valores para cada atributo

# Representação



tipo-entidade: empregado

entidades (instâncias)

CPF\_empregado: 192876433-0  
nome\_empregado: Maria  
endereço\_empregado: Rua X

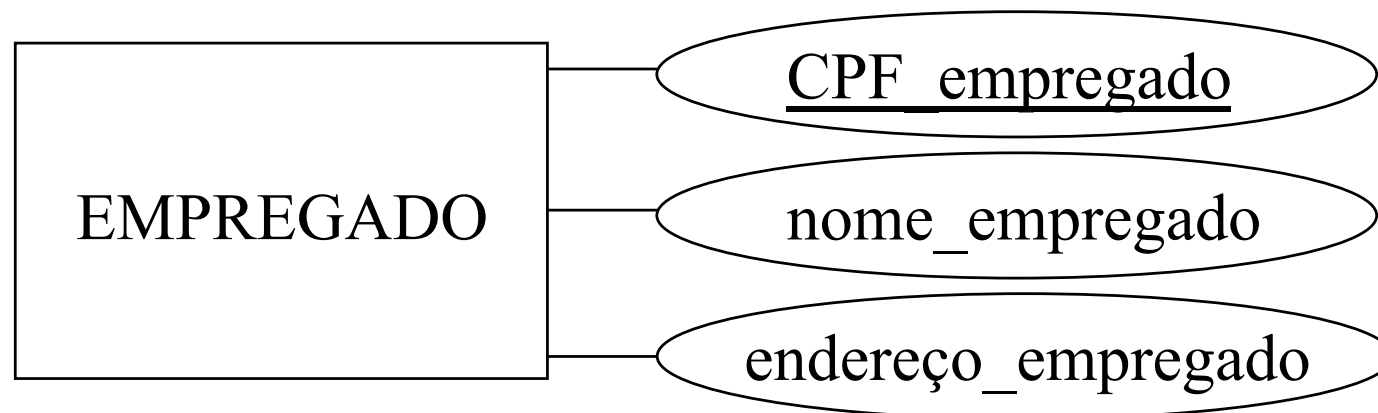
CPF\_empregado: 150150150-0  
nome\_empregado: José  
endereço\_empregado: Rua Y

# Restrição de Chave

## ◆ Chave primária

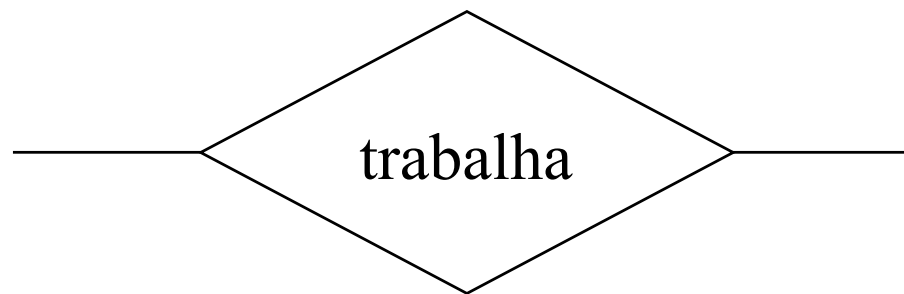
- conjunto mínimo de atributos que identificam de maneira única uma entidade
- escolhida pelo projetista do BD como o principal meio de identificação de um tipo-entidade

## ◆ Exemplo:



# Relacionamento e Tipo-relacionamento

- ◆ Relacionamento
  - associação entre entidades
- ◆ Tipo-relacionamento
  - conjunto de relacionamentos do mesmo tipo
- ◆ Exemplo: pessoa trabalha para empresa



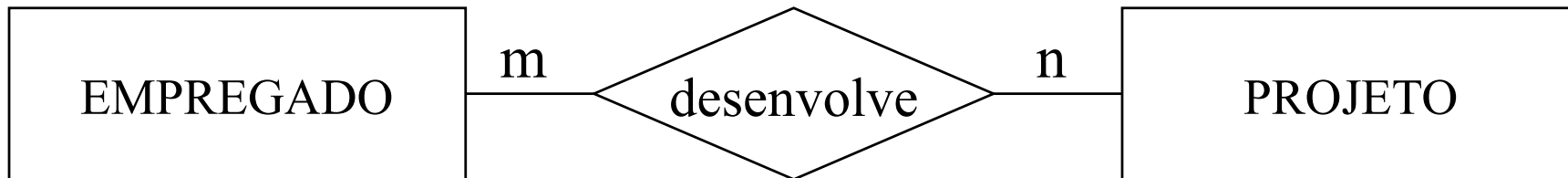
# Restrições nos Tipos-Relacionamento

- ◆ Limitam as combinações possíveis de entidades que podem participar dos tipos-relacionamento
- ◆ Derivadas do minimundo sendo analisado
- ◆ Restrições estruturais
  - cardinalidade
  - participação

