

# **Construindo modelos ER**

## Capítulo 3

# Construindo modelos ER - Temário

1. Conselhos práticos
2. Heurísticas
3. Notações alternativas
4. Processo de modelagem e alternativas

# Propriedades de modelos ER

- Modelo ER é um modelo **formal**
- Poder de expressão é **limitado**
- **Equivalência** entre modelos

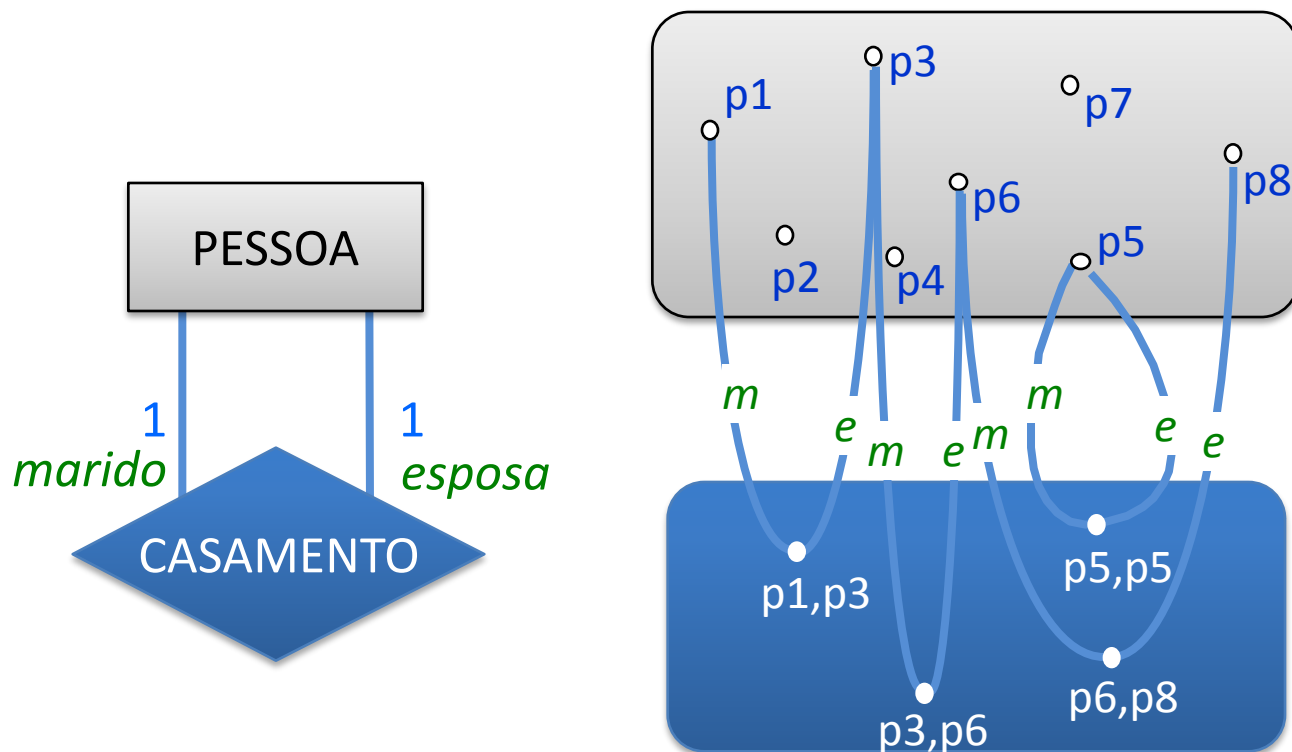
# Modelo ER é um modelo formal

- Modelo **preciso**, **não ambíguo** .
- Diferentes leitores de um mesmo modelo ER devem sempre entender exatamente o mesmo.
- DER pode ser usado como entrada a uma ferramenta CASE.
- Fundamental:
  - todos os envolvidos devem estar treinados na sua perfeita compreensão.
- Risco: sub-utilização .

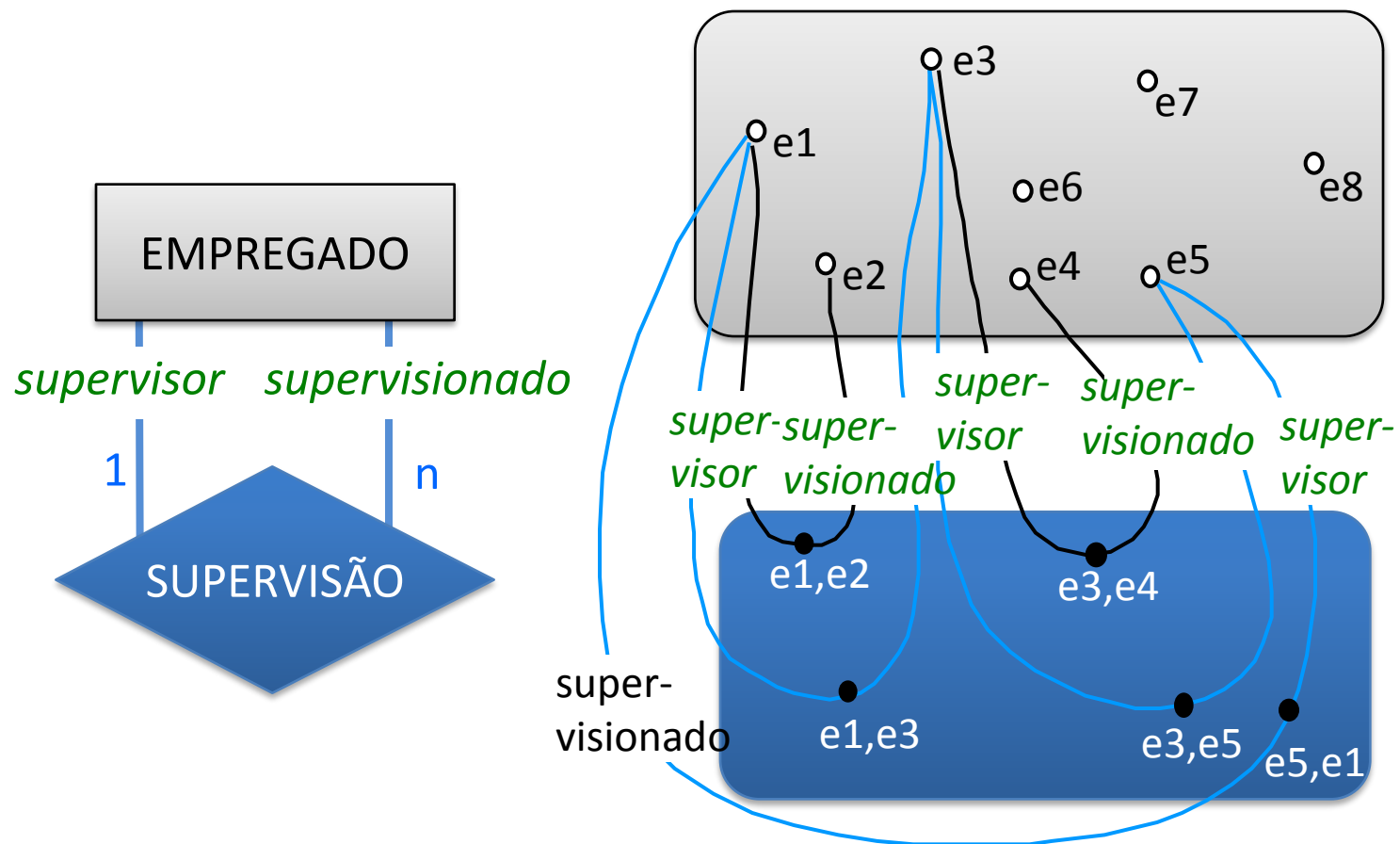
# Poder de expressão limitado

- Modelo ER apresenta apenas algumas propriedades de um banco de dados:
  - Foi concebido para o projeto da estrutura de um BD relacional.
- Poder de expressão limitado para expressar restrições de integridade genéricas (regras de negócio).

# Poder de expressão limitado - exemplo



# Poder de expressão limitado - exemplo



# Eqüivalência entre modelos

- Dois modelos ER diferentes podem ser equivalentes.
- Modelos equivalentes:
  - expressam o mesmo,
  - modelam a mesma realidade.
- Para fins de projeto de BD, dois modelos ER são equivalentes quando:
  - geram o mesmo esquema de BD.
- Considerar um conjunto de regras de tradução de modelos ER para modelos lógicos de BD.



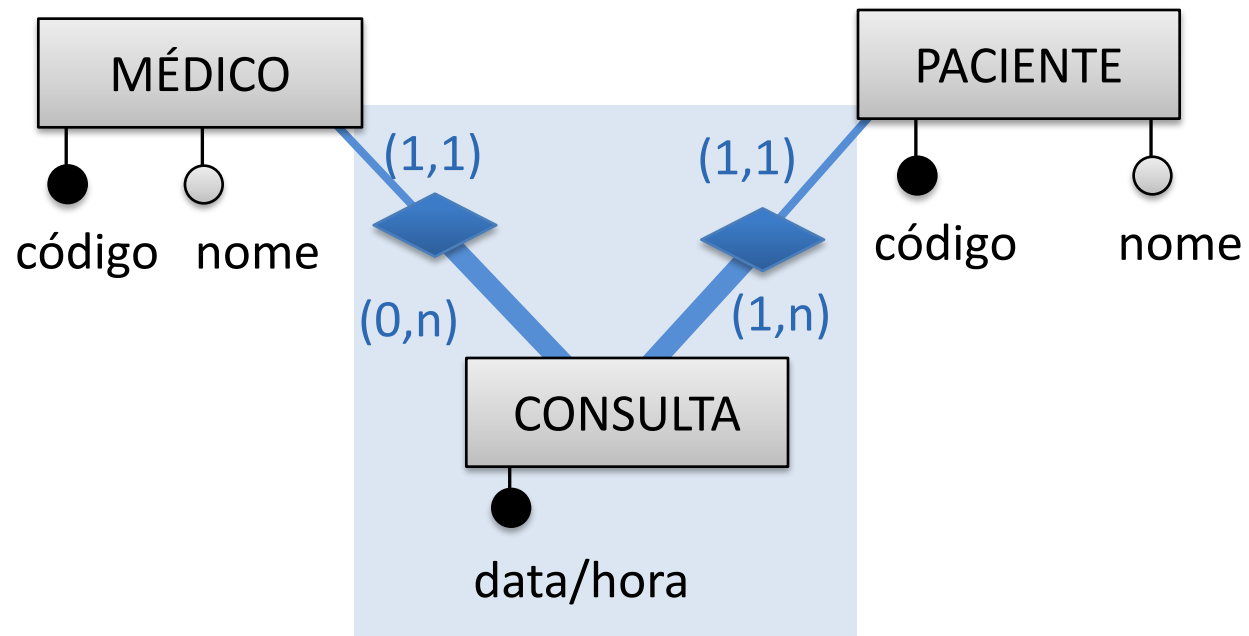
# Exemplo de equivalência entre modelos

a) CONSULTA como relacionamento n:n



# Modelo equivalente

## b) CONSULTA como entidade



# Transformação de relacionamento n:n em entidade (1)

1. O relacionamento n:n é representado como uma entidade.
2. A entidade criada é relacionada às entidades que originalmente participavam do relacionamento.
3. A entidade criada tem como identificador:
  - as entidades que originalmente participavam do relacionamento,
  - os atributos que eram identificadores do relacionamento original (caso o relacionamento original tivesse atributos identificadores).

## Transformação de relacionamento n:n em entidade (2)

4. Nos relacionamentos de que participa, a cardinalidade da entidade criada é sempre (1,1).
5. As cardinalidades das entidades que eram originalmente associadas pelo relacionamento são transcritas ao novo modelo conforme mostrado na figura.

# Modelo ER sem relacionamento n:n

- Relacionamento n:n pode ser transformado em entidade.
- É possível construir modelos sem relacionamentos n:n.
- Há variantes da abordagem ER, que
  - excluem o uso de relacionamentos n:n, ou
  - excluem apenas o uso de relacionamentos n:n com atributos
- Exemplo:
  - várias abordagens baseadas na Engenharia de Informações (ver adiante)

# Identificando construções

- Determinação da **construção** da abordagem ER (entidade, relacionamento,...) que **será usada para modelar** um **objeto** de uma realidade:
  - Não pode ser feita **através** da **observação do objeto isoladamente**.
  - É necessário conhecer o **contexto** (modelo dentro do qual o objeto aparece).

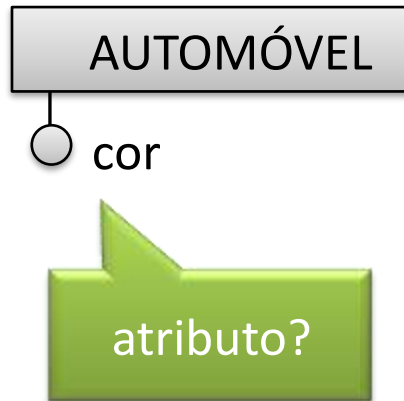
# Identificando construções

## Recomendação geral

- Decisão por uma construção para a modelagem de um objeto está **sujeita a alteração durante a modelagem**.
- Não despende um tempo excessivo em longas discussões sobre como modelar um objeto.
- Desenvolvimento do modelo e o aprendizado sobre a realidade irão **refinando e aperfeiçoando** o modelo.

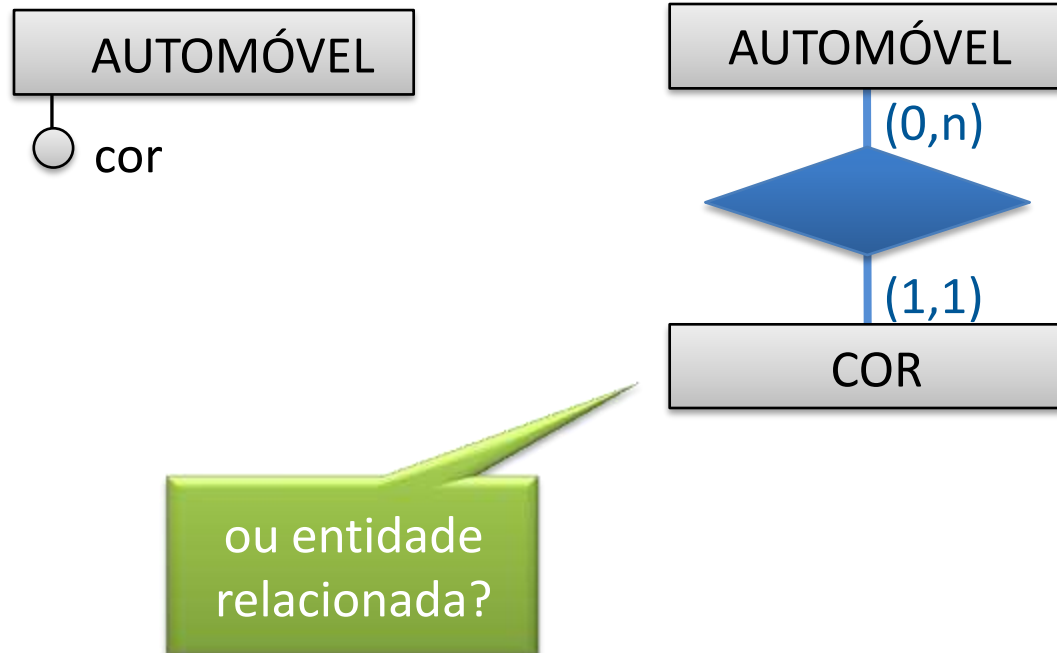
# Atributo versus entidade relacionada

Como deve ser modelada a cor de um automóvel?





# Atributo versus entidade relacionada



# Atributo versus entidade relacionada critérios (1)

- Objeto está relacionado com outros objetos:
  - deve ser modelado como entidade.
- Caso contrário:
  - pode ser modelado como atributo.

## Atributo versus entidade relacionada critérios (2)

- Conjunto de valores de um determinado objeto é fixo (domínio **fixo**):
  - pode ser modelado como **atributo**.
- Existem transações no sistema que alteram o conjunto de valores do objeto (domínio **variável**):
  - **não** deve ser modelado como **atributo**.

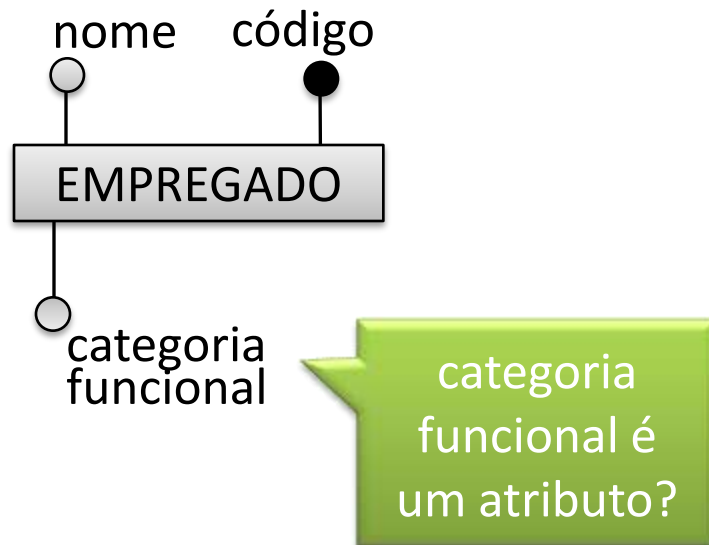
# Atributo versus generalização/especialização

- Questão:
  - modelar um determinado objeto  
(exemplo, a categoria funcional de cada empregado de uma empresa):
    - como **atributo**?  
(categoria funcional como atributo da entidade EMPREGADO)
    - ou como uma **especialização**?  
(cada categoria funcional corresponde a uma especialização da entidade empregado)

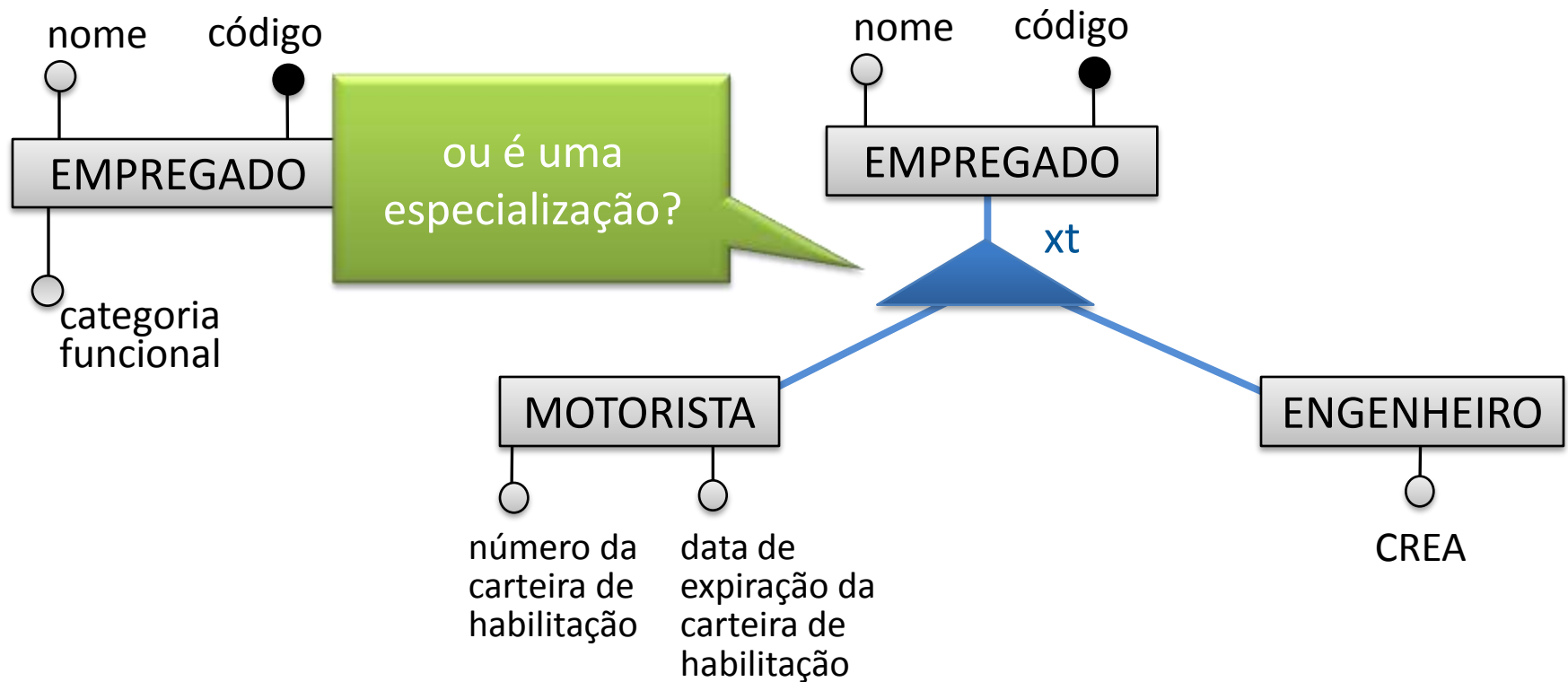
# Atributo versus generalização/especialização

- Especialização deve ser usada quando as classes especializadas de entidades possuem propriedades particulares:
  - atributos
  - relacionamentos
  - generalizações/especializações