# Restrição de Cardinalidade

- ◆ Determina o número de entidades às quais outras entidades podem ser associadas através de um relacionamento
- ◆ Cardinalidades
  - um-para-um ( 1 : 1 )
  - um-para-muitos ( 1 : n )
  - muitos-para-um ( n : 1 )
  - muitos-para-muitos ( m : n )

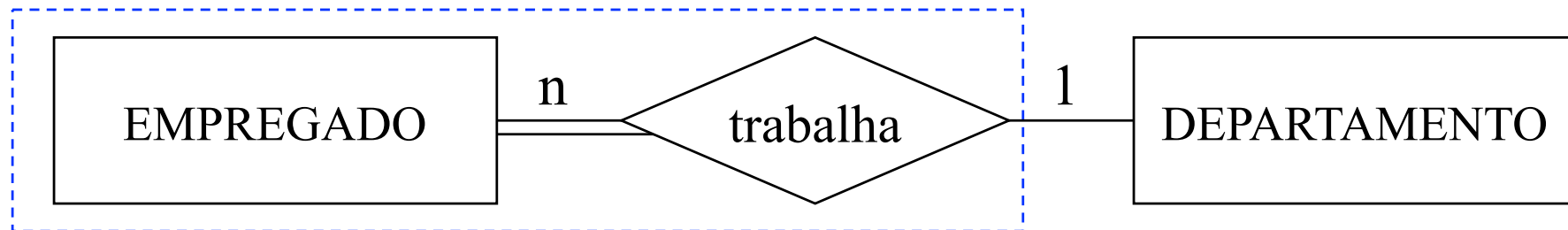
# Restrição de Cardinalidade



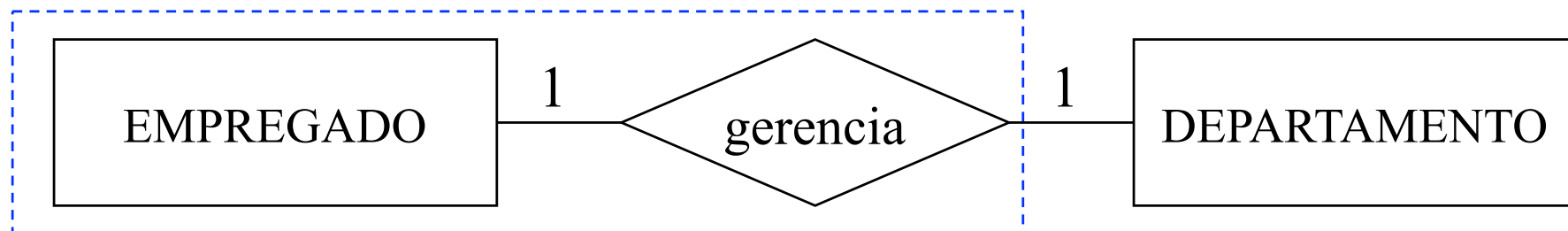
# Restrição de Participação

- ◆ Determina se a existência de uma entidade depende ou não do fato dela participar de um relacionamento
- ◆ Tipos de participação