# Atributo versus generalização/especialização



# Atributo versus generalização/especialização

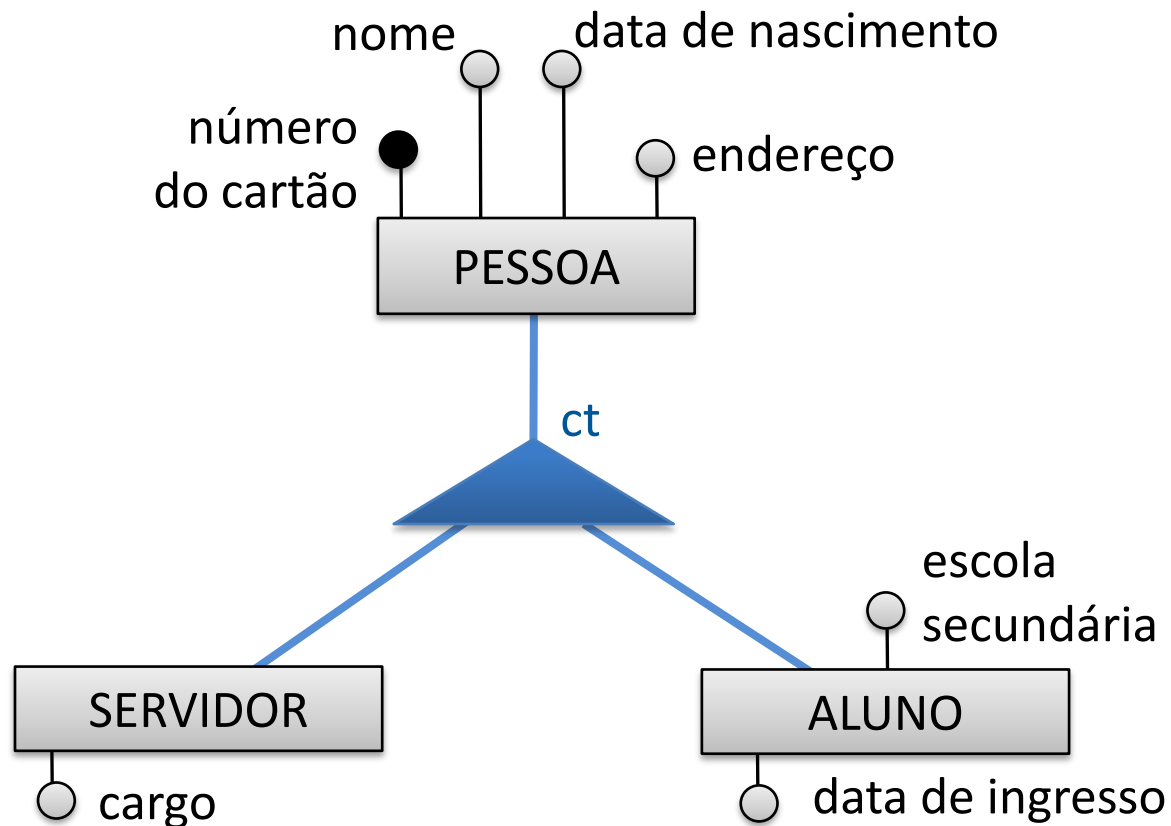


# Entidade versus especialização

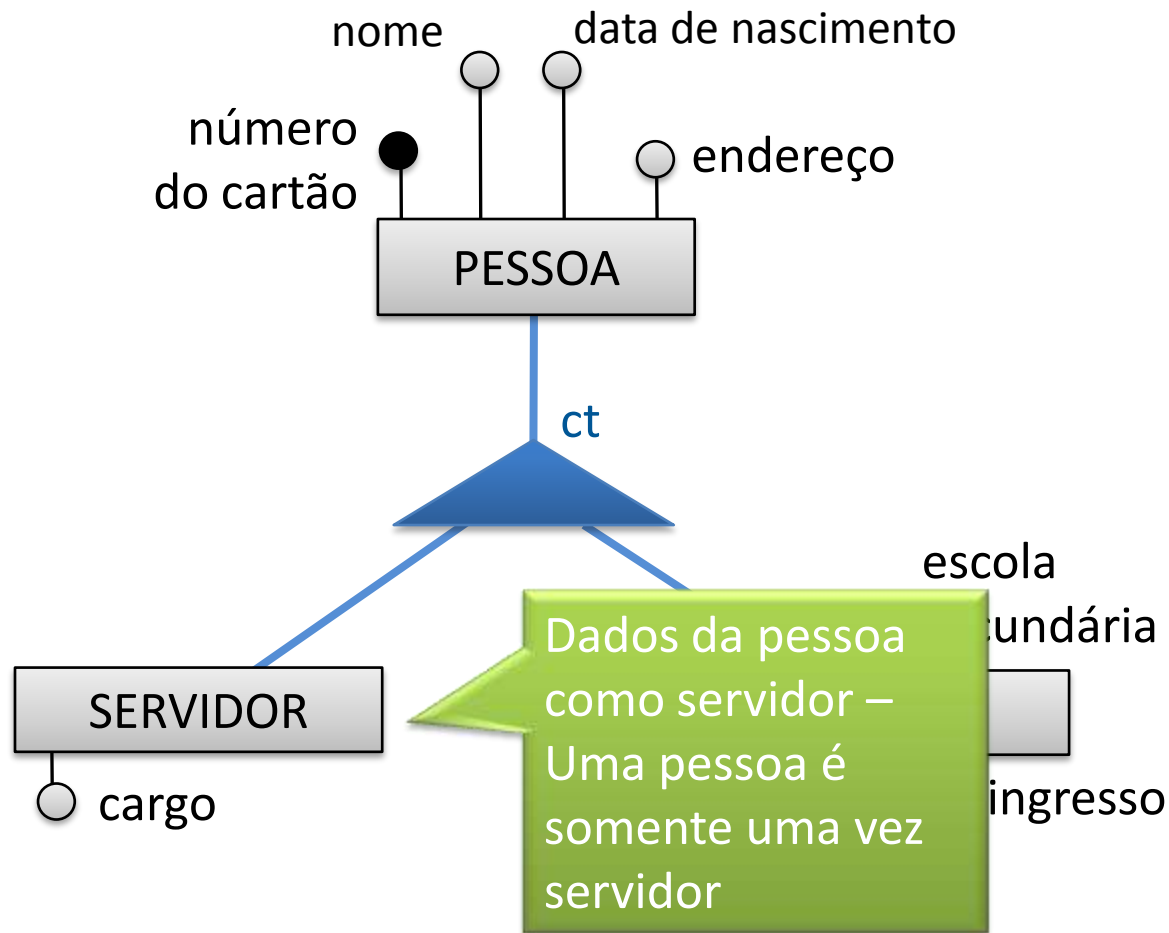
- Questão:
  - Deve-se modelar um determinado objeto como:  
uma entidade relacionada a outra?  
ou como uma especialização?
- Observar o identificador do objeto em questão:
  - Lembrar que uma entidade especializada herda o identificador de sua entidade genérica.



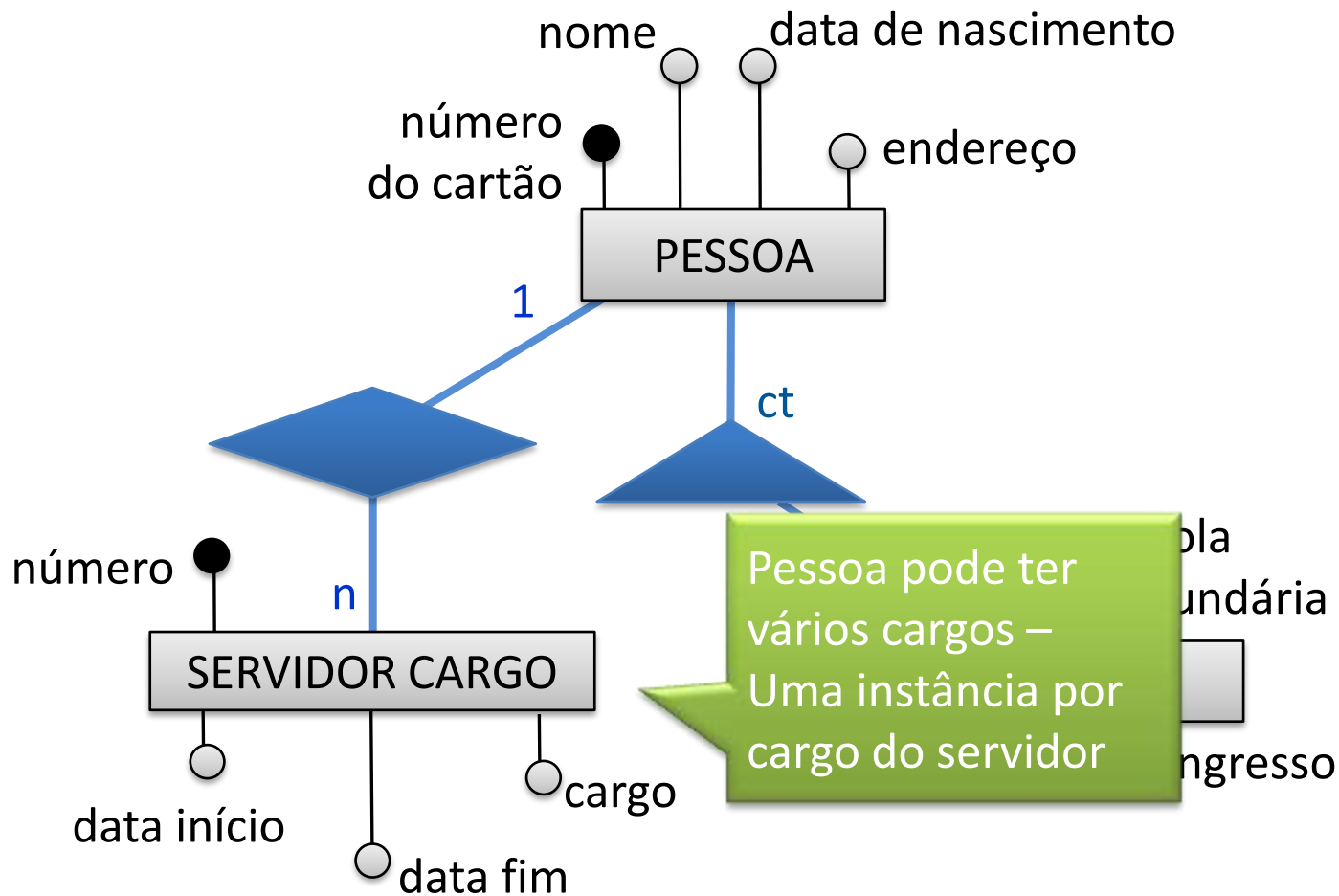
# Entidade versus especialização



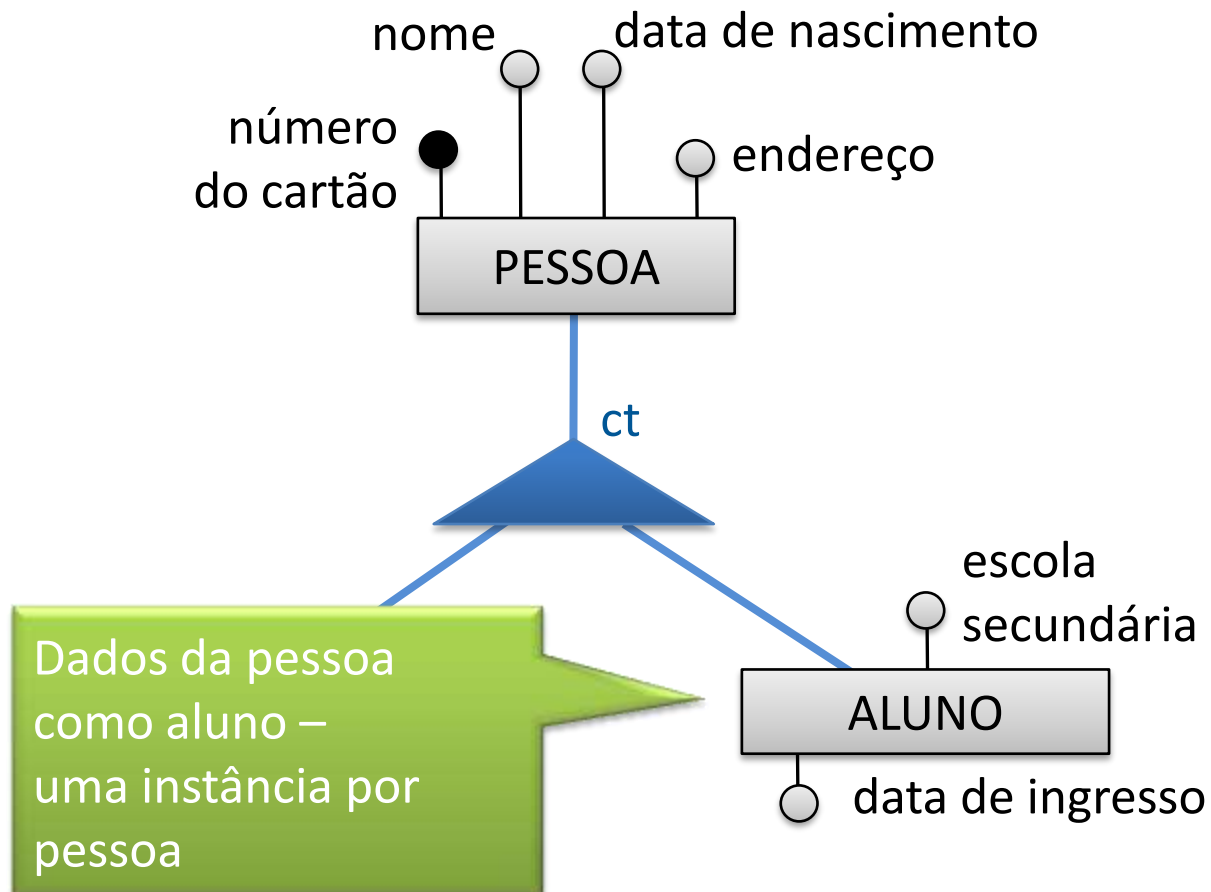
# Entidade versus especialização



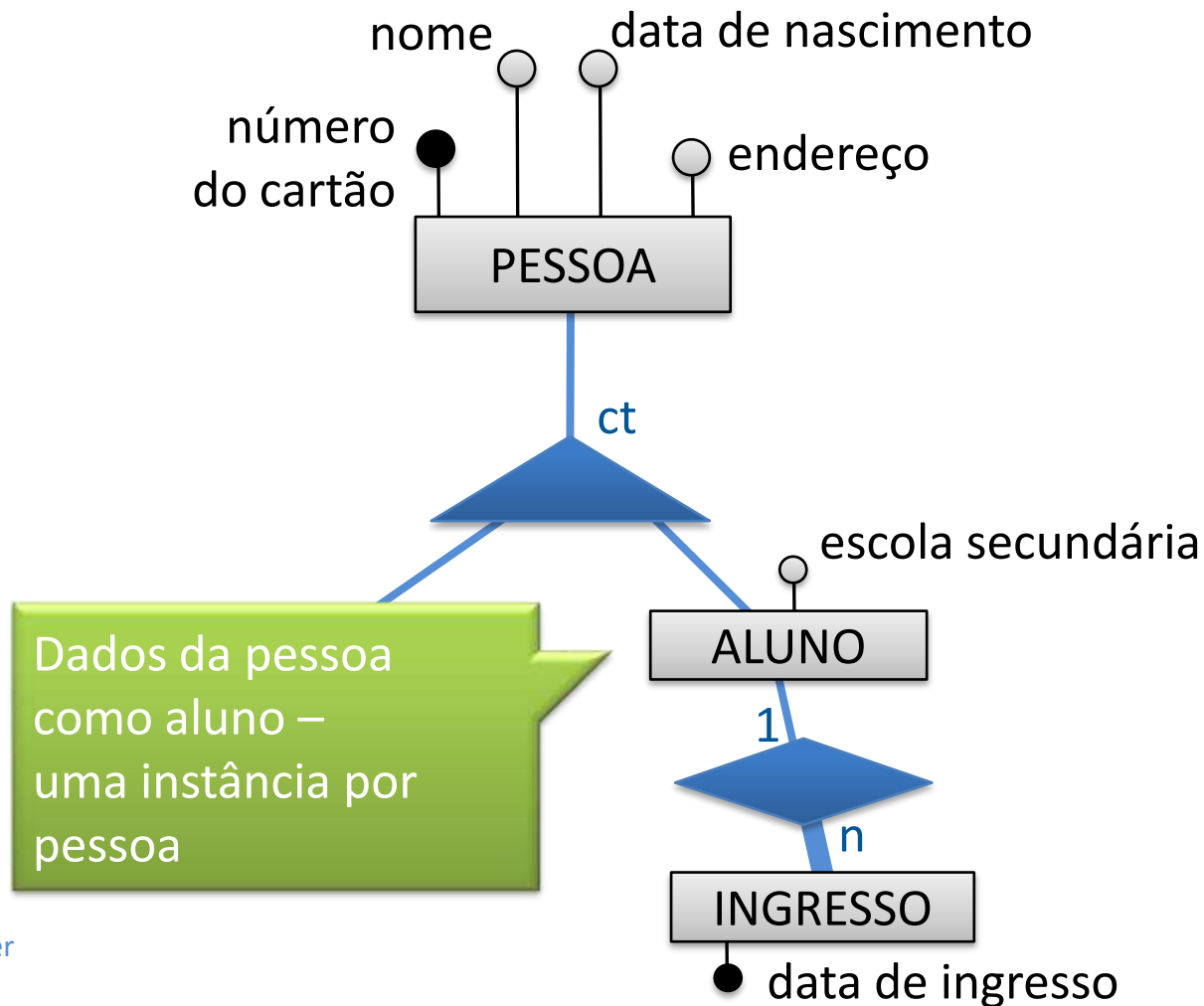
# Entidade versus especialização



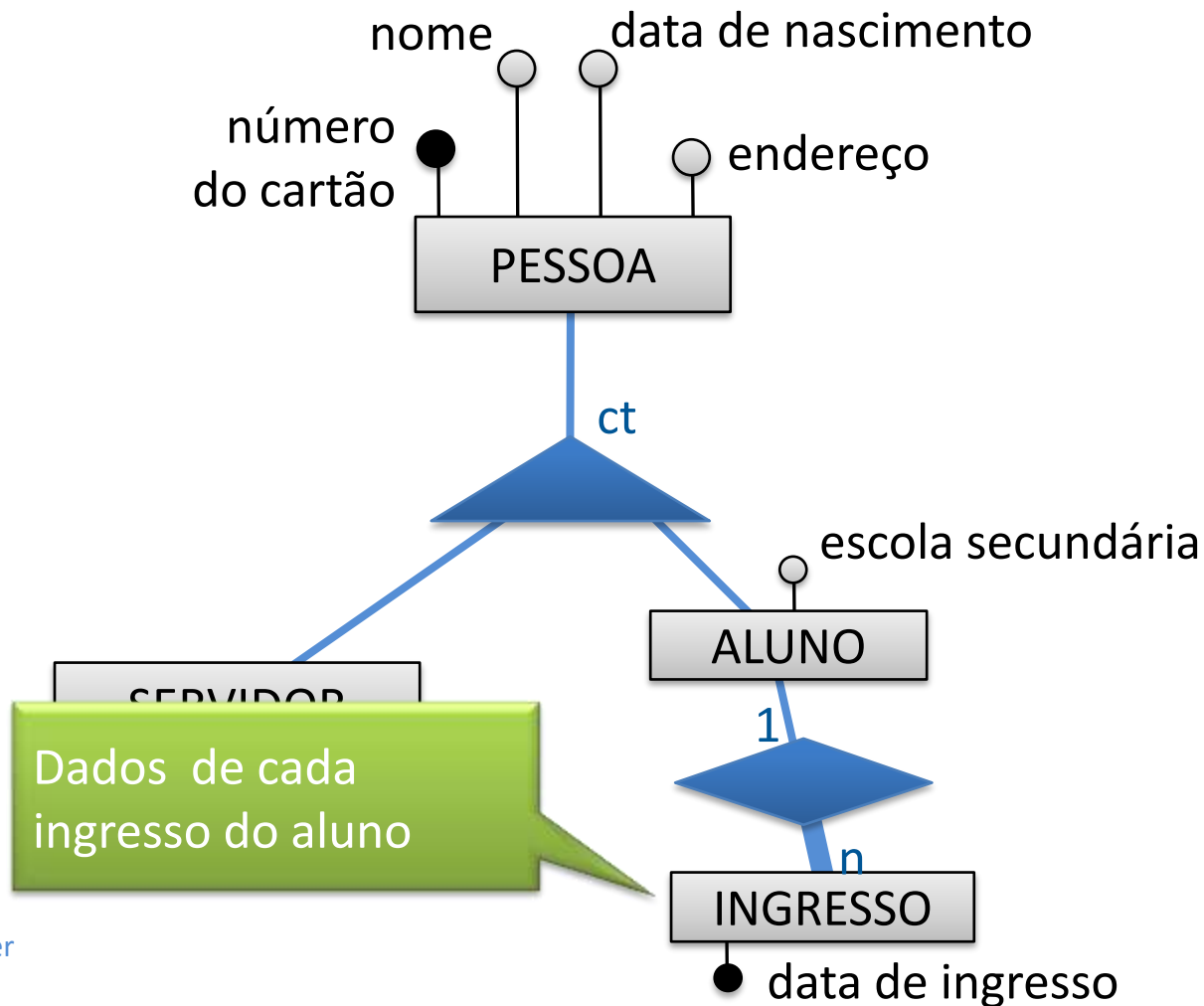
# Entidade versus especialização



# Entidade versus especialização

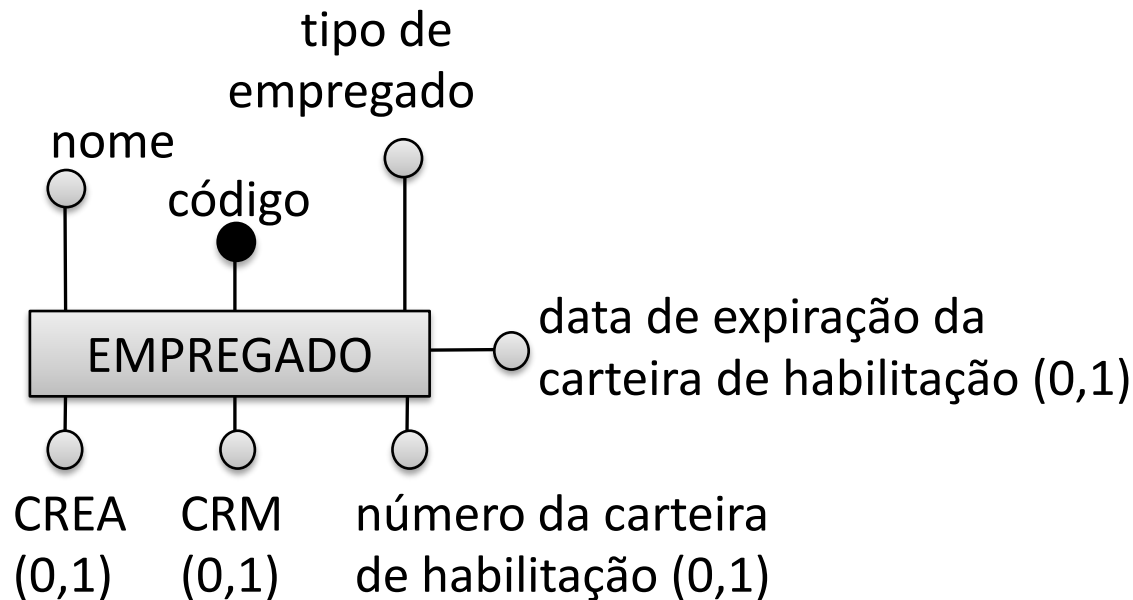


# Entidade versus especialização

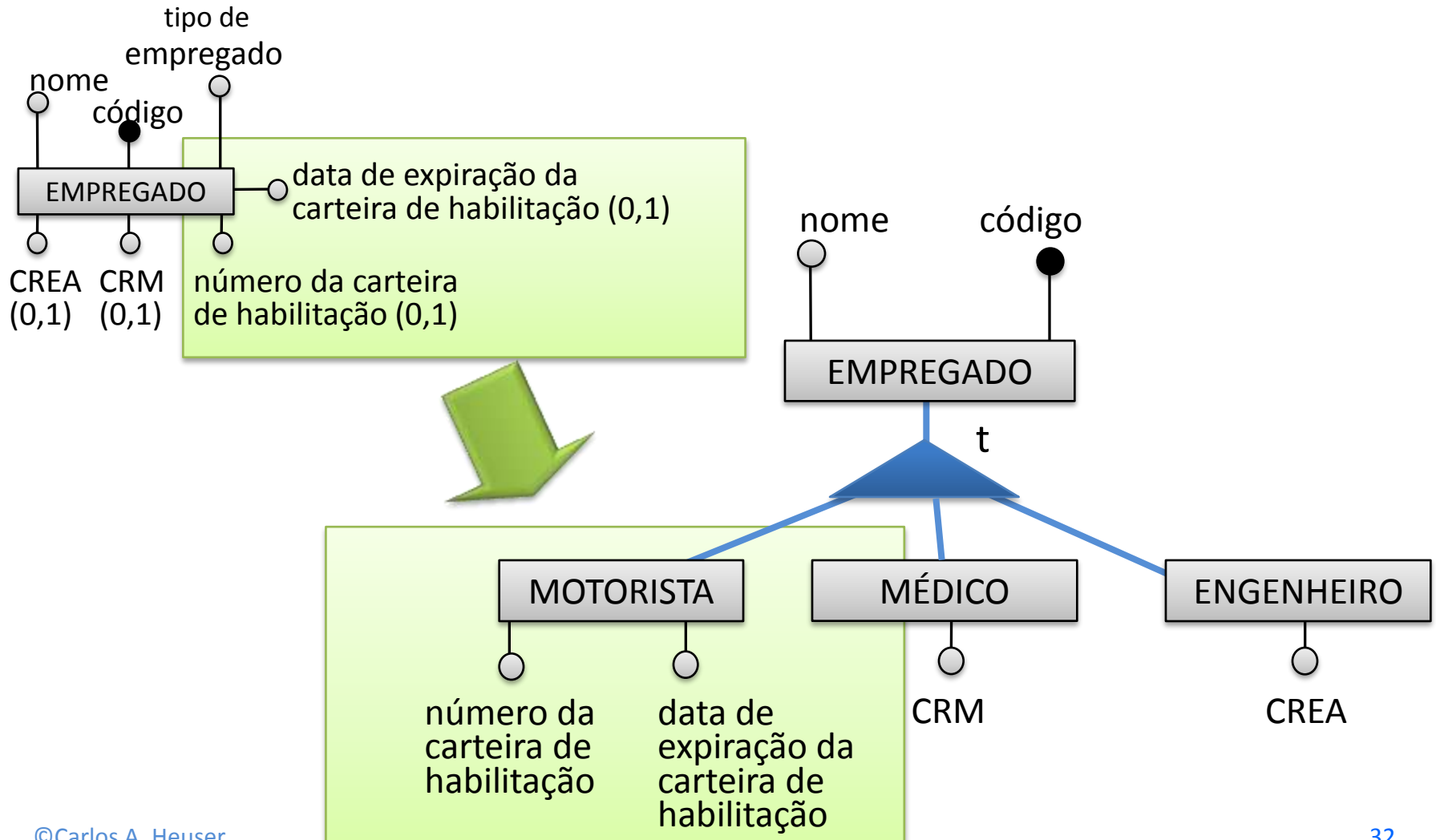


# Atributo opcional

- Atributo **opcional**:
  - Pode indicar subconjunto de entidade, que pode ser modelado mais corretamente através de **especialização**.

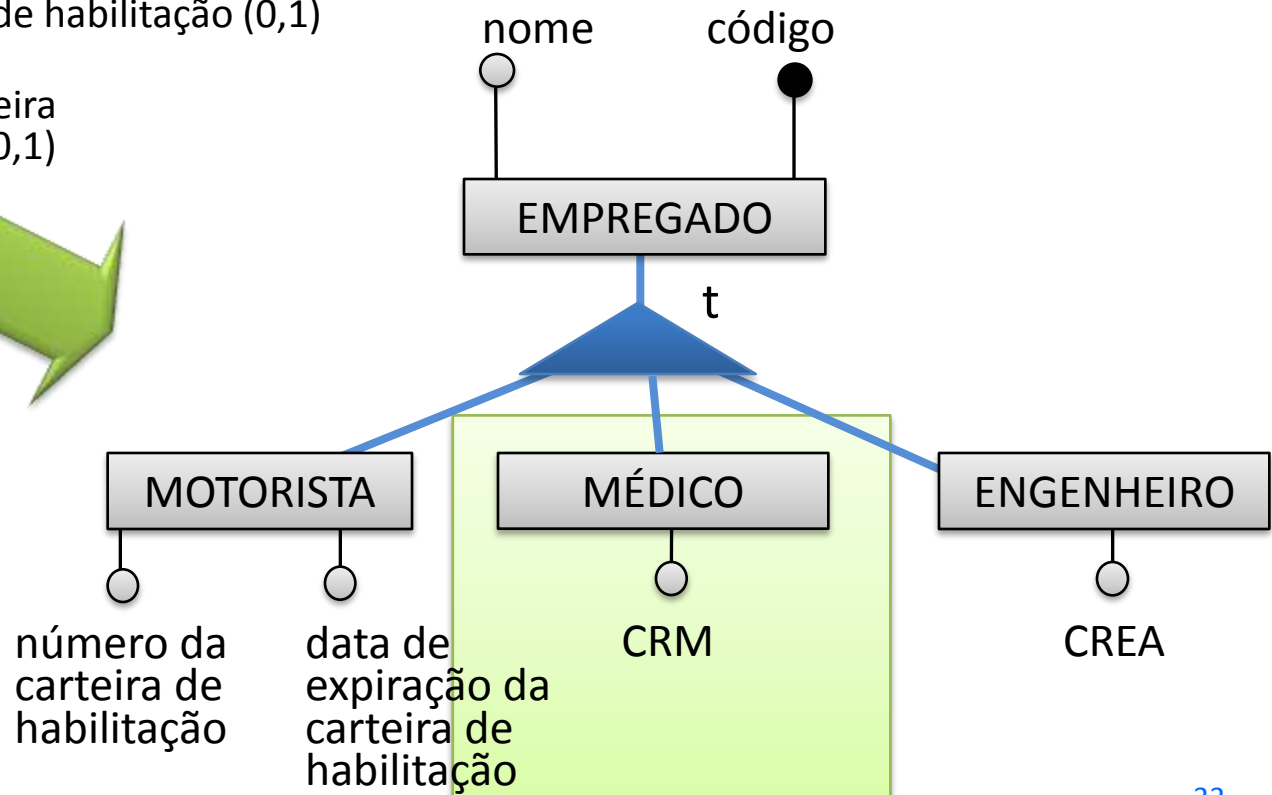
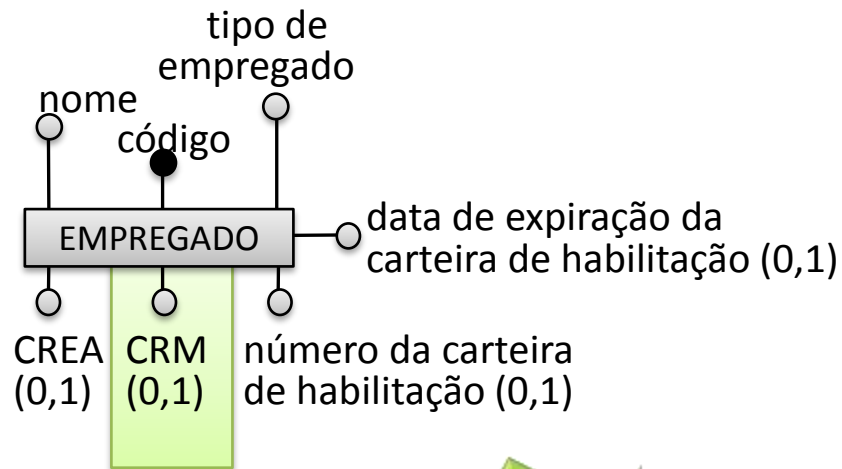


# Atributo opcional

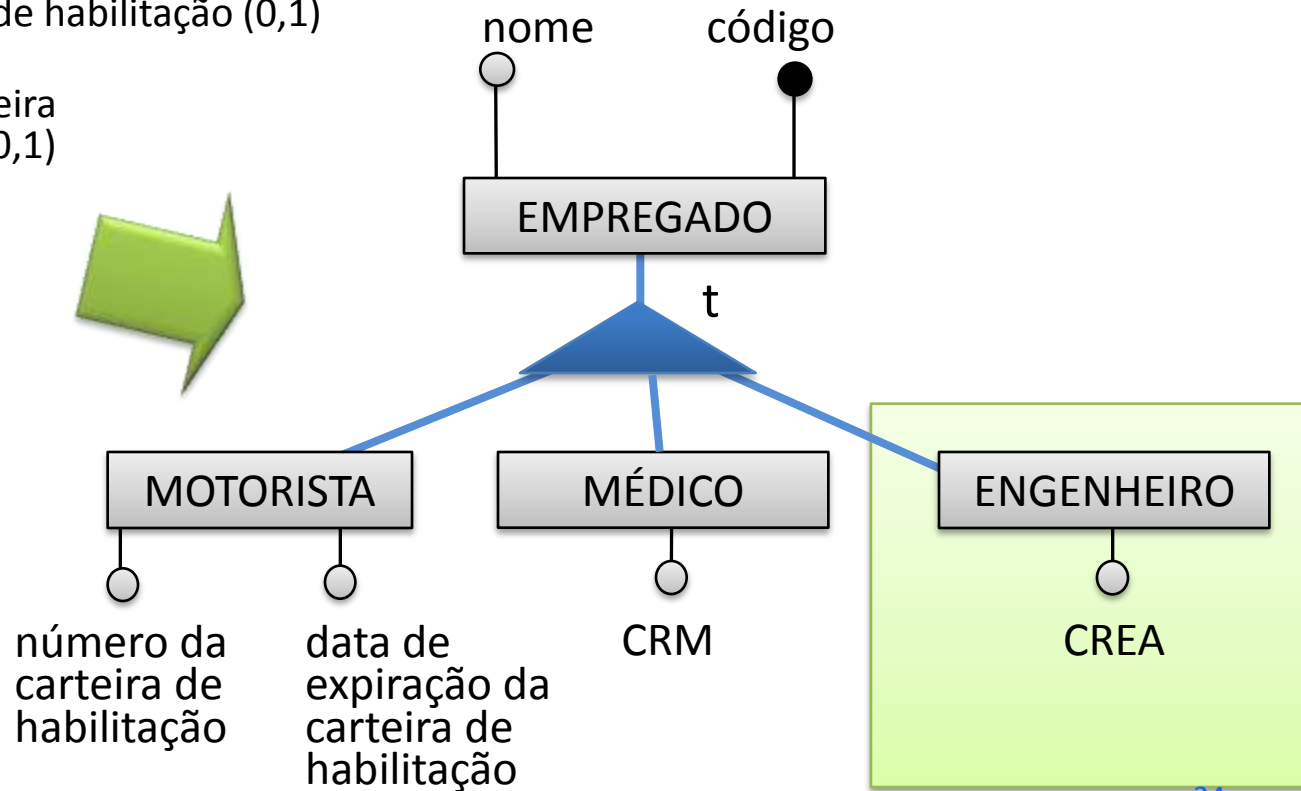
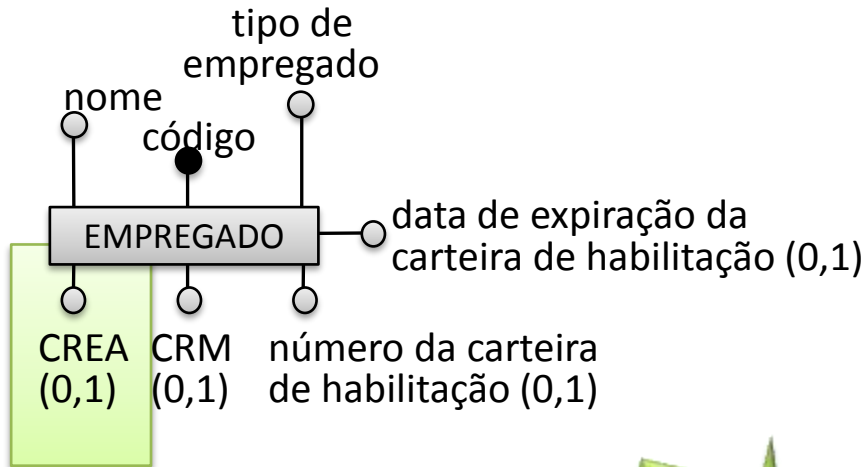




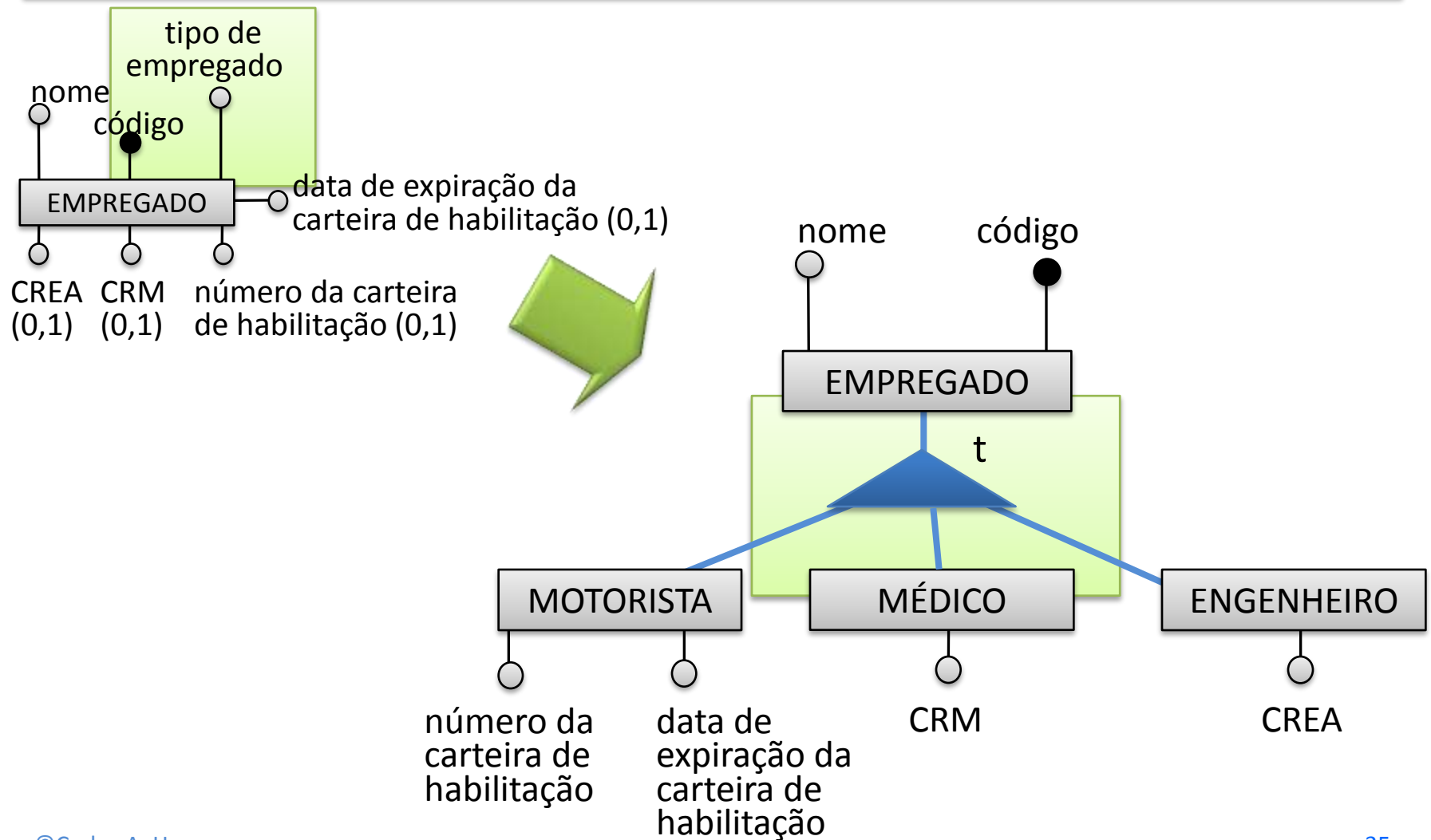
# Atributo opcional



# Atributo opcional



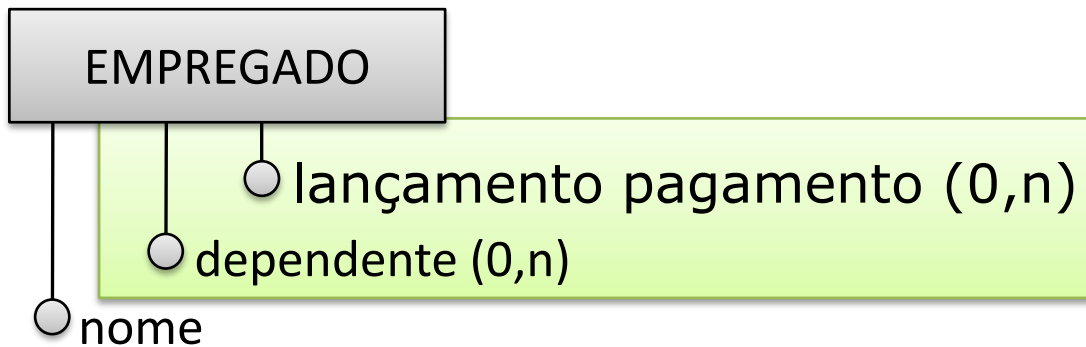
# Atributo opcional



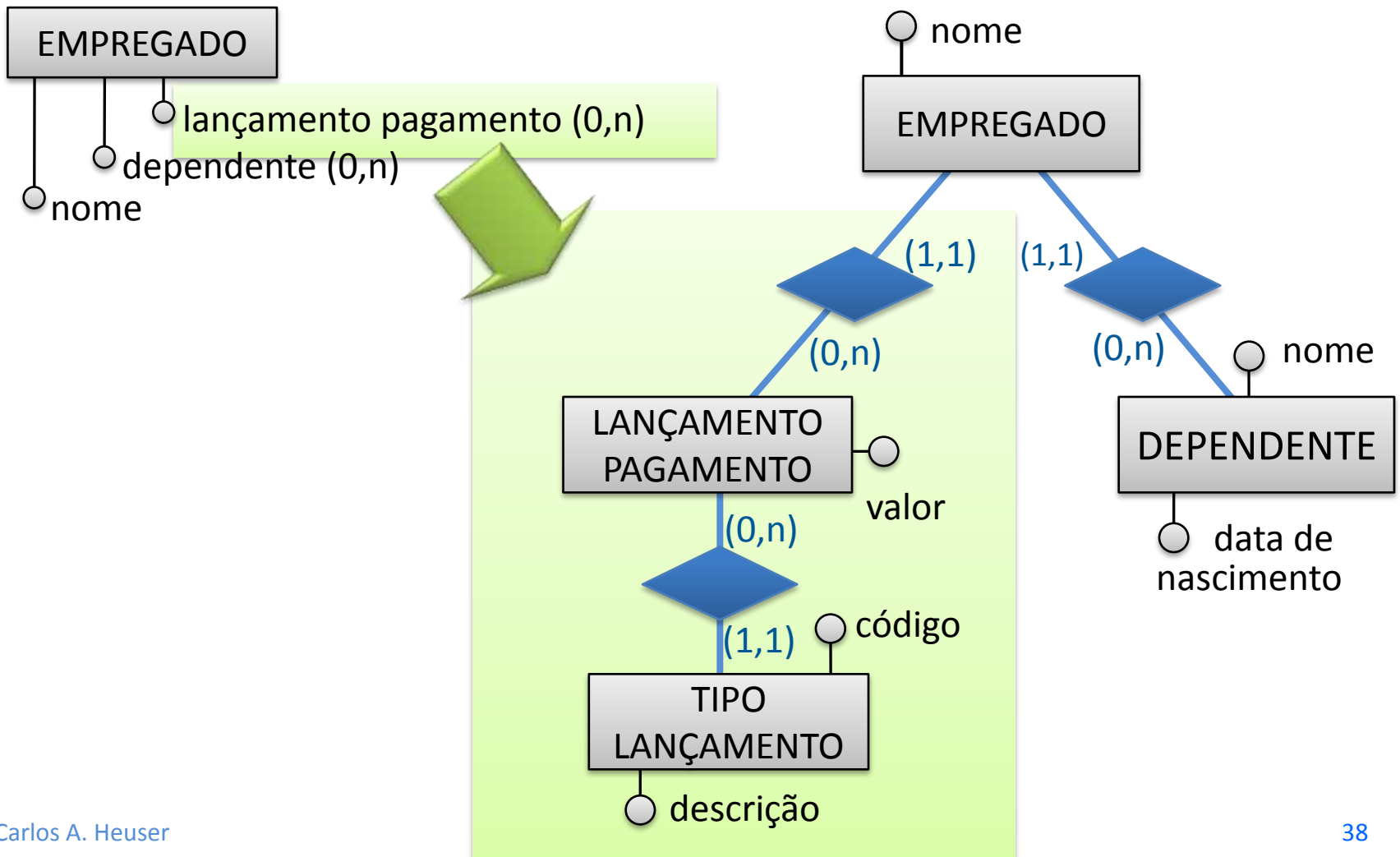
# Atributo multivalorado é indesejável

- SGBD relacional que segue o padrão SQL/2:
  - Atributo multivalorado não possui implementação direta.
- SGBD OO ou objeto/relacional:
  - Atributo multi-valorado normalmente é modelado como classe separada.
- Atributos multivalorados podem induzir a um erro de modelagem
  - Ocultar entidades e relacionamentos em atributos multivalorados