– total



– parcial

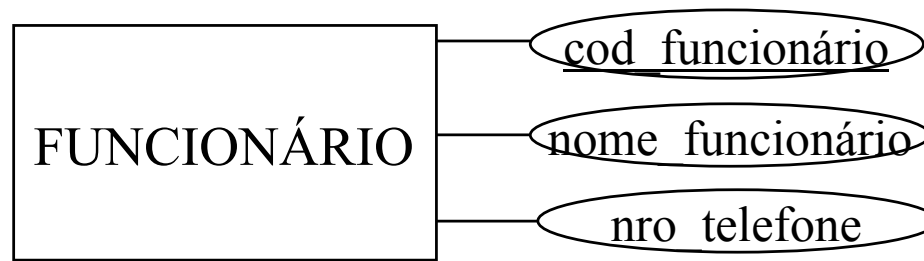




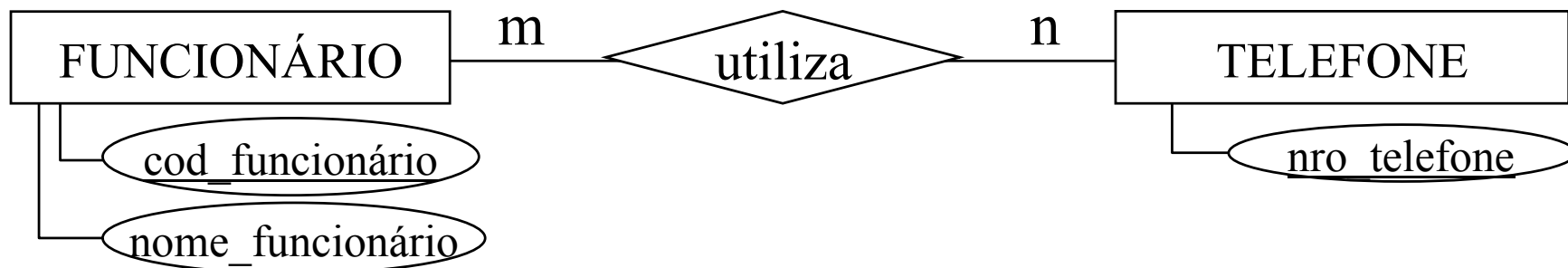
# Exercício

- ◆ Diferencie semanticamente as duas modelagens representadas a seguir:

a)

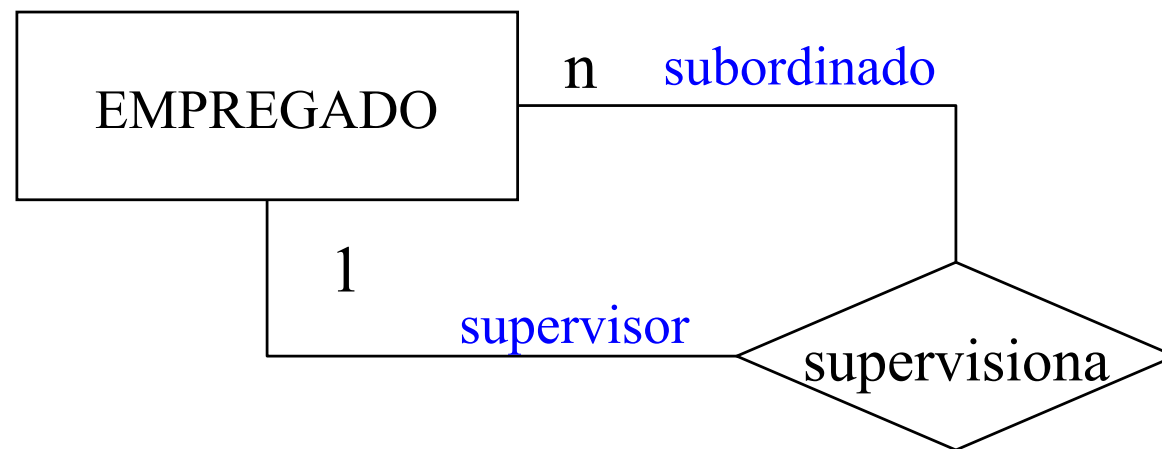


b)



# Grau de Tipos-Relacionamento

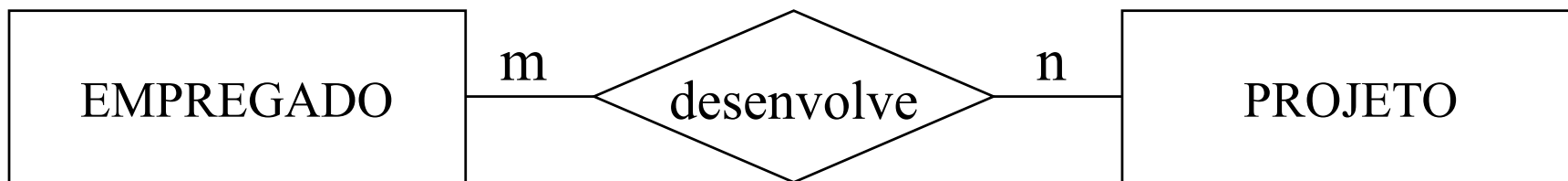
- ◆ Grau de um tipo-relacionamento
  - número de tipos-entidade participantes
- ◆ Unário (ou recursivo)
  - relaciona um tipo-entidade com ela mesma
  - indicado utilizar nomes de papéis