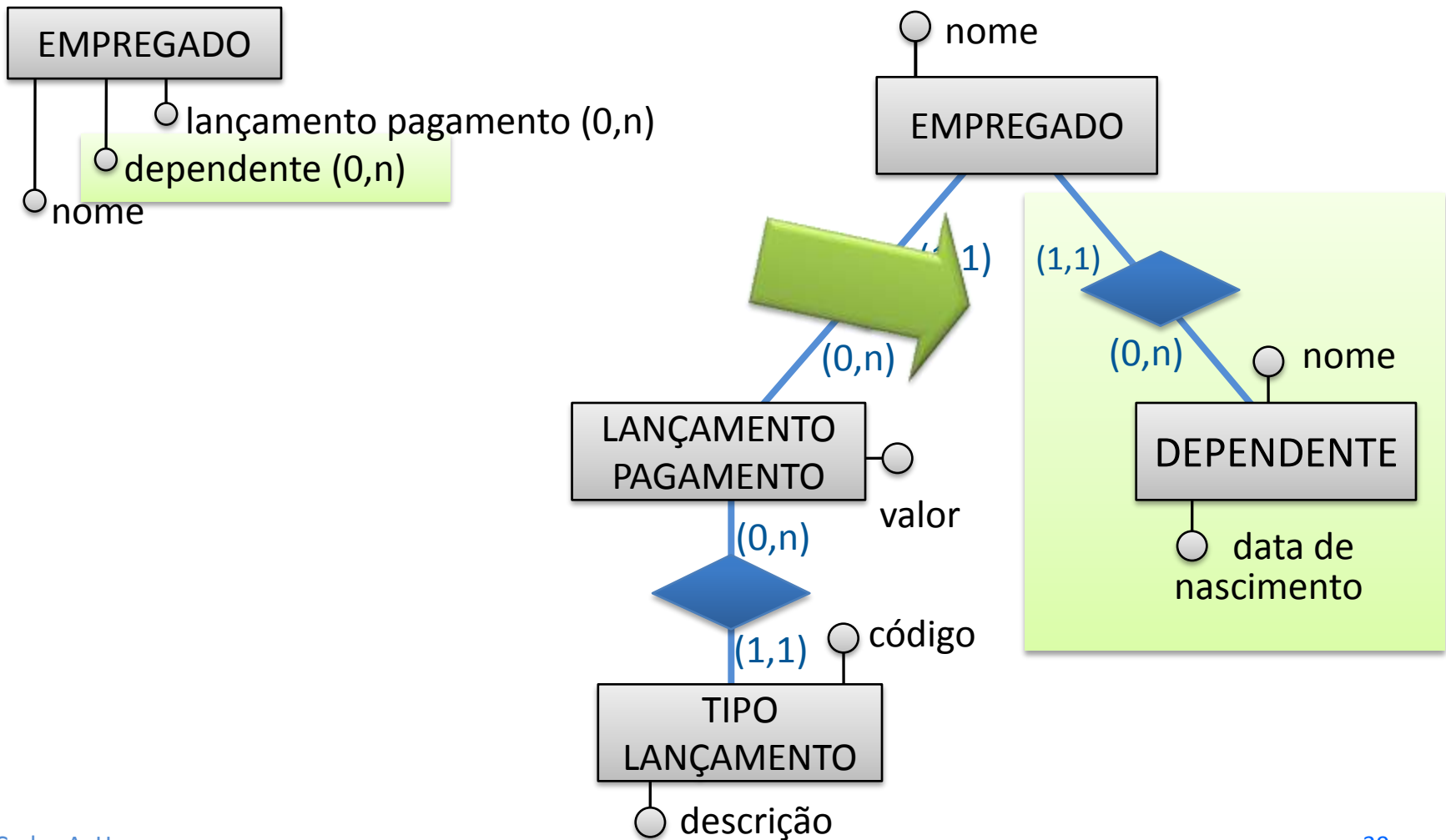
# Atributo multivalorado



# Atributo multivalorado eliminação



# Atributo multivalorado eliminação



# Verificação do modelo

- Modelo deve ser **correto**
- Modelo deve ser **completo**
- Modelo deve ser **livre de redundâncias**



# Modelo deve ser correto

- Erros:
  - sintáticos
  - semânticos
- Erros semânticos mais difíceis de verificar.
- Regras de normalização auxiliam na validação.

# Exemplos de erros semânticos

- Estabelecer associações incorretas:
  - associar a uma entidade um atributo que na realidade pertence a outra entidade.
- Usar uma entidade como atributo de outra entidade.
- Usar o número incorreto de entidades em um relacionamento.
  - fundir em um único relacionamento ternário dois relacionamentos binários independentes

# Modelo deve ser completo

- Deve fixar todas propriedades desejáveis do banco de dados.
- Somente pode ser verificado por alguém que conhece profundamente o sistema a ser implementado:
  - Envolvimento de **especialista no domínio da aplicação** (“usuário”).

# Verificação de completitude

- Forma de verificar:
  - dados que devem ser obtidos do banco de dados estão presentes?
  - todas as transações de modificação do banco de dados podem ser executadas sobre o modelo?
- Requisito é aparentemente conflitante com a falta de poder de expressão de modelos ER.

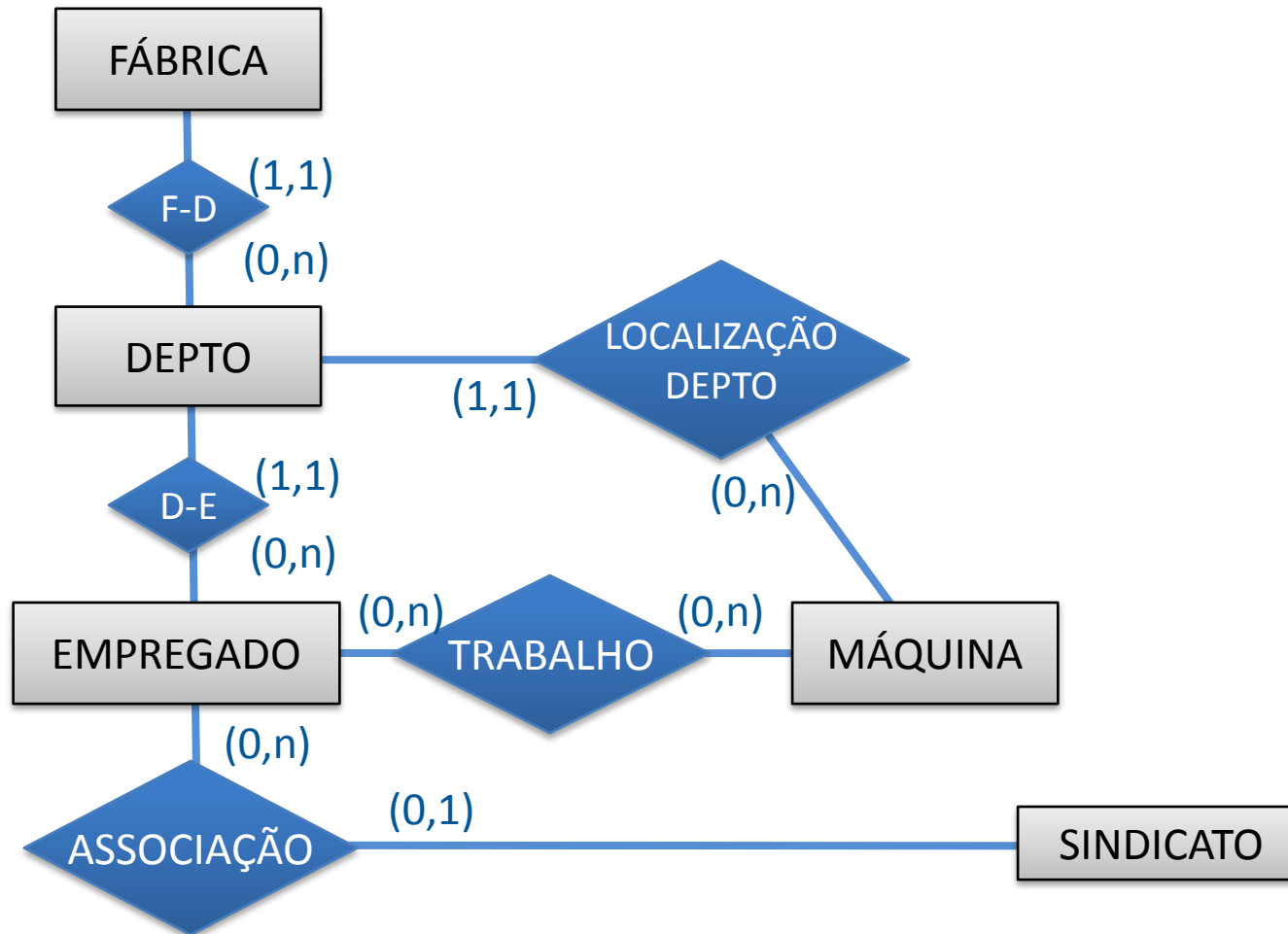
# Modelo deve ser livre de redundâncias

- Modelo deve ser mínimo, isto é não deve conter conceitos redundantes.
- Tipos de redundância:
  - Relacionamentos redundantes
  - Atributos redundantes

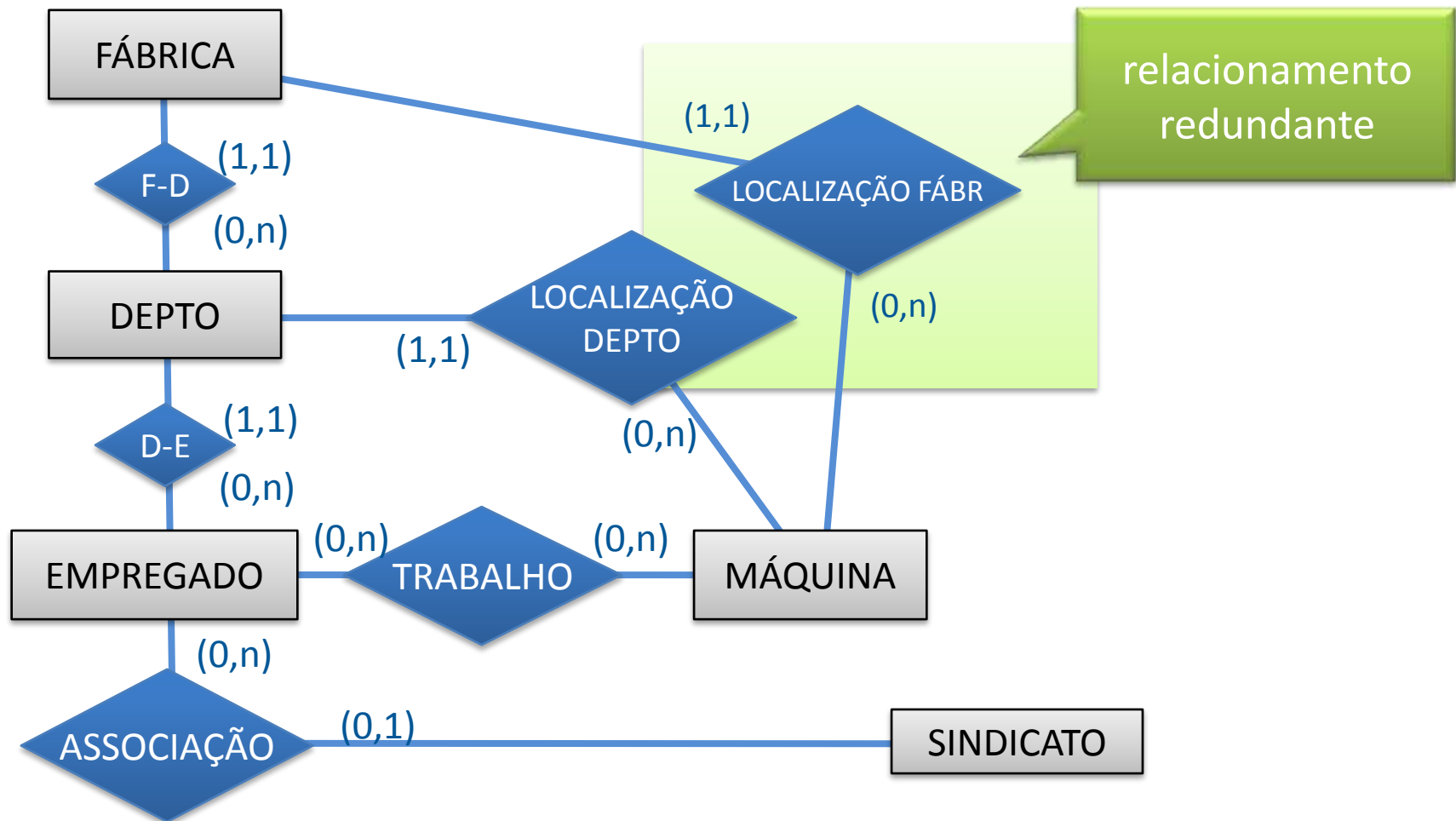
# O que fazer com construções redundantes?

- Alternativas
  - não devem aparecer no modelo ou
  - devem aparecer indicadas como redundantes
- Implementação pode conter redundância controlada de dados (desempenho, tolerância a falhas)

# Exemplo de modelo

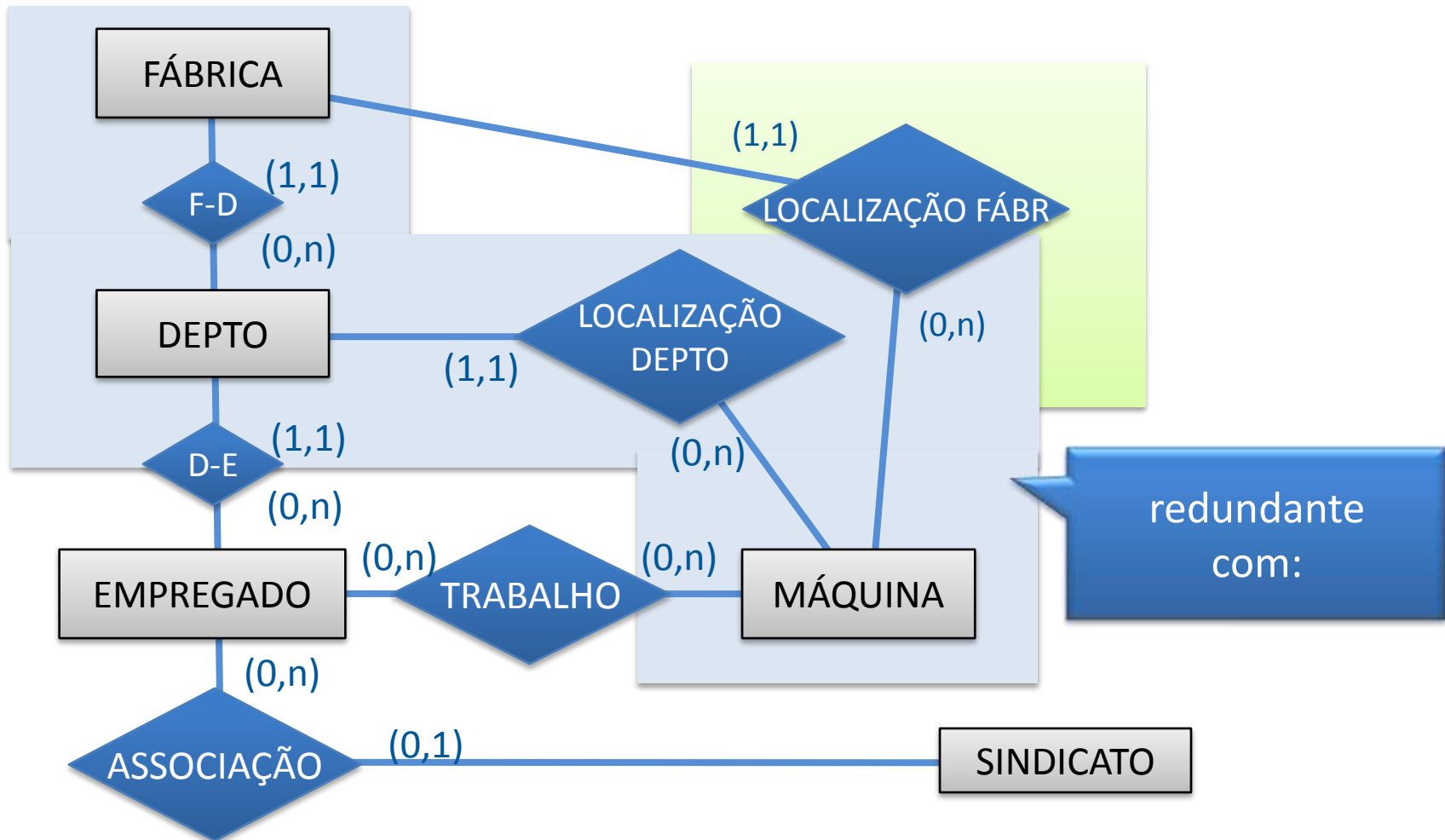


# Relacionamentos redundantes

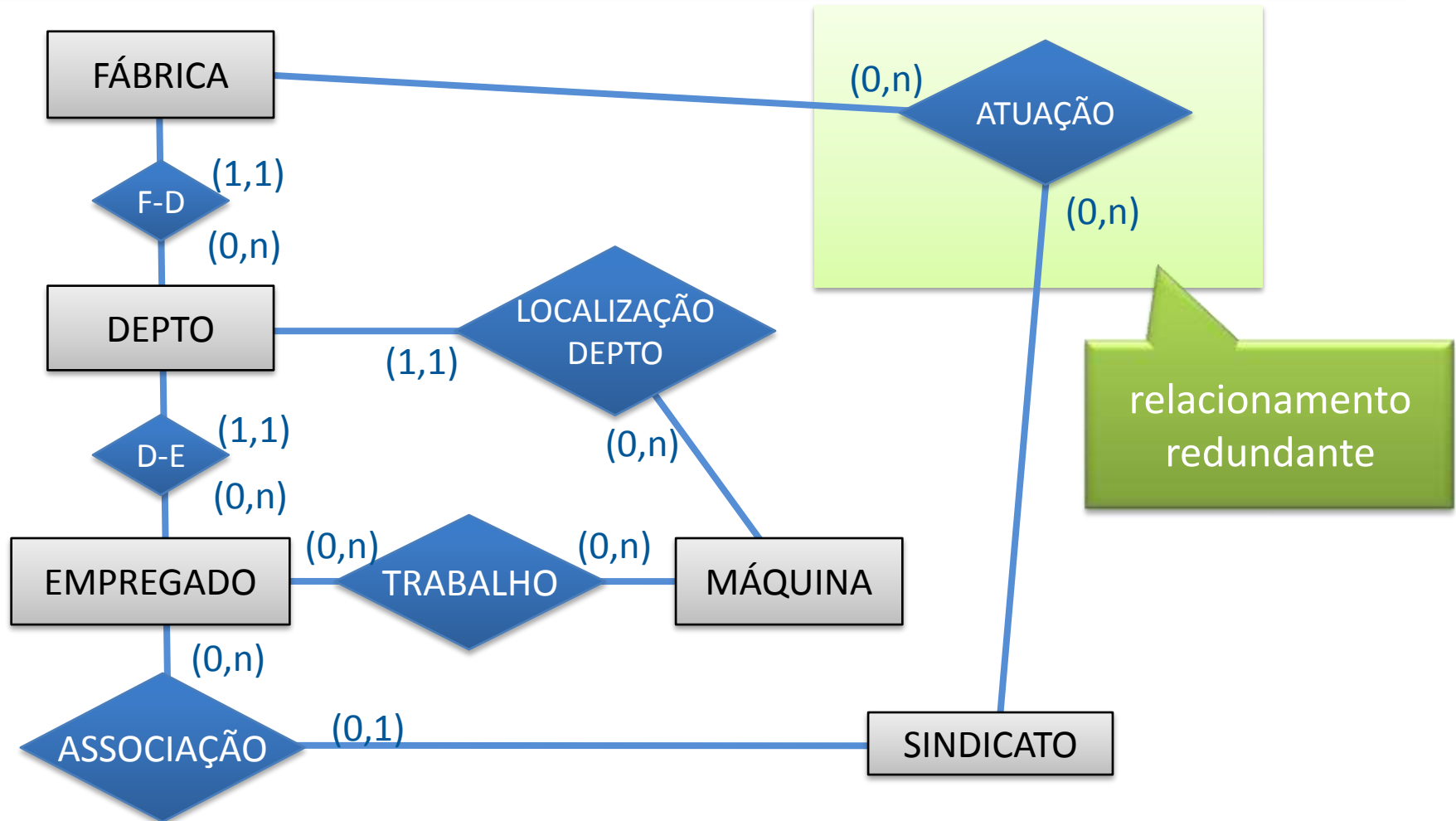




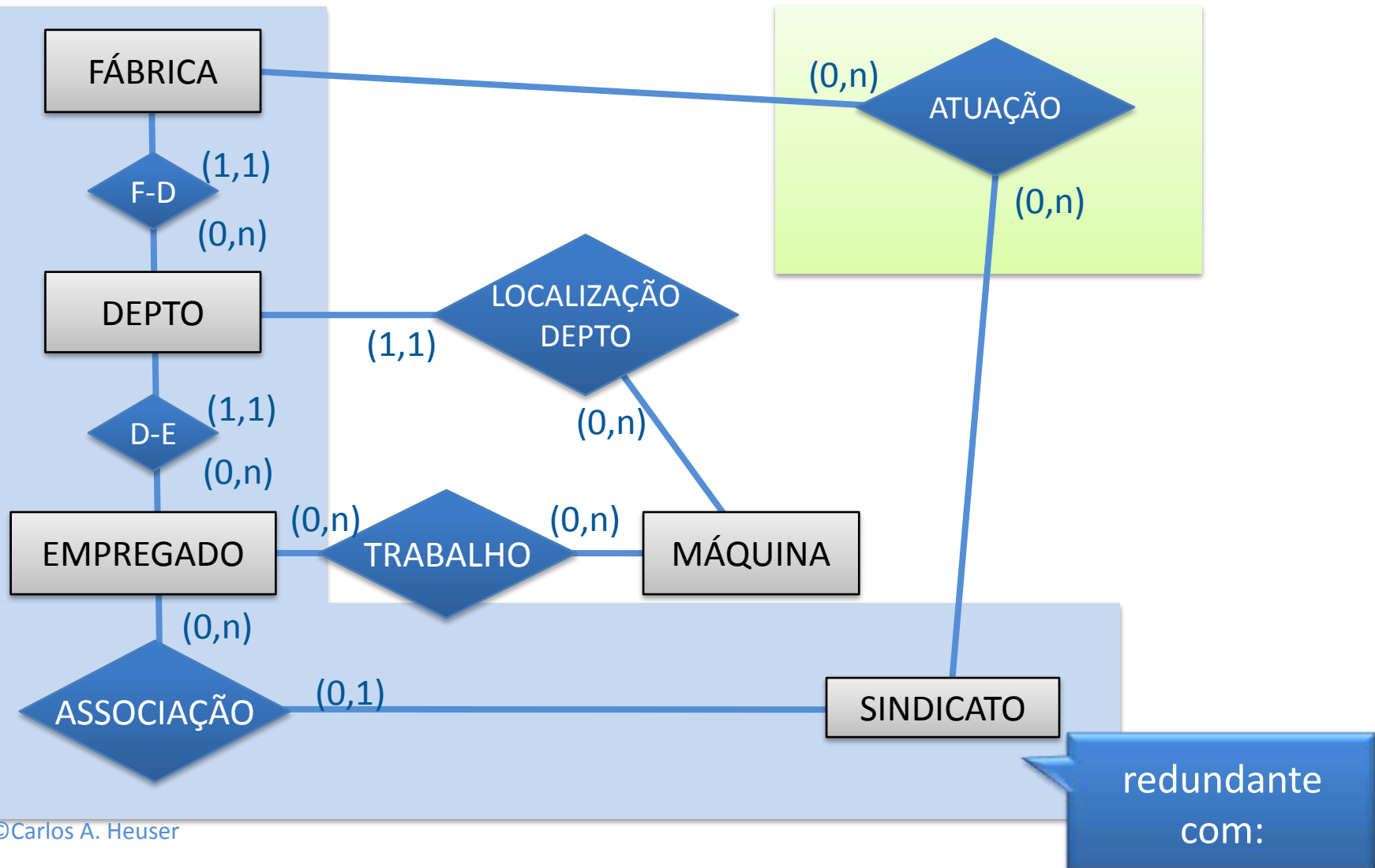
# Relacionamentos redundantes



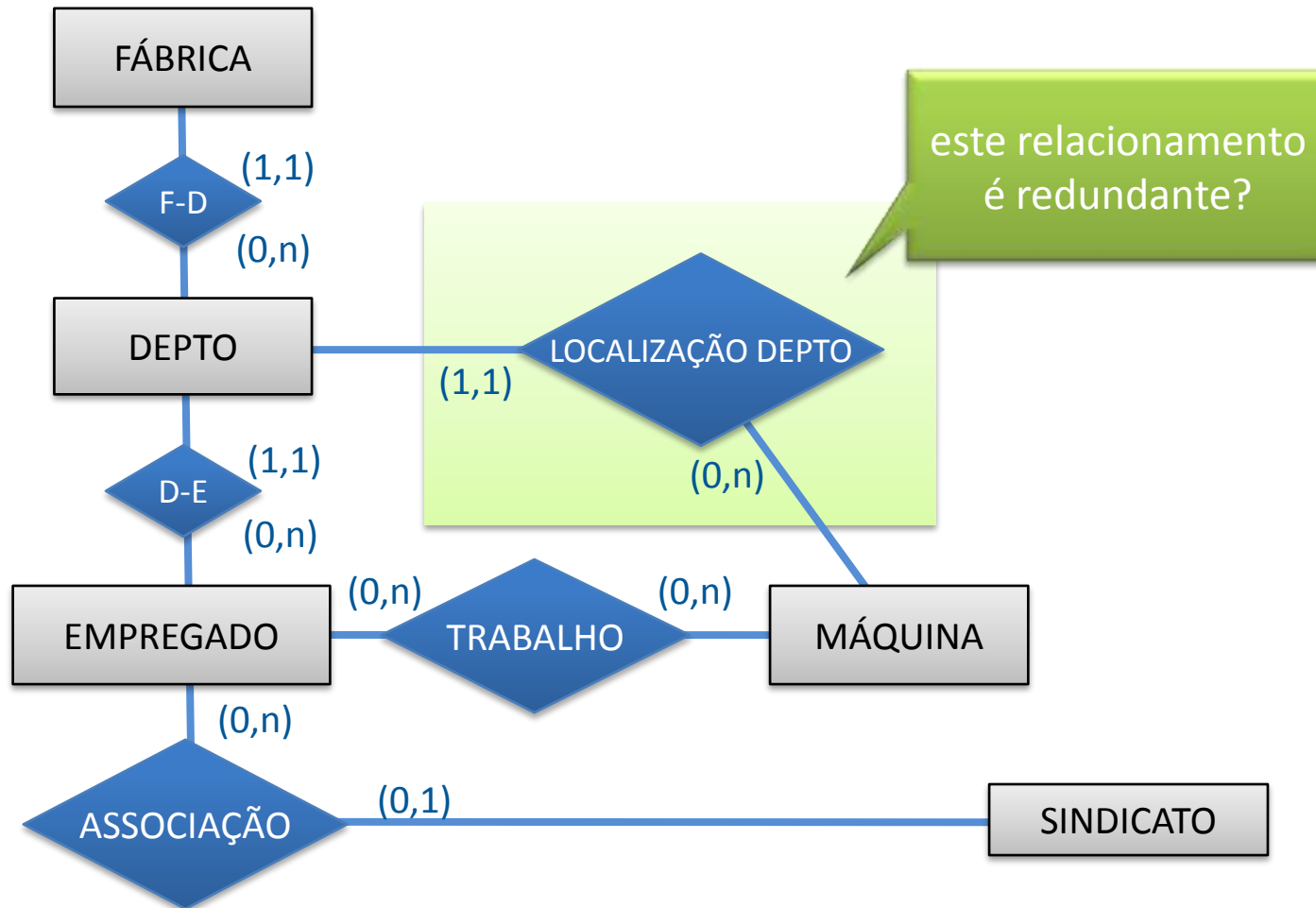
# Relacionamentos redundantes



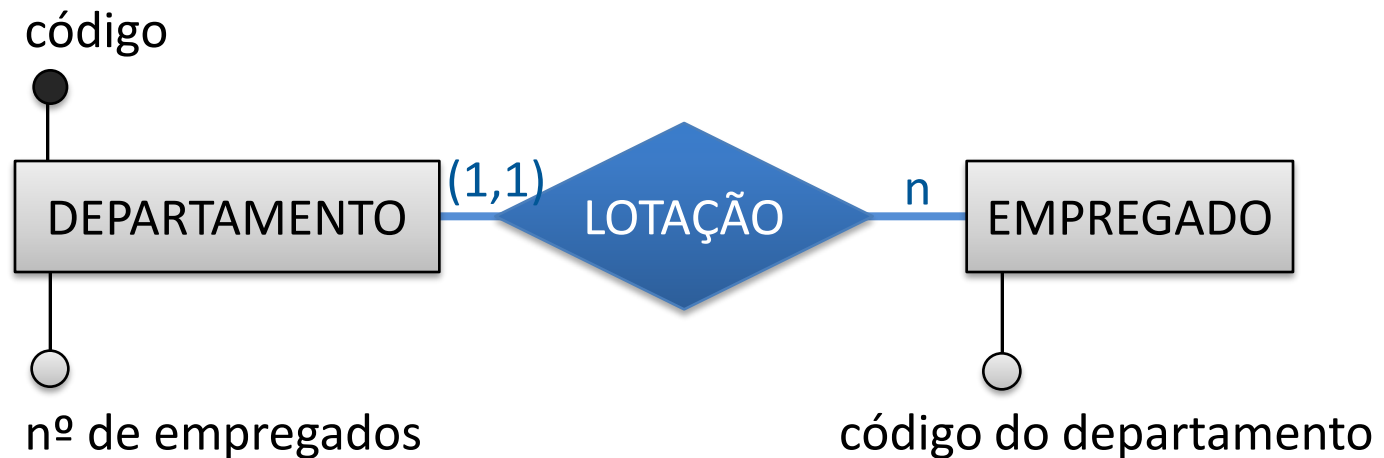
# Relacionamentos redundantes



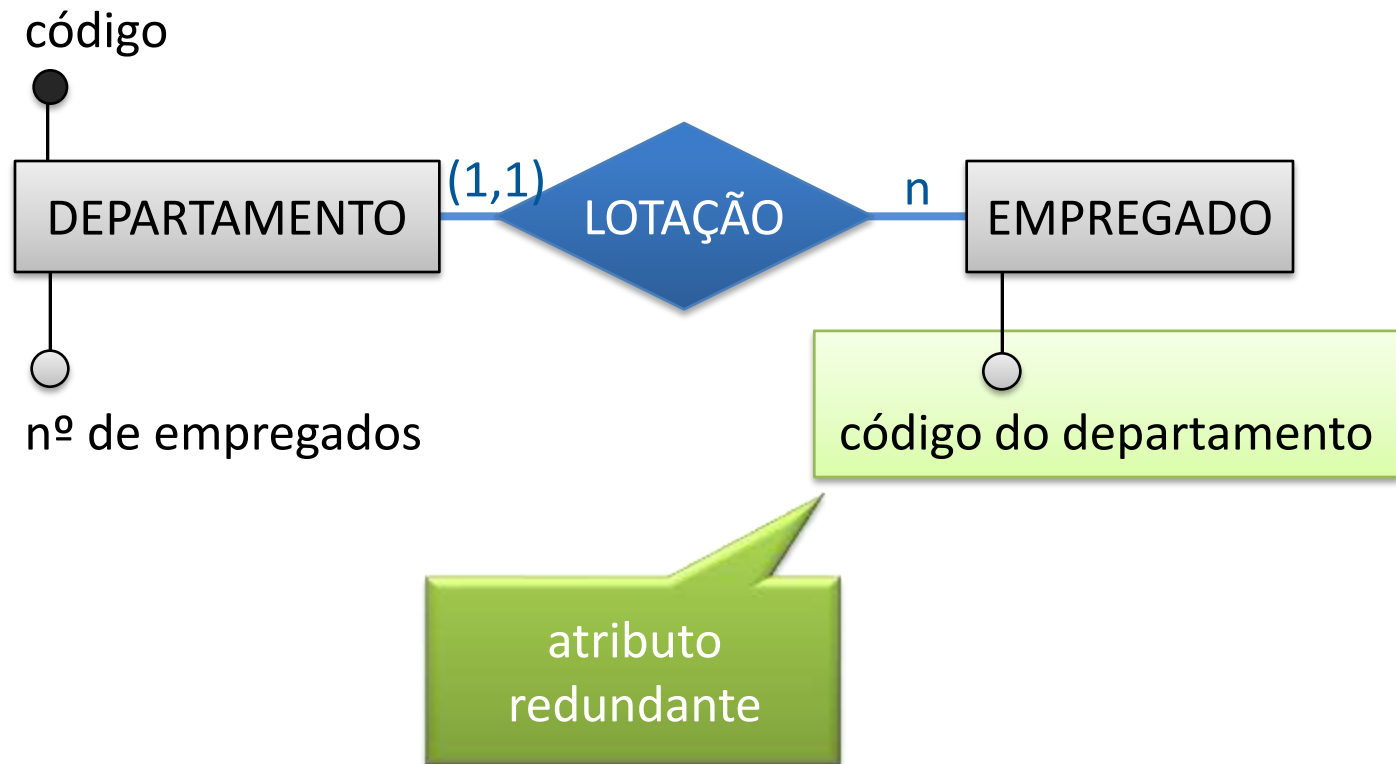
# Relacionamentos redundantes



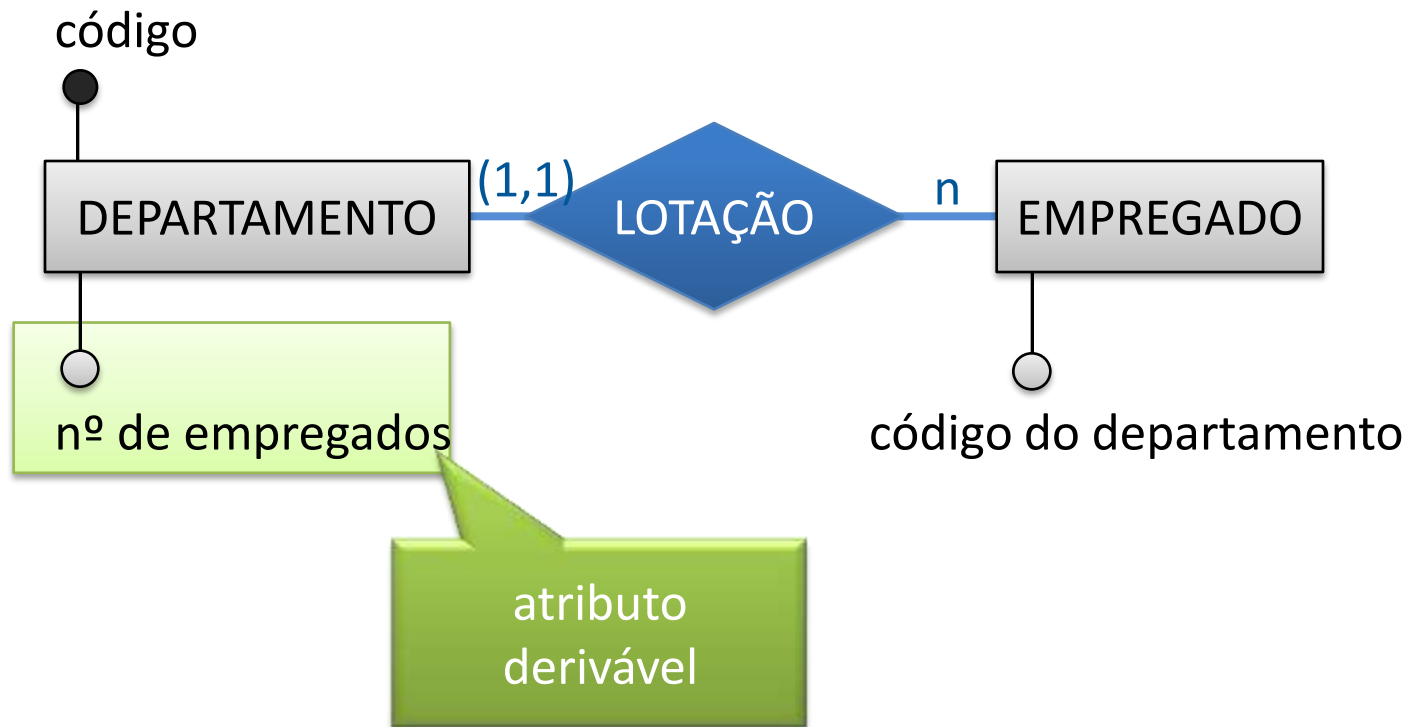
# Atributos redundantes ou deriváveis



# Atributos redundantes ou deriváveis



# Atributos redundantes ou deriváveis



# Modelo deve refletir o aspecto temporal

- Dados temporais:
  - dados que mudam ao longo do tempo e
  - para as quais BD mantém histórico.
- Tipos de dados **temporais**:
  - **Atributos** cujos valores modificam ao longo do tempo;
  - **Relacionamentos** que modificam ao longo do tempo.



# Atributos temporais



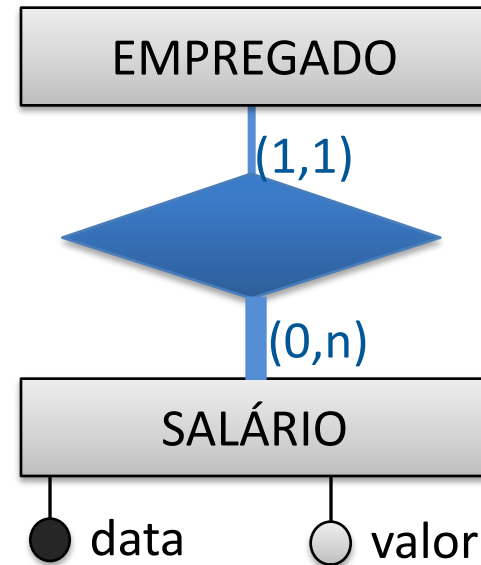
cada empregado tem  
um único salário  
(salário atual)

# Atributos temporais



(a)

Banco de dados contém apenas o salário atual



(b)

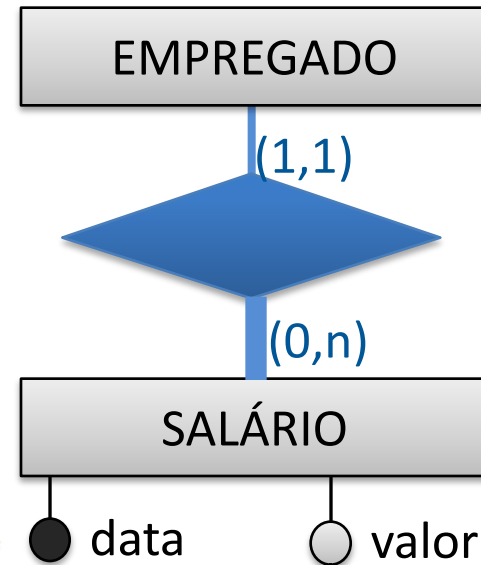
Banco de dados contém a história dos salários

# Atributos temporais



(a)

Banco de dados contém apenas o salário atual

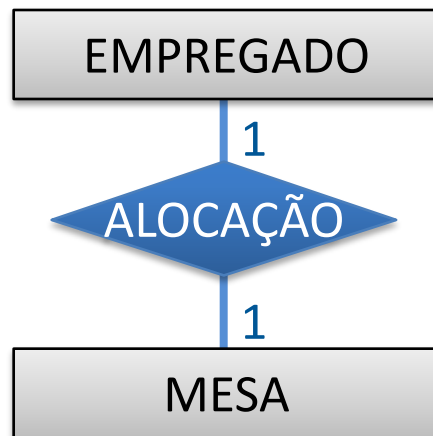


(b)

Banco de dados contém a história dos salários

ponto no tempo ou  
faixa (intervalo) de  
tempo

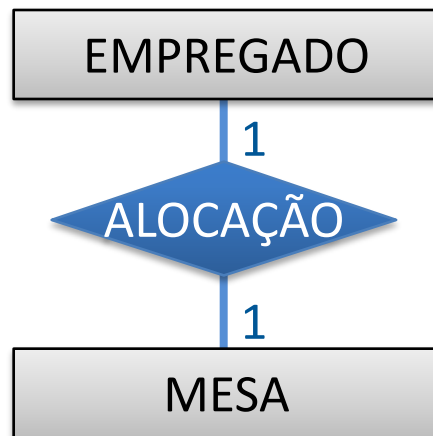
# Relacionamento 1:1 temporal



(a)

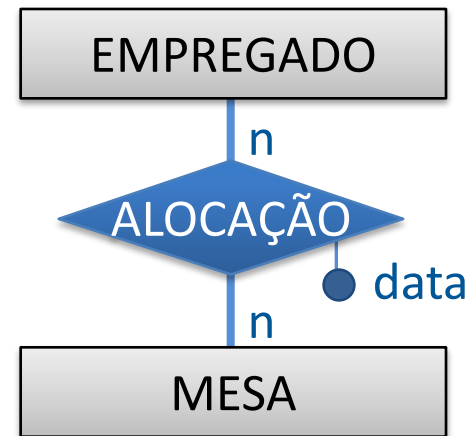
Base de dados  
contém apenas a  
alocação atual

# Relacionamento 1:1 temporal



(a)

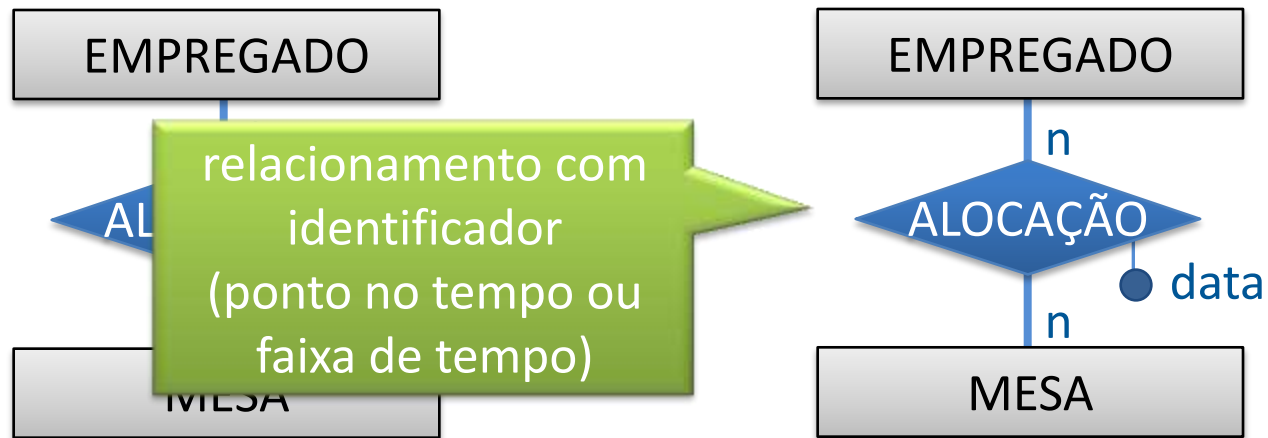
Base de dados  
contém apenas a  
alocação atual