# Tipos de Relacionamentos

## ◆ Binário

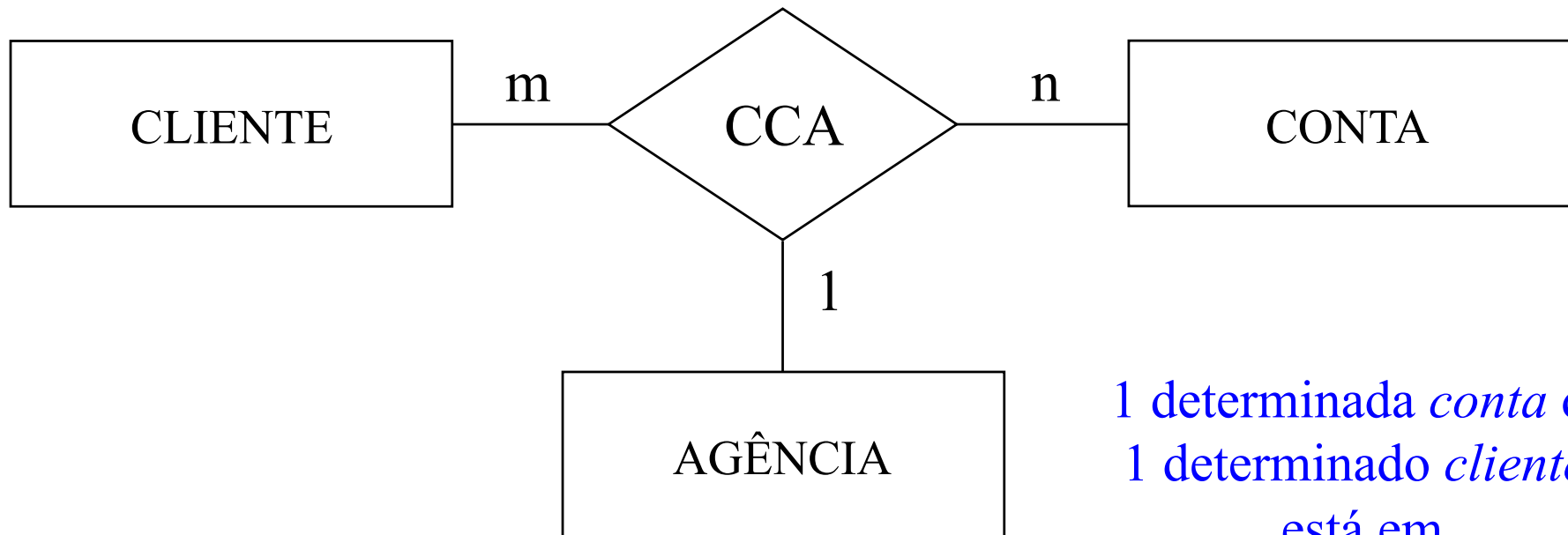
- relaciona um tipo-entidade a outro tipo-entidade
- grau de relacionamento mais utilizado



# Tipos de Relacionamentos

## ◆ Ternário

– relaciona três tipos-entidade



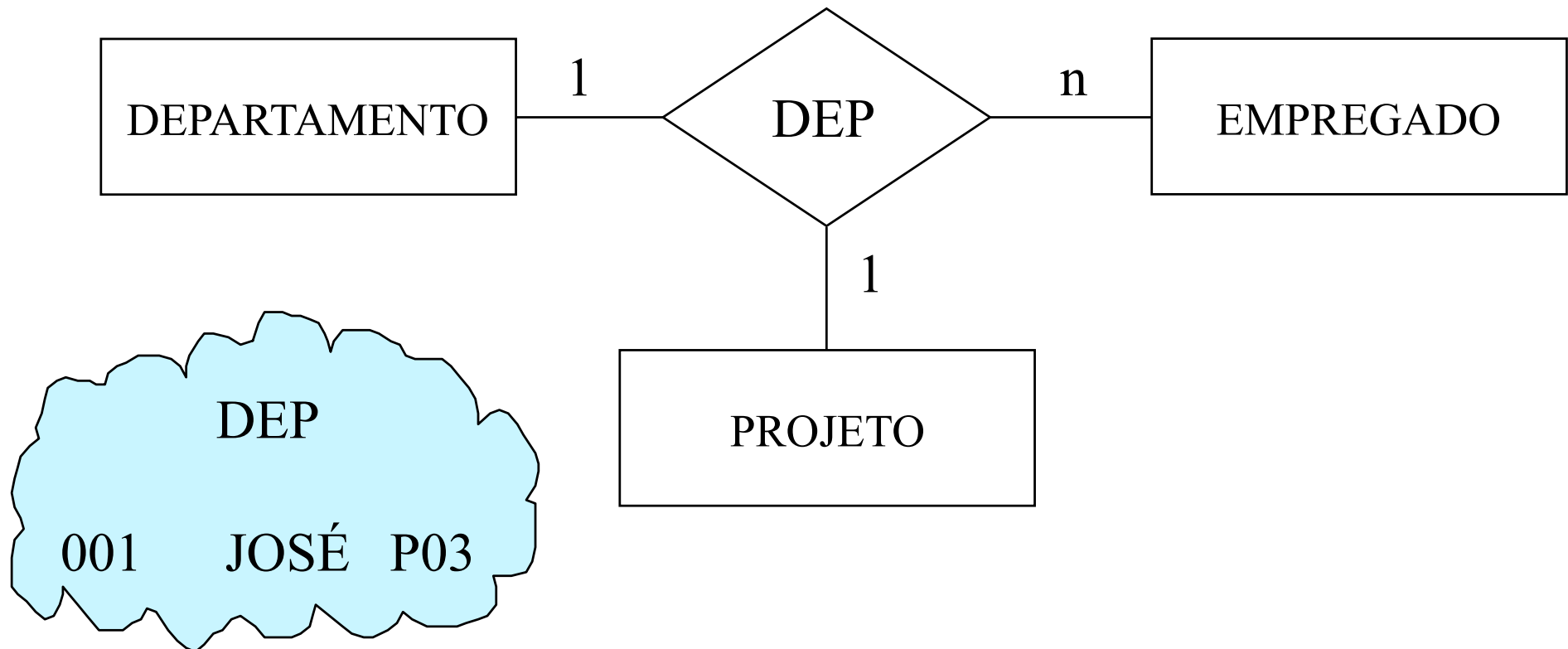
1 determinada *conta* de  
1 determinado *cliente*  
está em  
1 determinada *agência*