(b)

Base de dados  
contém a **história das**  
**alocações**

# Relacionamento 1:1 temporal



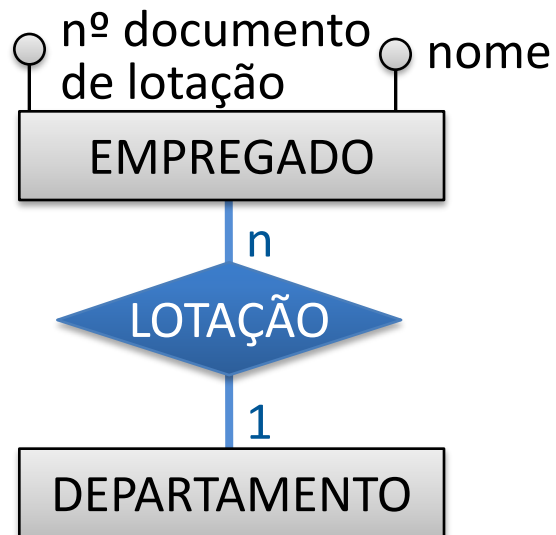
(a)

Base de dados  
contém apenas a  
alocação atual

(b)

Base de dados  
contém a **história das**  
**alocações**

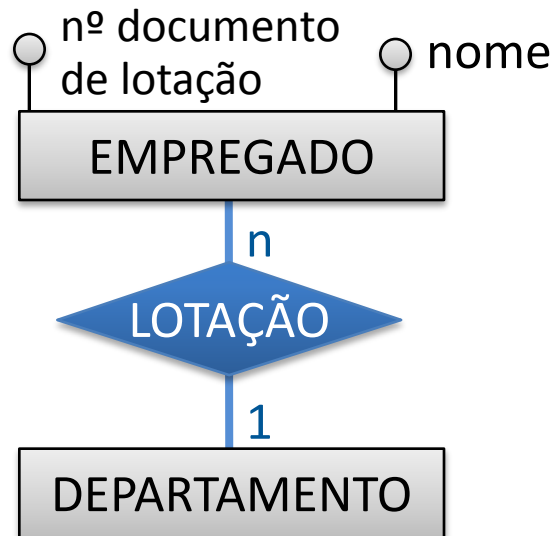
# Relacionamento 1:n temporal



(a)

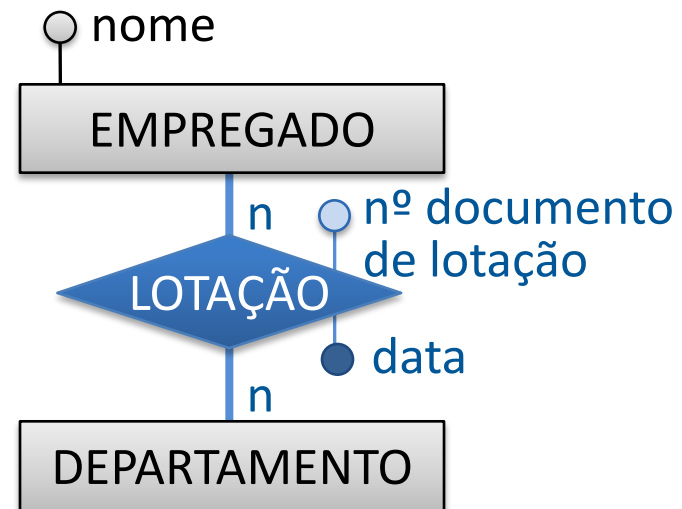
Base de dados contém  
apenas a lotação atual

# Relacionamento 1:n temporal



(a)

Base de dados contém apenas a lotação atual

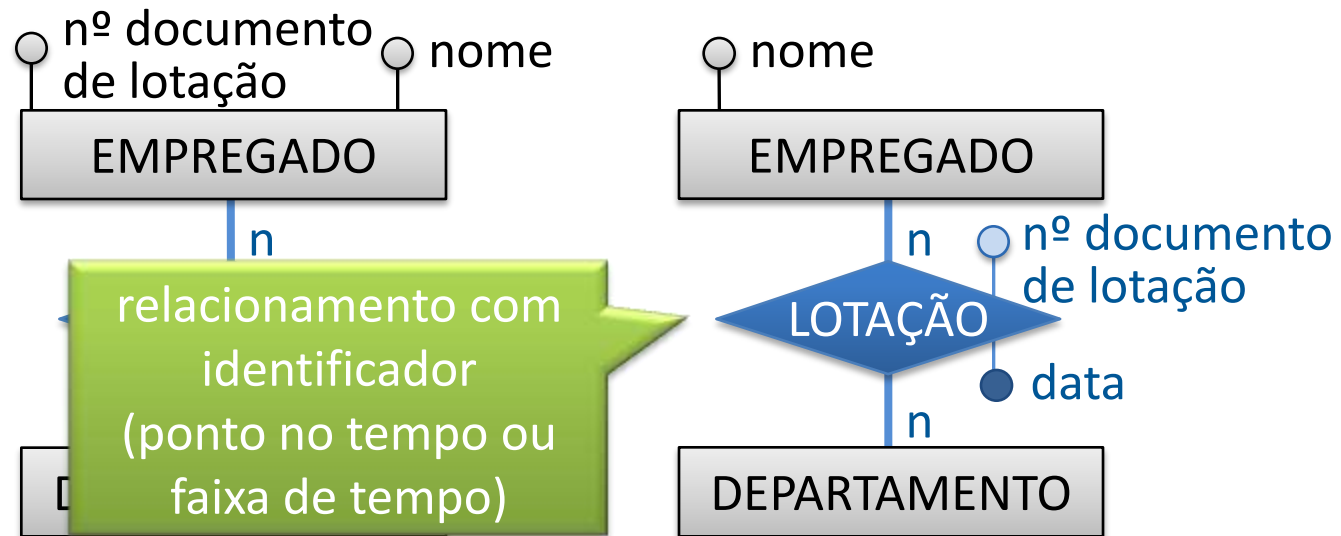


(b)

Base de dados contém a história das lotações



# Relacionamento 1:n temporal



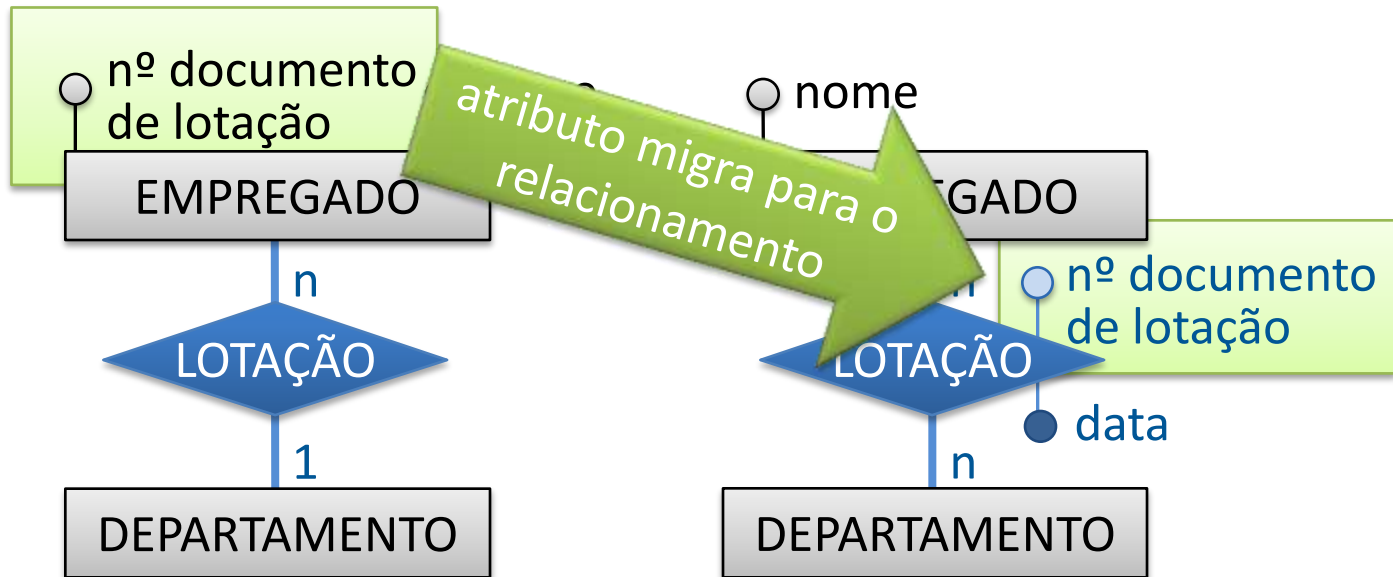
(a)

Base de dados contém apenas a lotação atual

(b)

Base de dados contém a história das lotações

# Relacionamento 1:n temporal



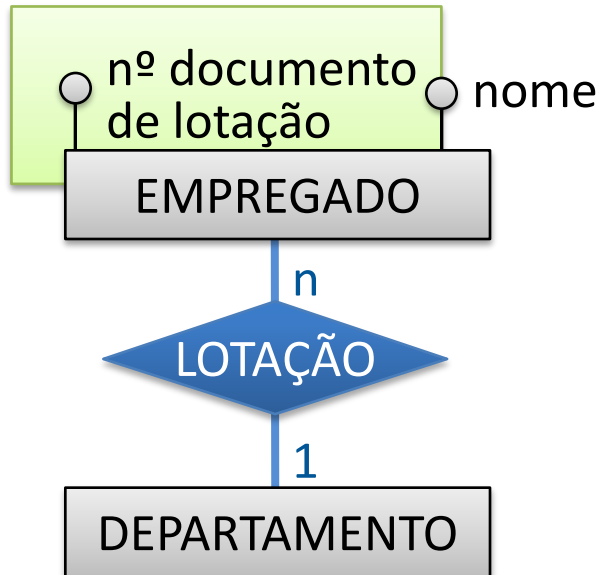
(a)

Base de dados contém apenas a lotação atual

(b)

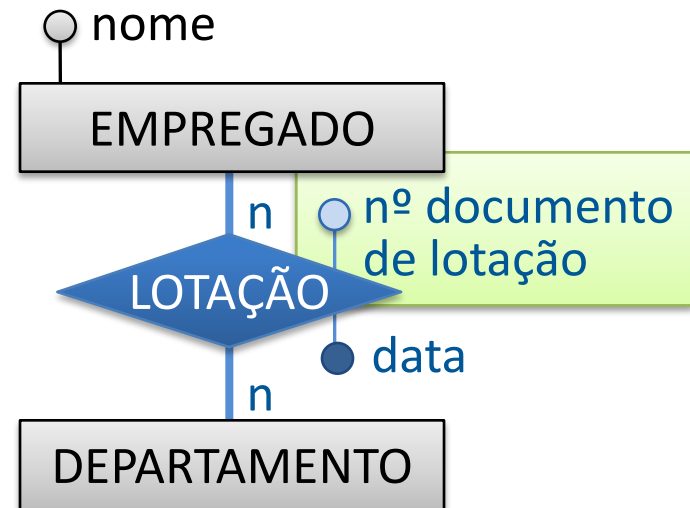
Base de dados contém a história das lotações

# Relacionamento 1:n temporal



(a)

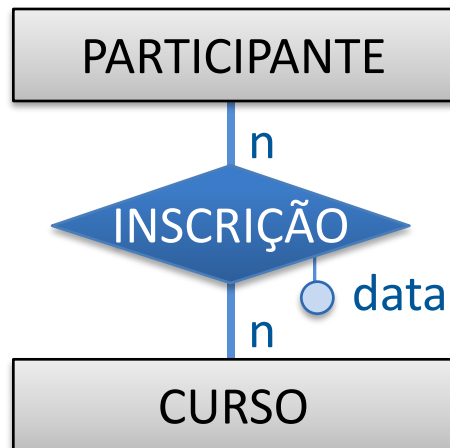
Base de dados contém apenas a lotação atual



(b)

Base de dados contém a história das lotações

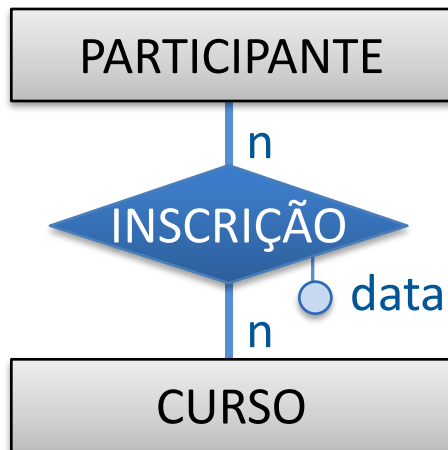
# Relacionamento n:n temporal



(a)

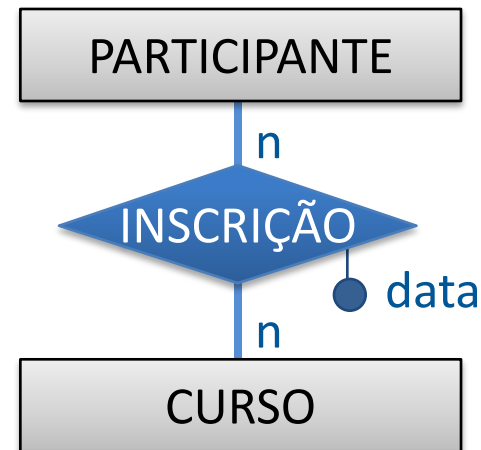
Base de dados contém  
apenas a inscrição atual

# Relacionamento n:n temporal



(a)

Base de dados contém apenas a inscrição atual



(b)

Base de dados contém a história das inscrições

# Consultas a dados referentes ao passado

- Muitas vezes, informações referentes ao passado são eliminadas da base de dados (**arquivamento**).
- Podem ser necessárias no futuro:
  - por **motivos legais**
  - para realização de **auditorias**
  - para **tomada de decisões**

## Dados referentes ao passado planejar arquivamento

- Solução que poderia ser considerada:
  - reincluir as informações no banco de dados, quando elas forem necessárias.
  - Problema: restrições de **integridade referencial**.
- Planejar informações **estatísticas**:
  - Quando informações antigas são necessárias apenas para **tomada de decisões**.
  - Pode ser conveniente manter no banco de dados informações compiladas e eliminar as informações usadas na compilação.

# Entidade isolada

## Entidade isolada

Entidade que não apresenta  
relacionamento com outras entidades.



# Entidade isolada

- Analisar:
  - Caso raro, mas não incorreto.
- Caso típico:
  - Entidade que modela a organização na qual o sistema implementado pelo BD está embutida.

# Entidade isolada exemplo

- Exemplo: BD de uma universidade.
- A entidade **UNIVERSIDADE** pode ser necessária, caso se deseje manter no BD alguns atributos da universidade.
- O modelo **não deveria conter** o relacionamento desta entidade com outras, como **ALUNO** ou **CURSO**:
  - BD modela **uma única universidade**;
  - Não é necessário informar no BD em que universidade o aluno está inscrito ou a qual universidade o curso pertence.

# Estabelecimento de padrões

- Modelos de dados são usados para comunicação:
  - com pessoas da organização,
  - com programas (ferramentas CASE, geradores de código,...).
- É necessário estabelecer padrões de confecção de modelos.
- Na prática e na literatura:
  - Muitas variantes de modelo ER.
  - Variantes em:
    - sintaxe,
    - semântica.

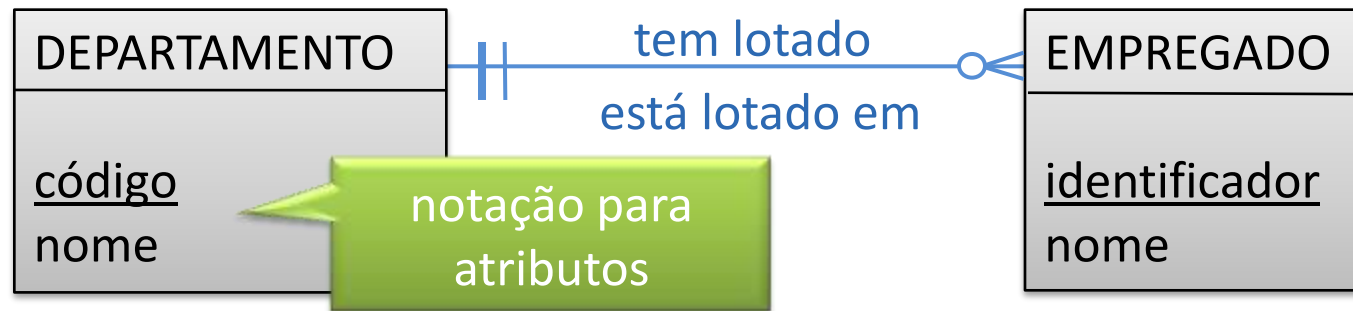
# Variantes de modelos ER

- Peter Chen (acadêmica)
- Engenharia de Informações
- UML
- Merise (notação Européia)

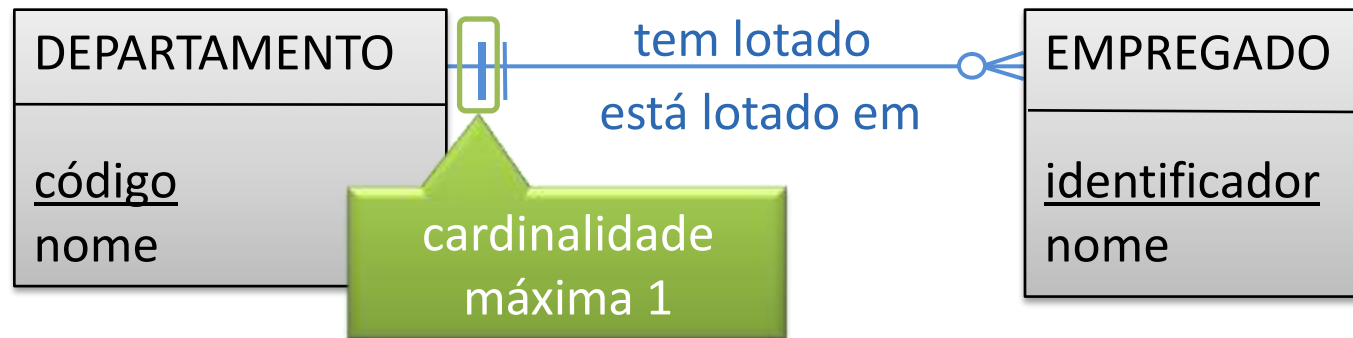
# Notação acadêmica (P. Chen)



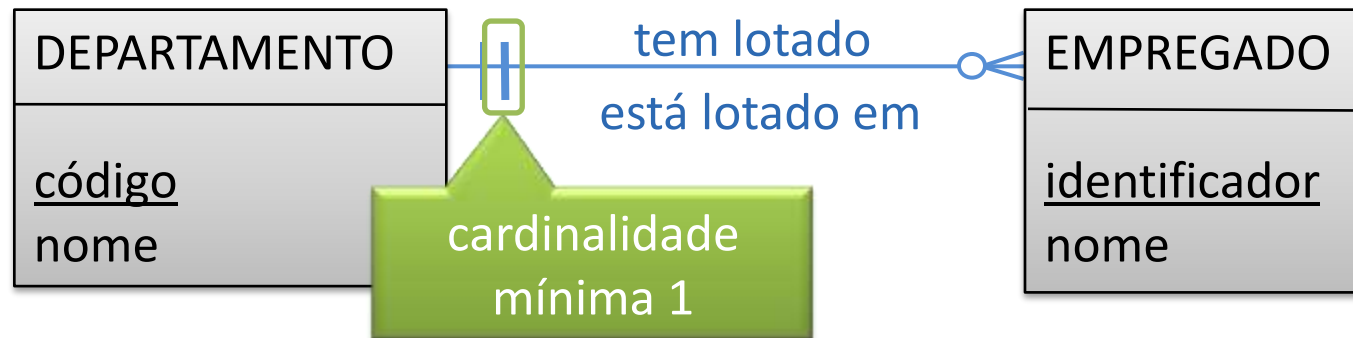
# Notação Engenharia de Informações



# Notação Engenharia de Informações

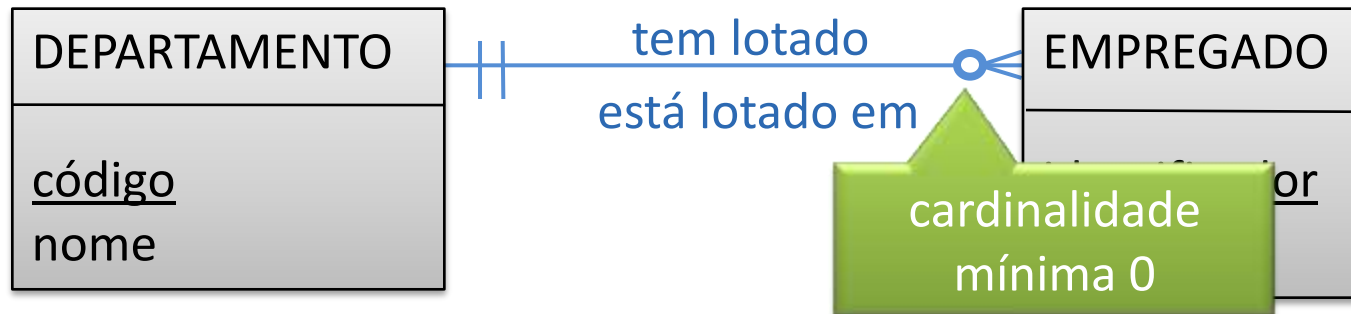


# Notação Engenharia de Informações

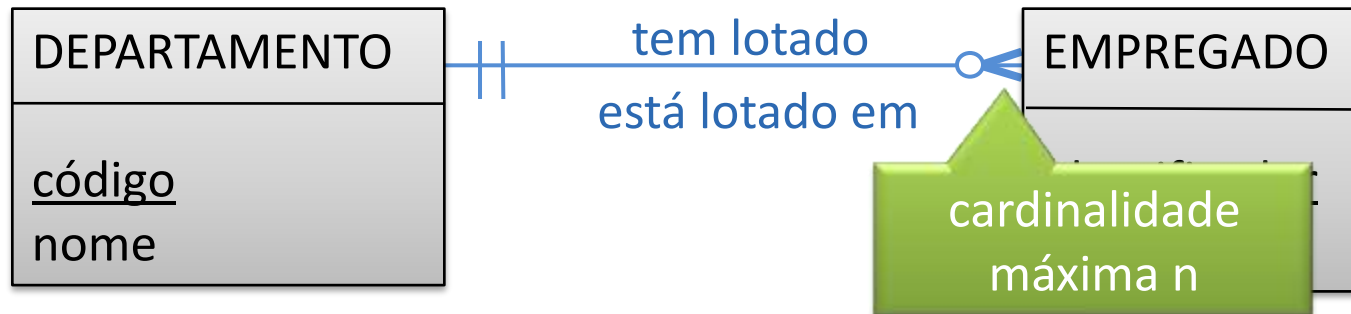




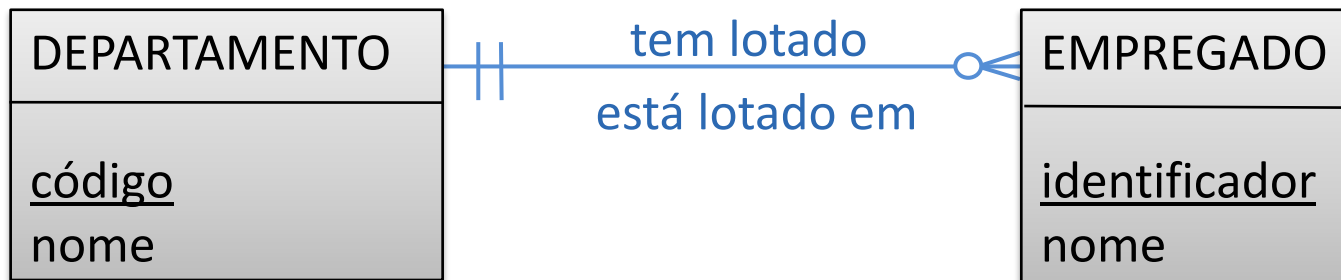
# Notação Engenharia de Informações



# Notação Engenharia de Informações



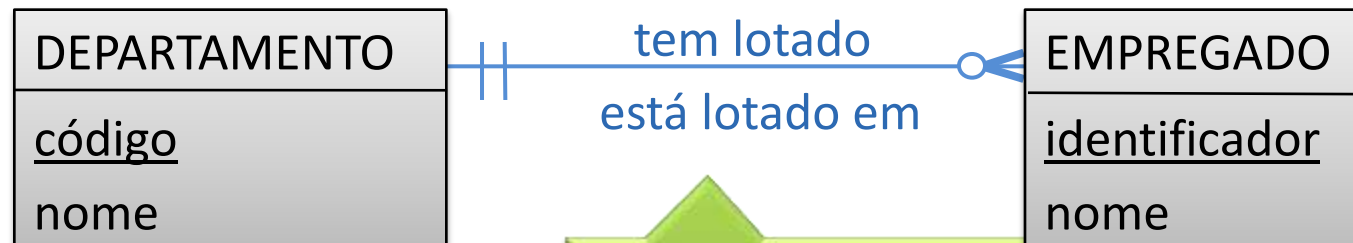
# Notação Engenharia de Informações



Notação para cardinalidade máxima e mínima:

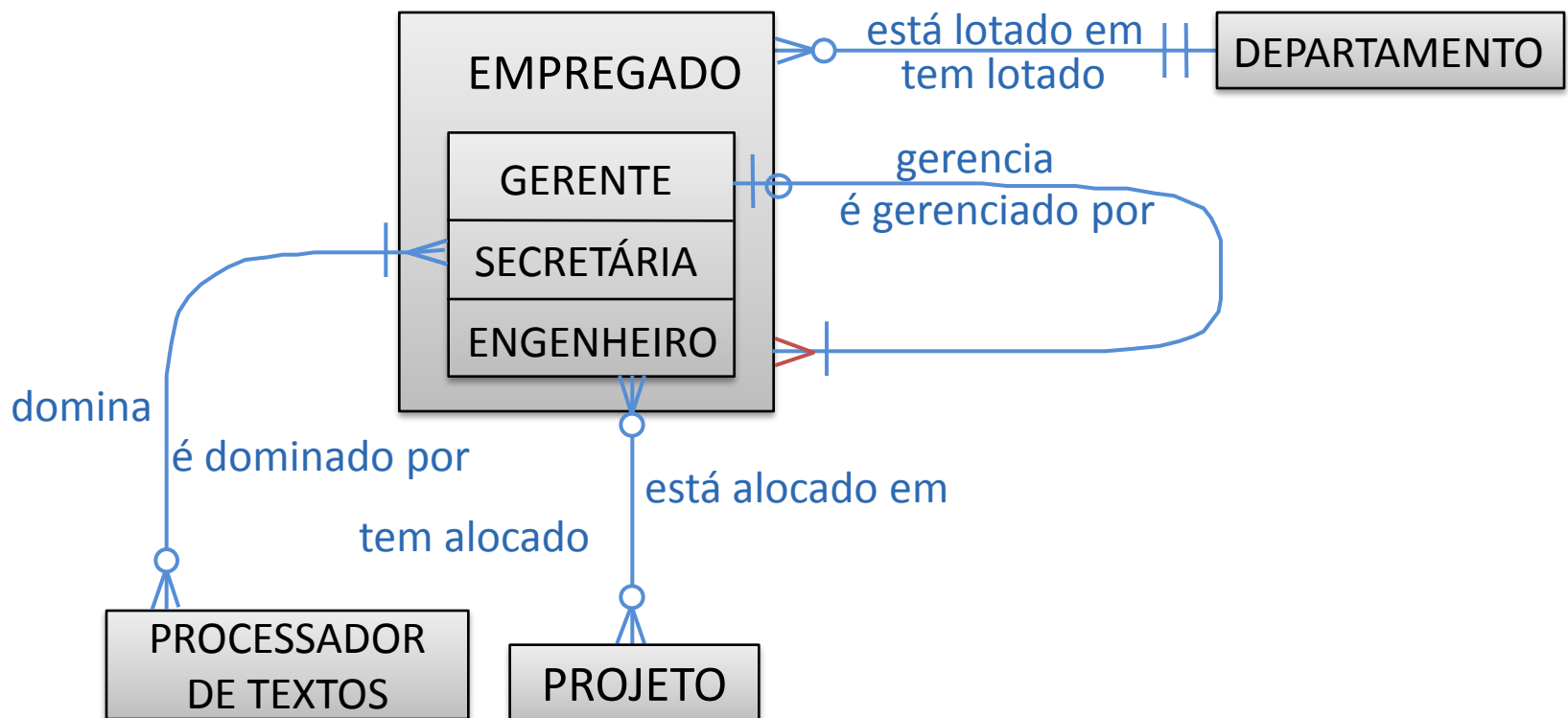
- | Cardinalidade (mínima, máxima) 1
- Cardinalidade mínima 0
- ≧ Cardinalidade máxima n

# Notação Engenharia de Informações



relacionamentos  
binários somente:  
sem atributos

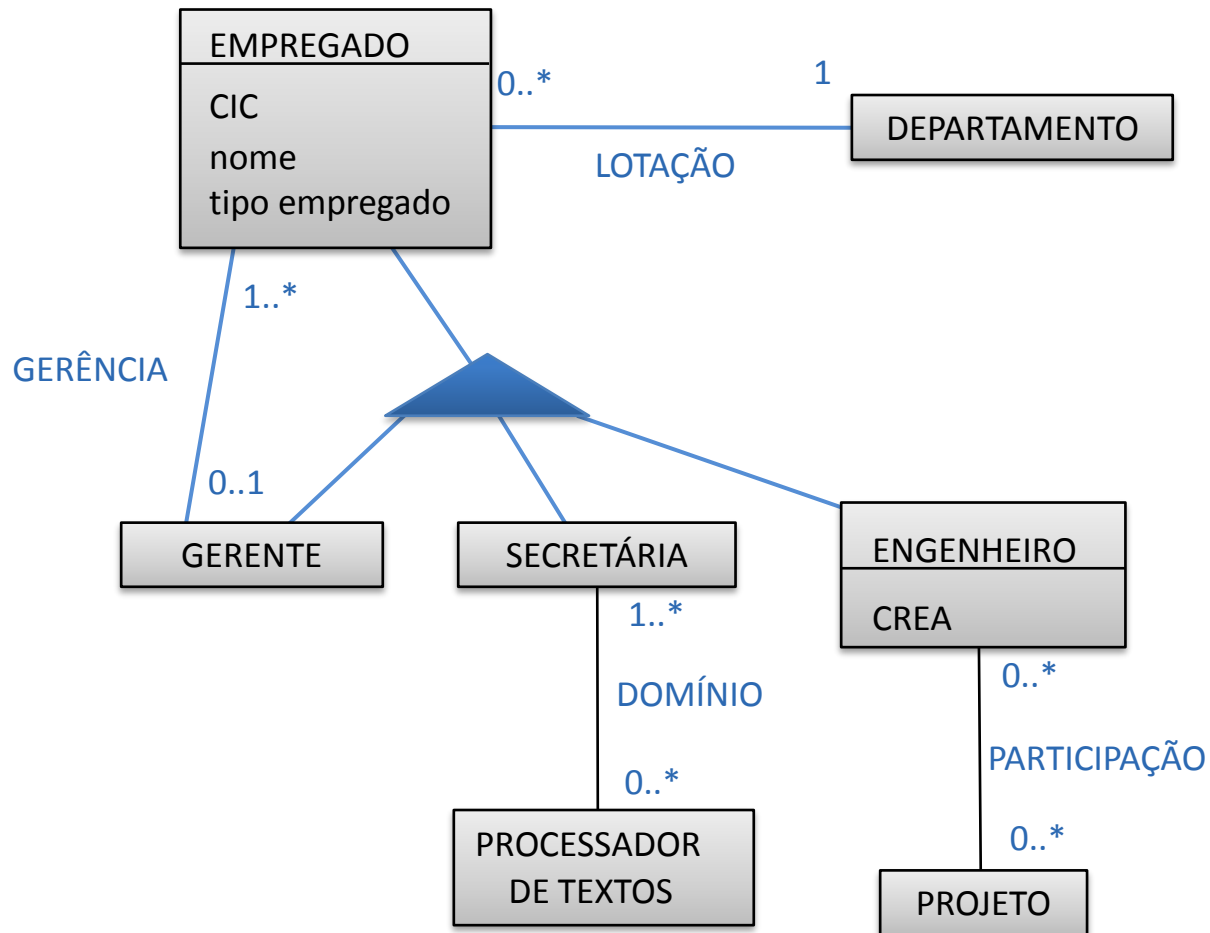
# Engenharia de informações especialização: subtipos de entidades



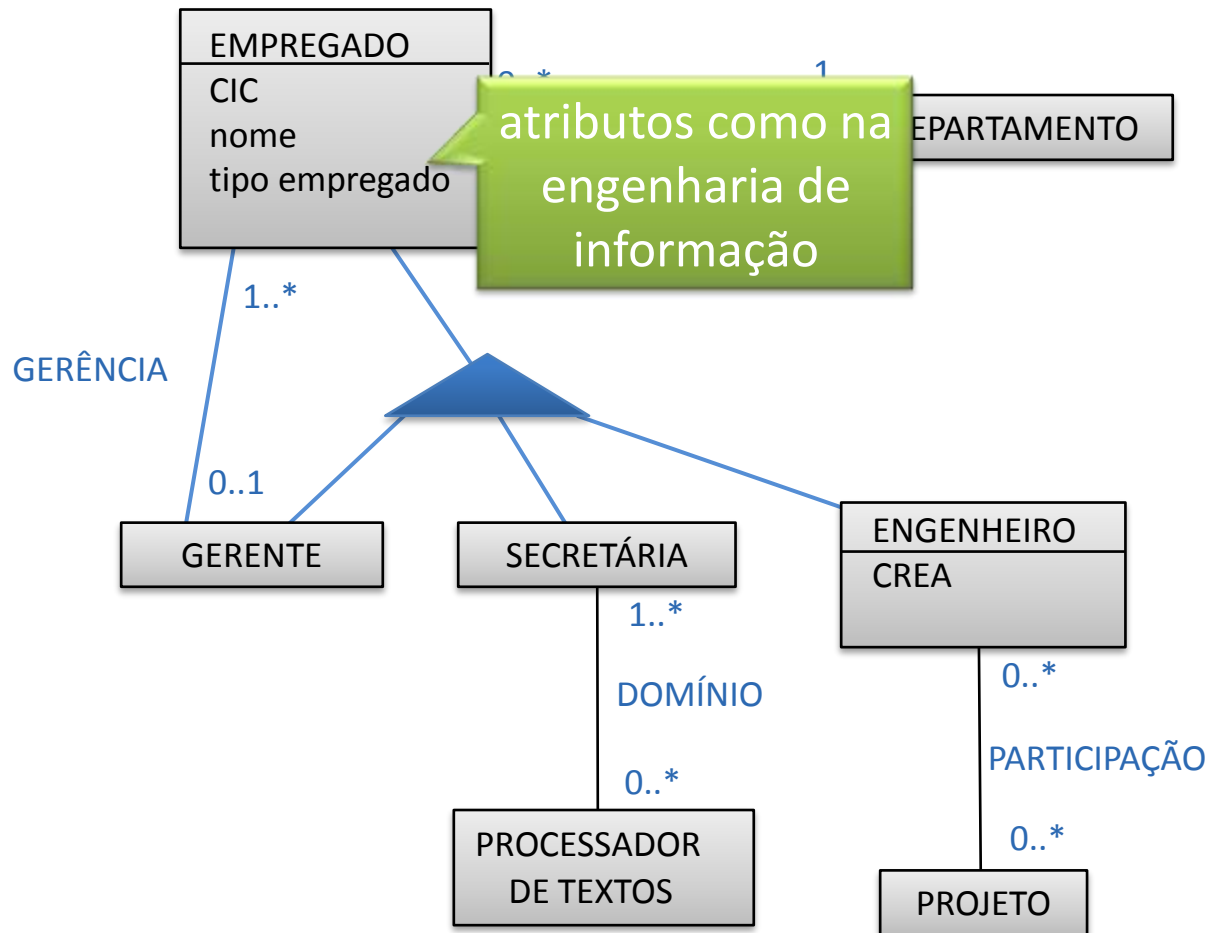
# UML

- UML (“unified modeling language”):
  - Conjunto de modelos diagramáticos para modelagem de software.
- Diagrama de classes pode ser usado como modelo conceitual.
- Terminologia é diferente:

ER	UML
Entidade	Classe
Relacionamento	Associação
Cardinalidade	Multiplicidade
Generalização/especialização	Generalização