# Relacionamentos Ternários

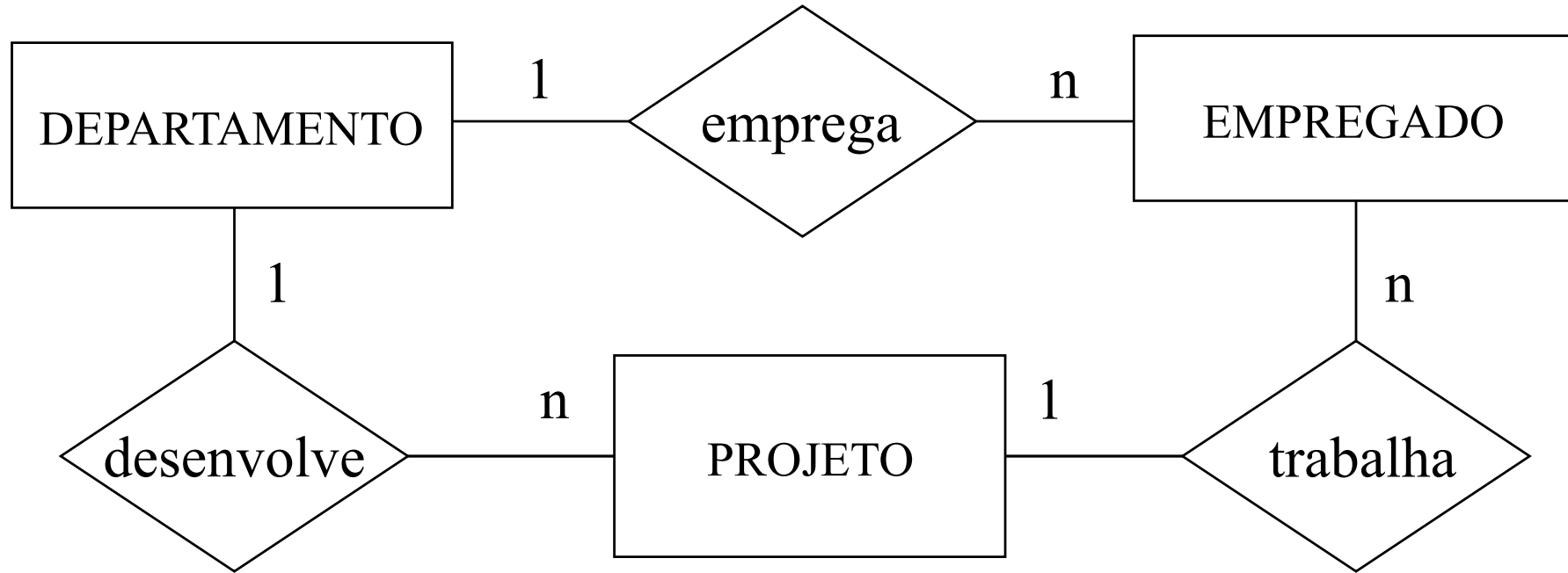
- ◆ Regra para a determinação das multiplicidades:
  - fixa-se dois elementos (dois tipos-entidade)
  - verifica-se quantos elementos do outro tipo-entidade podem surgir com relação a um elemento de cada tipo-entidade fixada
  - se a quantidade for indeterminada ou variável  
então considera-se **n**  
senão considera-se **1**

# Relacionamentos Ternários

- ♦ Um relacionamento ternário em geral representa informações diferentes das dos três tipos-relacionamento binários



# Relacionamentos Ternários



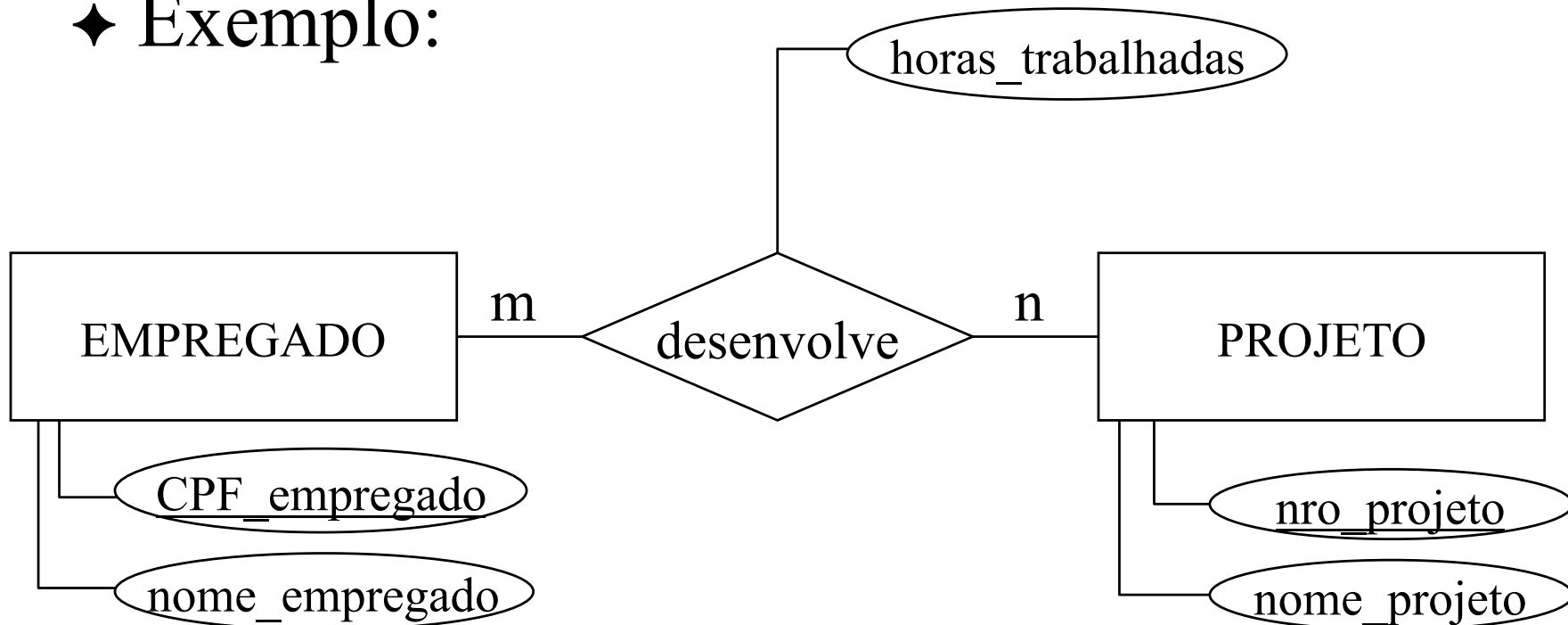
emprega	trabalha	desenvolve
001 JOSÉ	JOSÉ P03	P03 004

◆ **Decisão: semântica do problema**

# Atributo de Tipo-Relacionamento

◆ Determinado pela combinação das entidades participantes em uma instância do relacionamento

◆ Exemplo:





# Exercício

- ◆ Dados os tipos-entidade **curso** e **disciplina**
  - atributos de **curso**: código\_curso, nome\_curso
  - atributos de **disciplina**: código\_disciplina, nome\_disciplina, carga\_horária
- ◆ Faça duas diferentes modelagens, de acordo com as especificações a seguir
  - uma disciplina é obrigatória ou optativa, independentemente do curso (**modelagem 1**)
  - uma disciplina pode ser obrigatória para um curso e optativa para outro curso (**modelagem 2**)

# Tipo-Entidade Fraca

- ◆ Entidades de um tipo-entidade fraca:
  - não podem ser distinguíveis porque a combinação dos valores de seus atributos pode ser idêntica
  - são identificadas através da relação que possuem com entidades pertencentes a tipos-entidade forte
- ◆ Representa dependência de existência
  - um tipo-entidade fraca sempre tem uma restrição de participação total com respeito ao relacionamento identificador