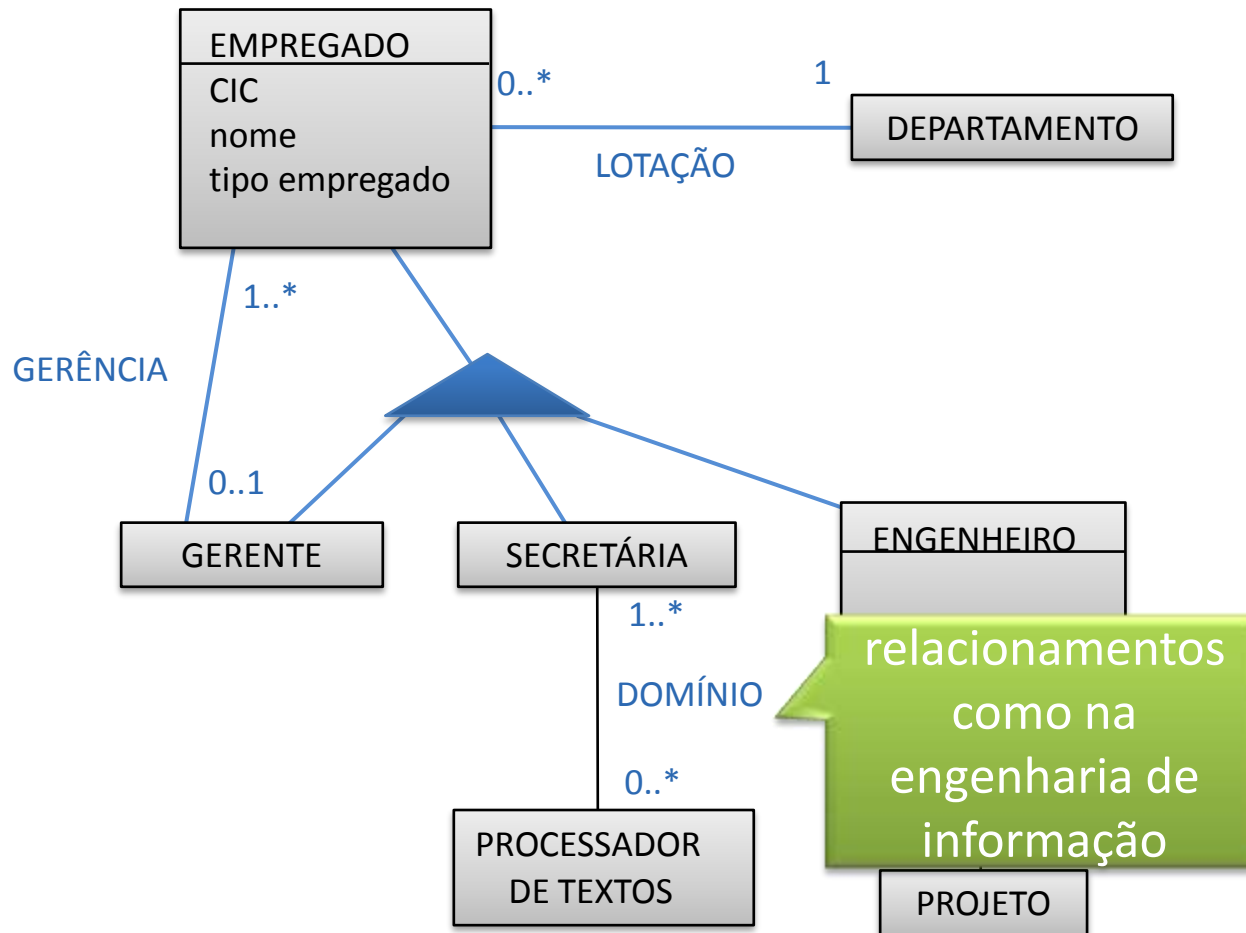


# UML - notação

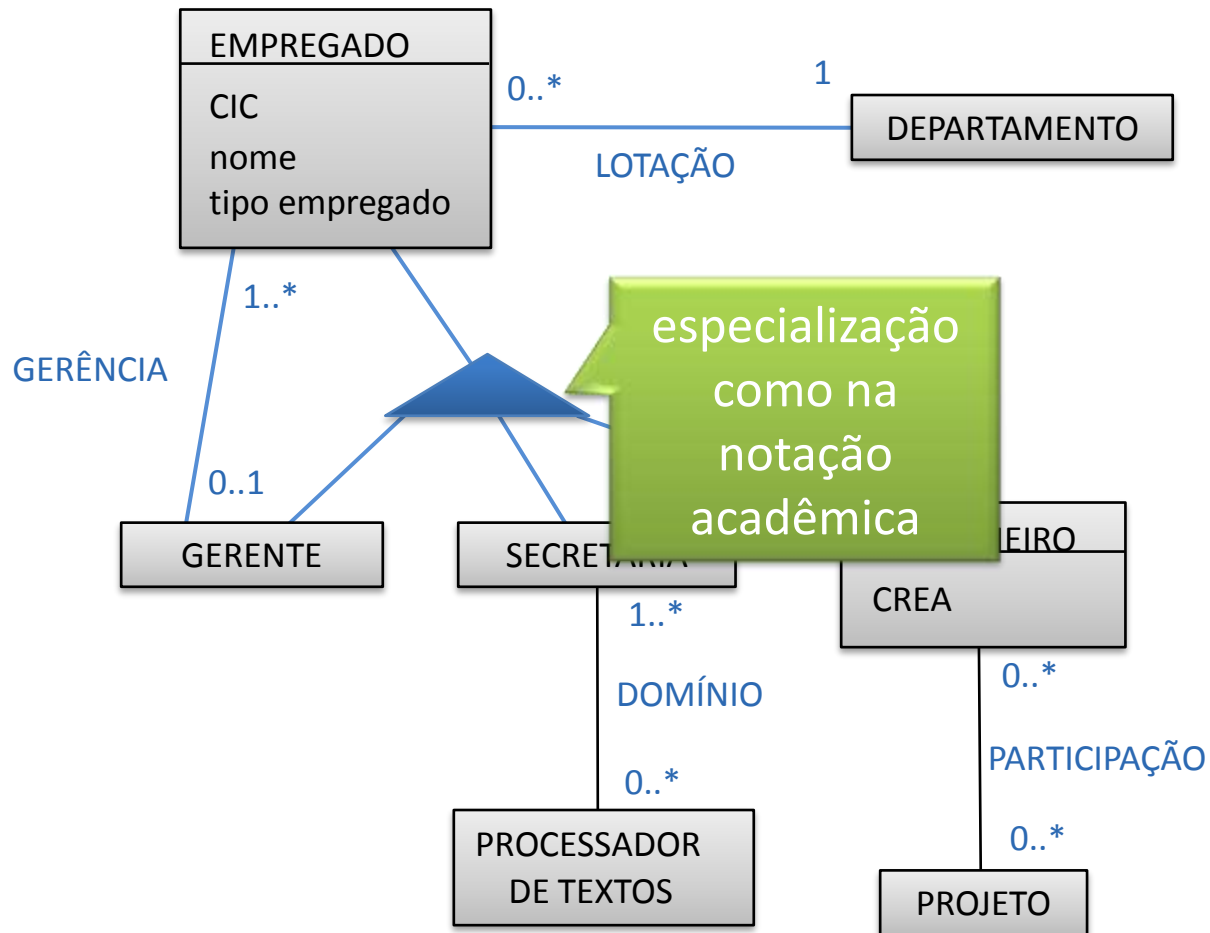




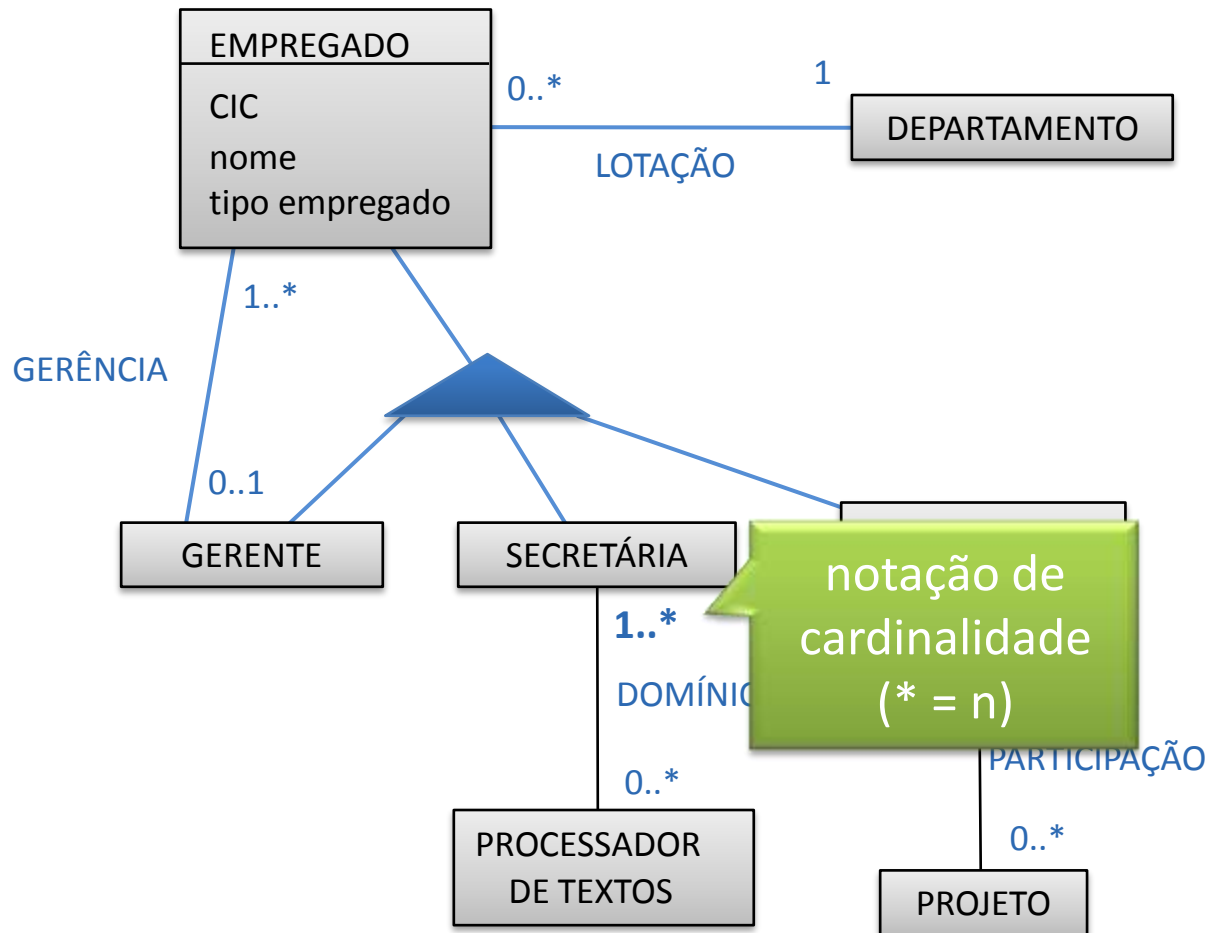
# UML - notação



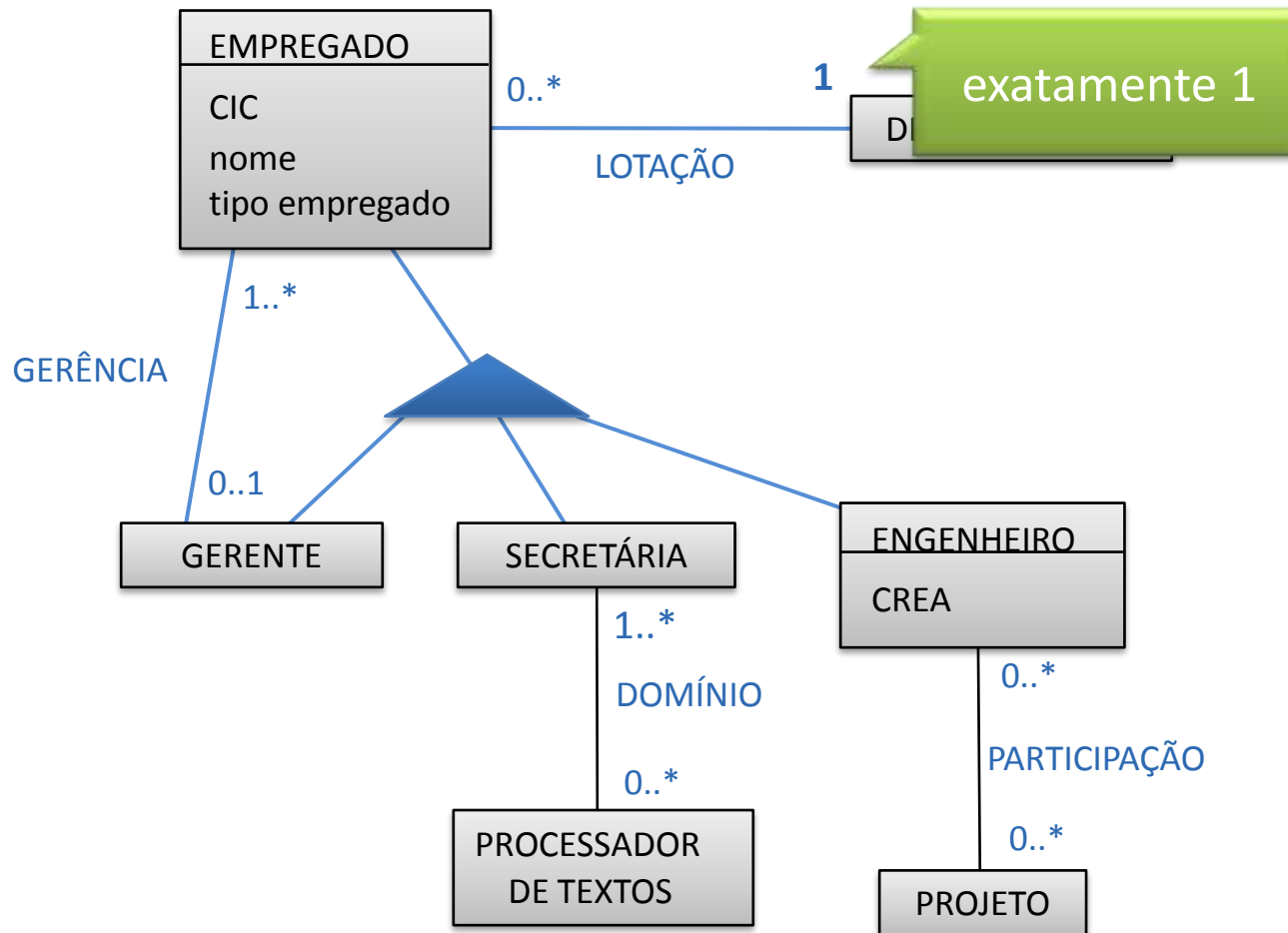
# UML - notação



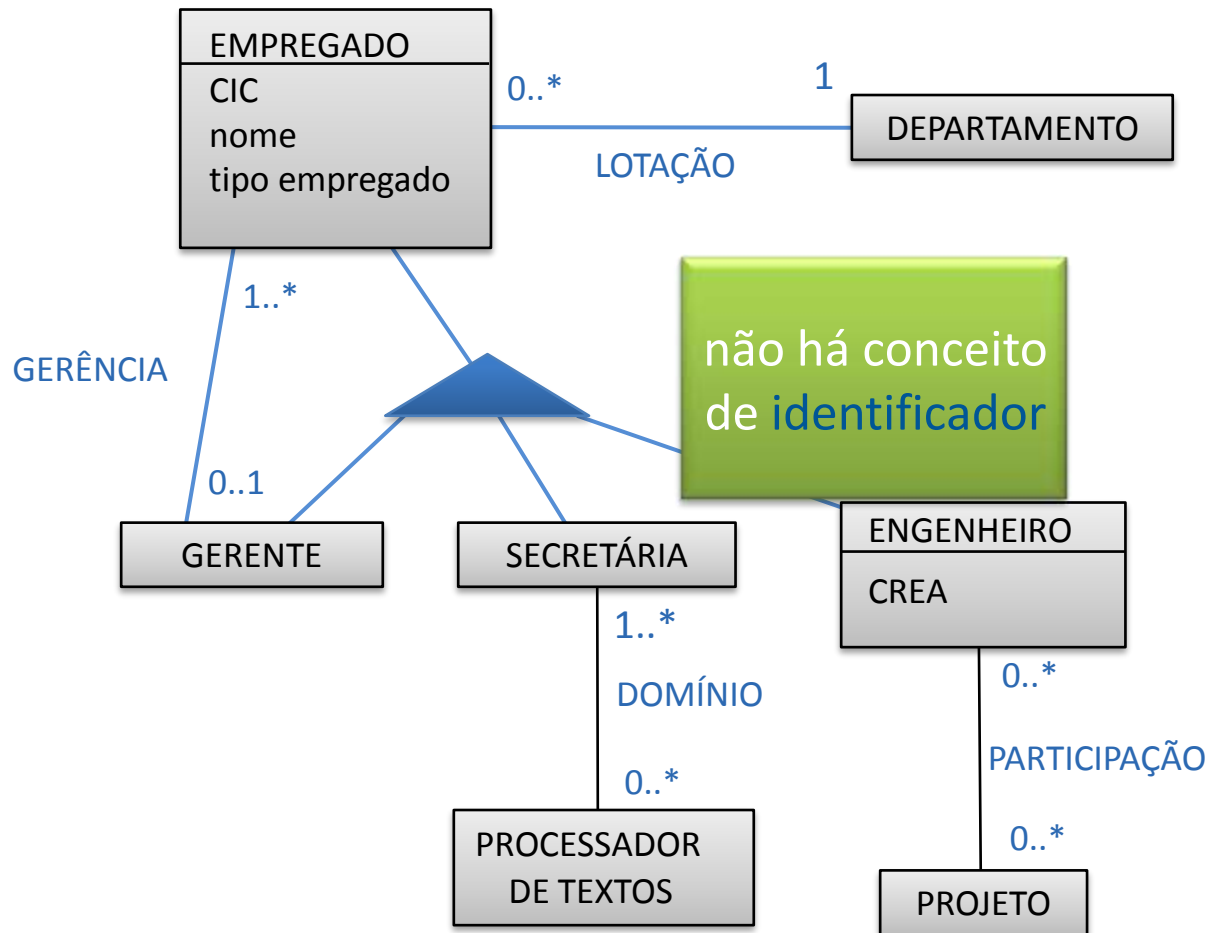
# UML - notação



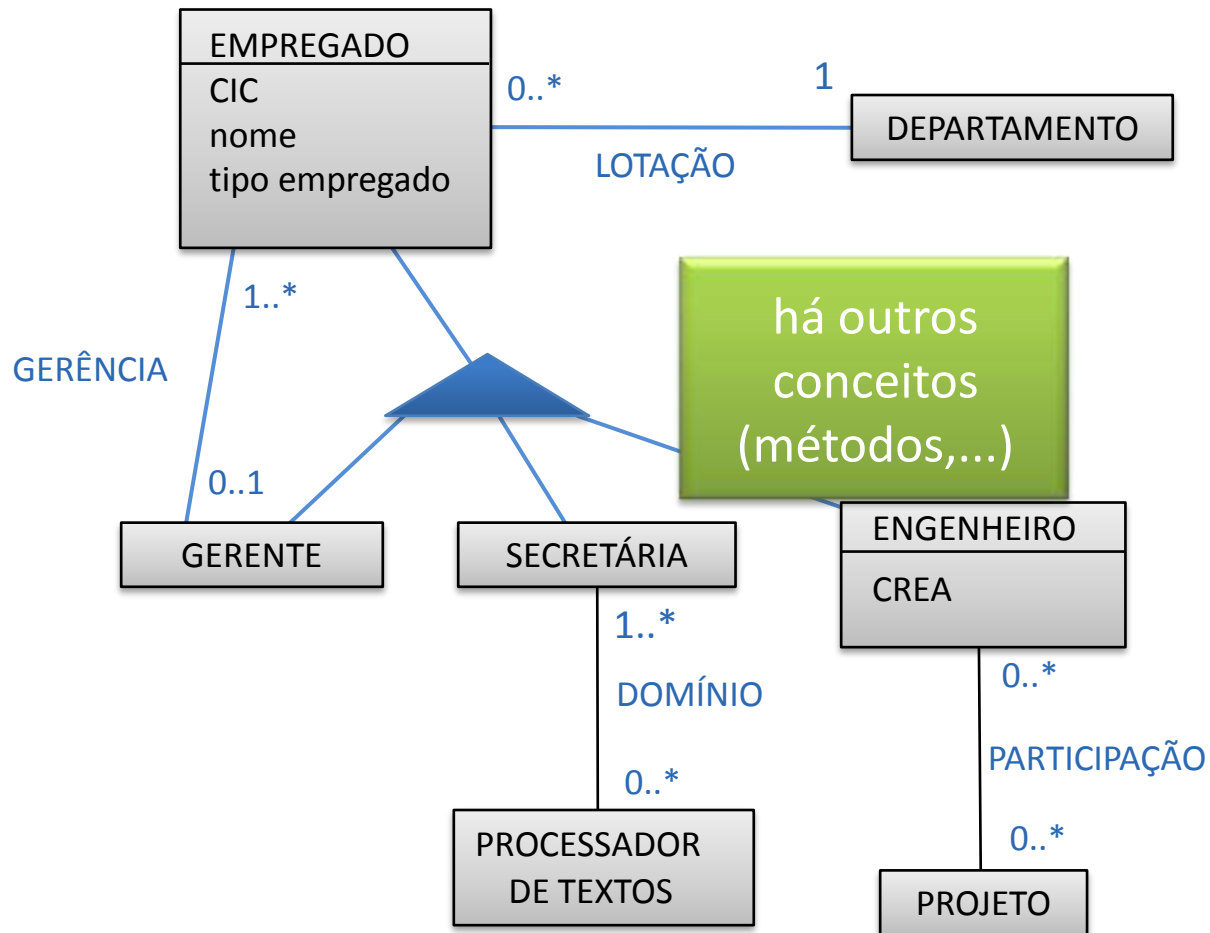
# UML - notação



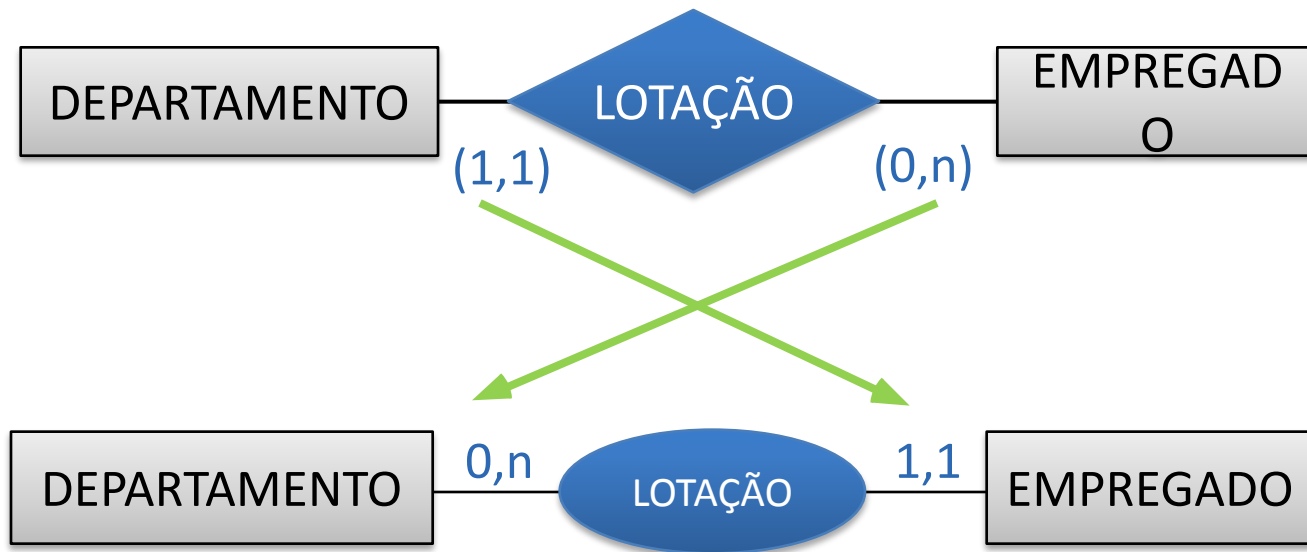
# UML - notação



# UML - notação



# Notação Européia (MERISE, DBMain,...)



# Notação Européia (MERISE, DBMain,...)

semântica associativa:

cardinalidade diz  
quantas vezes uma instância de  
entidade participa de um  
relacionamento





# Uso de ferramentas de modelagem

- Diagrama ER não deve ser confeccionado manualmente:
  - muito **trabalhoso**,
  - **revisões** são freqüentes,
  - diagramas feitos à mão não são atualizados, quando de alterações do esquema.
- Recomendável que seja usada uma **ferramenta em computador** para apoio à modelagem.
- Alternativas:
  - Uso de uma **ferramenta CASE**.
  - Uso de programas de propósito geral.

# Estratégias de modelagem

- Estratégia de modelagem ER:
  - uma **seqüência de passos** (uma “receita-de-bolo”) de **transformação** de modelos, desde o modelo inicial de modelagem, até o final.
- Diferentes estratégias:
  - *Bottom-up*
  - *Top-down*
  - *Inside-out*

# Definição da estratégia de modelagem

- Na prática:
  - Nenhuma das estratégias propostas na literatura é universalmente aceita.
- Normal é:
  - Combinação das diversas estratégias de modelagem.
- Compreensível, pois:
  - Processo de modelagem é um processo de aprendizagem.

# Definição da estratégia de modelagem

- Identificar qual a **fonte de informações** principal para o processo de modelagem:
  - **Descrições de dados existentes**:
    - estratégia *bottom-up*.
  - **Conhecimento de pessoas** sobre o sistema :
    - estratégia *top-down* (ou *inside-out*).

## Estratégia “*top-down*”

- Partir de conceitos mais abstratos (“de cima”).
- Ir gradativamente refinando estes conceitos em conceitos mais detalhados.

# Estratégia “*top-down*” processo (1)

## 1. Modelagem superficial:

- a) Enumeração das entidades.
- b) Identificação dos relacionamentos (cardinalidade máxima) e hierarquias de generalização/especialização entre as entidades.
- c) Determinação dos atributos de entidades e relacionamentos.
- d) Determinação dos identificadores de entidades e relacionamentos.
- e) O banco de dados é verificado quanto ao aspecto temporal.

# Estratégia “*top-down*” processo (2)

## 2. Modelagem detalhada:

- a) Domínios dos atributos
- b) Cardinalidades mínimas.
- c) Demais restrições de integridade.

## 3. Validação do modelo:

- a) Construções redundantes ou deriváveis a partir de outras no modelo.
- b) Validação com o usuário.

# Estratégia “*inside-out*”

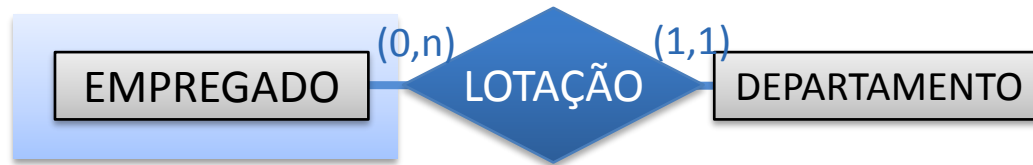
## Exemplo – passo #1

EMPREGADO



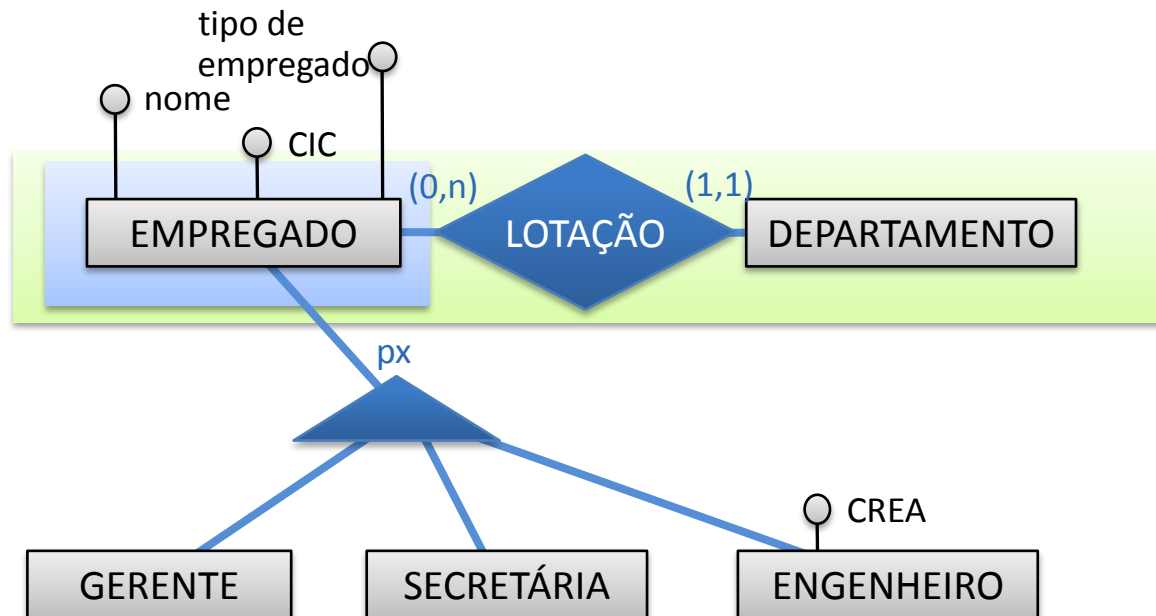
# Estratégia “*inside-out*”

## Exemplo – passo #2



# Estratégia “*inside-out*”

## Exemplo – passo #3



# Estratégia “inside-out”

## Exemplo – passo #4