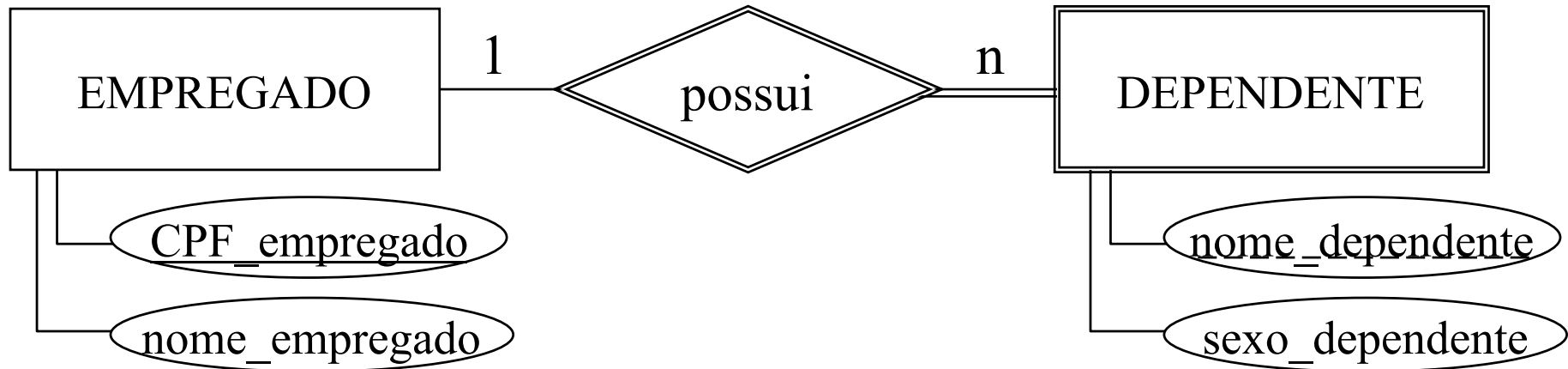
# Dependência de Existência

- ◆ Se uma entidade  $x$  depende da existência de uma entidade  $y$ , então:
  - $x$  : entidade subordinada
  - $y$  : entidade dominante
- ◆ Se  $y$  for removida  
então  $x$  também deve ser removida
- ◆ Exemplos:
  - empregado e dependente
  - conta e transações

# Tipo-Entidade Fraca

- ◆ Entidade forte
  - tem chave primária
- ◆ Entidade fraca
  - possui somente uma **chave parcial**
- ◆ Chave primária de uma entidade fraca
  - chave primária da entidade forte correspondente
  - +
  - chave parcial da entidade fraca

# Representação

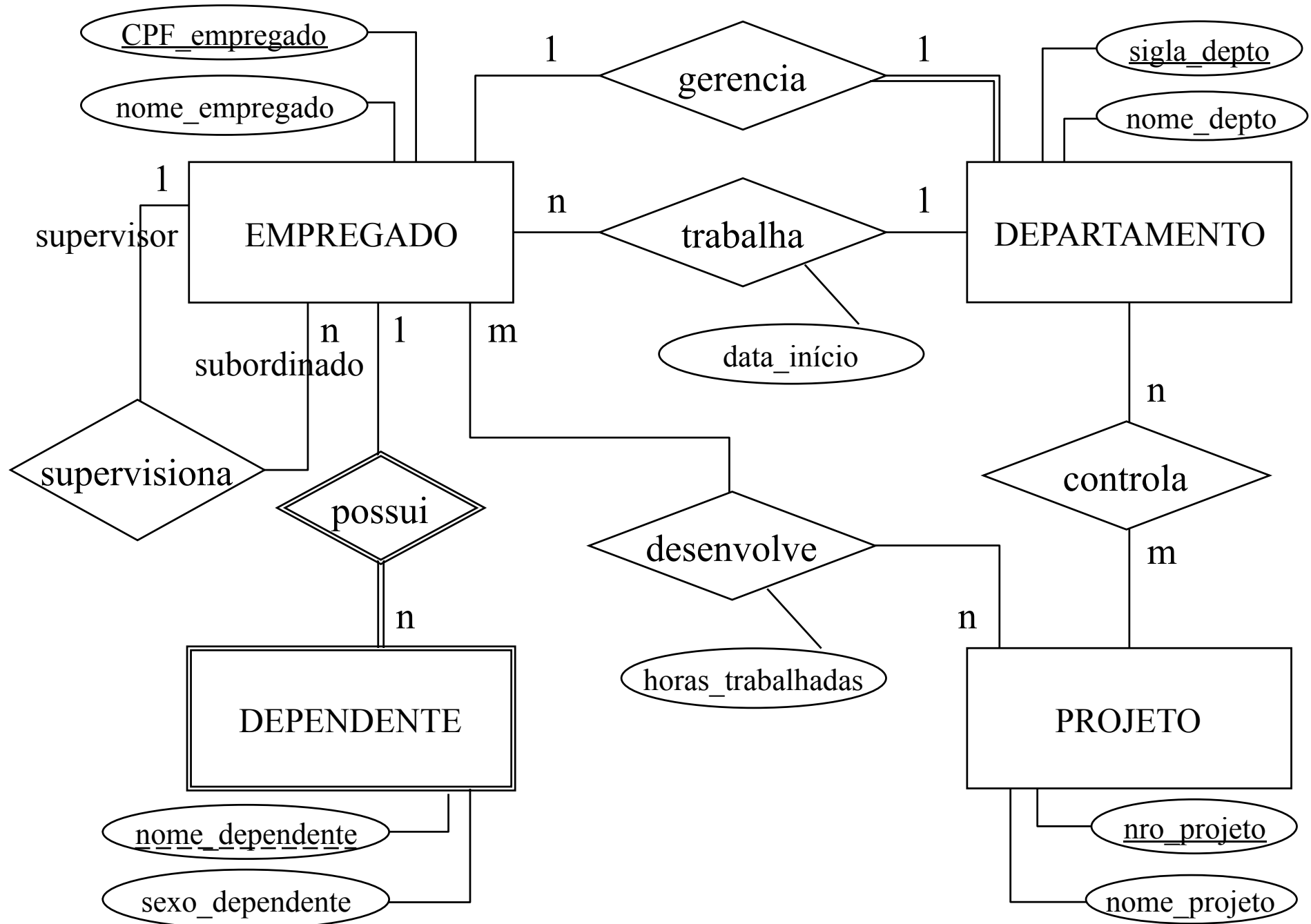


chave primária de empregado: CPF\_empregado

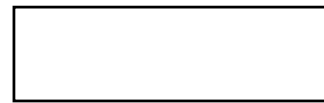
chave primária de dependente: CPF\_empregado + nome\_dependente

entidade fraca (subordinada): **dependente**

entidade forte (dominante): **empregado**



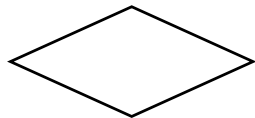
# Resumo da Notação



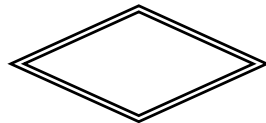
tipo entidade forte



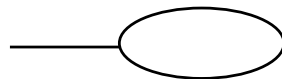
tipo entidade fraca



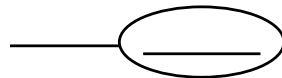
tipo relacionamento



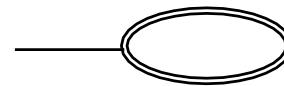
identificador de relacionamento



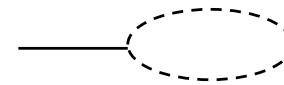
atributo



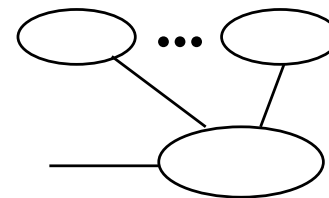
atributo chave



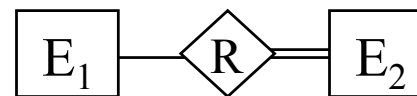
atributo multivalorado



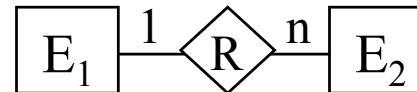
atributo derivado



atributo composto



participação total de  $E_2$  em R



cardinalidade 1:N

# Projeto Lógico de BD

- ◆ Identificar tipos-entidade e atributos
  - tipos-entidade possuem informações descritivas, atributos não
  - atributos devem ser relacionados às entidades que eles descrevem
- ◆ Identificar chaves primárias



# Projeto Lógico de BD

- ◆ Identificar tipos-relacionamento e seus atributos
  - determinar o grau dos tipos-relacionamento
    - ◆ definir tipos-relacionamento ternários cuidadosamente
  - identificar as restrições que se aplicam sobre cada tipo-relacionamento
    - ◆ cardinalidade
    - ◆ participação

# Projeto Lógico de BD

- ◆ Identificar tipo-entidade forte e tipo-entidade fraca

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

# Exercícios

- ◆ Fazer os exercícios da lista de exercícios “Modelo Entidade Relacionamento”
  - 1. Controle Acadêmico I
  - 2. Sistema de Controle Bancário