

[MAC0439] Laboratório de Bancos de Dados  
Aula 2  
O Projeto Conceitual de Bancos de Dados  
e o Modelo Entidade-Relacionamento

Kelly Rosa Braghetto

DCC-IME-USP

14 de agosto de 2015

# Projeto de bancos dados

Envolve as seguintes etapas:

1. Levantamento e análise dos requisitos
2. Projeto conceitual
3. Projeto lógico
4. Projeto físico

# Fase 1: Levantamento e análise dos requisitos

Nessa fase, o projetista:

- ▶ Registra concisamente os requisitos dos usuários com relação aos dados
- ▶ Define requisitos funcionais conhecidos das aplicações
  - ▶ operações/transações empregadas no BD

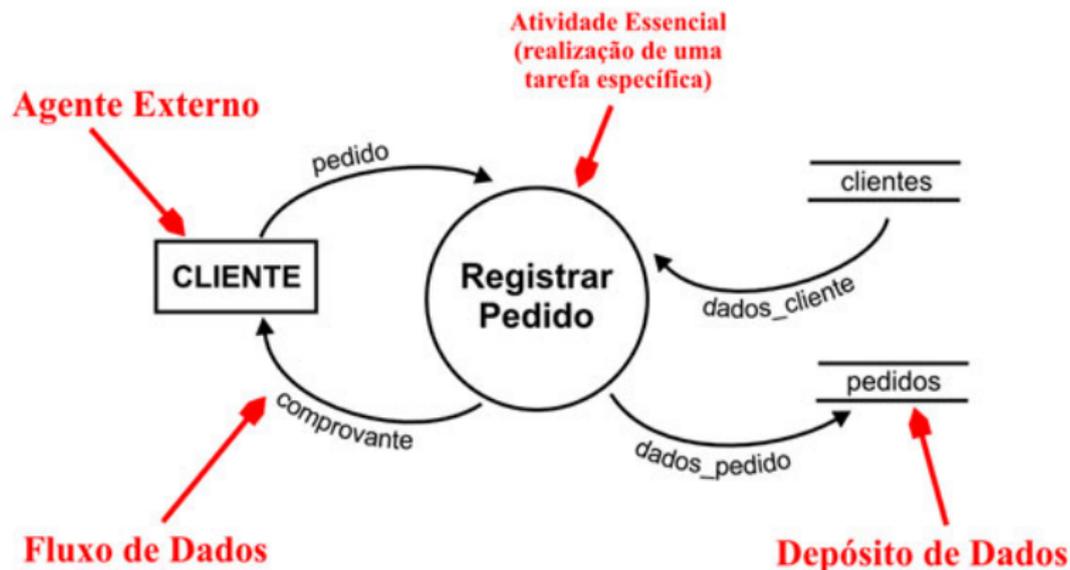
# Fase 1: Levantamento e análise dos requisitos

## Sobre os requisitos:

- ▶ São levantados por meio de entrevistas com os produtores e os usuários dos dados, e podem ser descritos por meio de técnicas para a especificação de requisitos, como os **Diagramas de Fluxos de Dados** (DFDs)
- ▶ Incluem os dados exigidos para processamento, os seus relacionamentos naturais e as informações relevantes para a escolha da plataforma de software para o BD

# Fase 1: Levantamento e análise dos requisitos

## Exemplo de diagrama DFD



Fonte: <http://www.linhadecodigo.com.br/artigo/1293/a-importancia-do-modelagem-de-objetos-no-desenvolvimento-de-sistemas.aspx>

## Fase 2: Projeto conceitual

Fase de criação de um esquema conceitual para o BD, utilizando um modelo de dados conceitual de alto nível.

### Esquema conceitual (definição):

Descrição concisa de requisitos de dados dos usuários, contendo descrições detalhadas sobre os tipos de entidades, relacionamentos e restrições, expressos de acordo com os conceitos existentes no modelo de dados conceitual de alto nível empregado

## Fase 2: Projeto conceitual

### Características de um esquema conceitual:

- ▶ Fácil compreensão (não envolve detalhes de implementação)
- ▶ Pode ser usado na comunicação com usuários não técnicos

### Exemplos bastante usados:

- ▶ Modelo ER (Entidade-Relacionamento)
- ▶ Diagramas de classe da UML

## Fase 3: Projeto lógico

Essa fase corresponde ao mapeamento do esquema conceitual para um modelo de dados de implementação.

⇒ Passo necessário para a implementação do BD utilizando um SGBD comercial.

Exemplos de modelos de implementação bastante usados:

- ▶ modelo relacional
- ▶ modelo objeto-relacional

## Fase 3: Projeto lógico

### Refinamento do esquema:

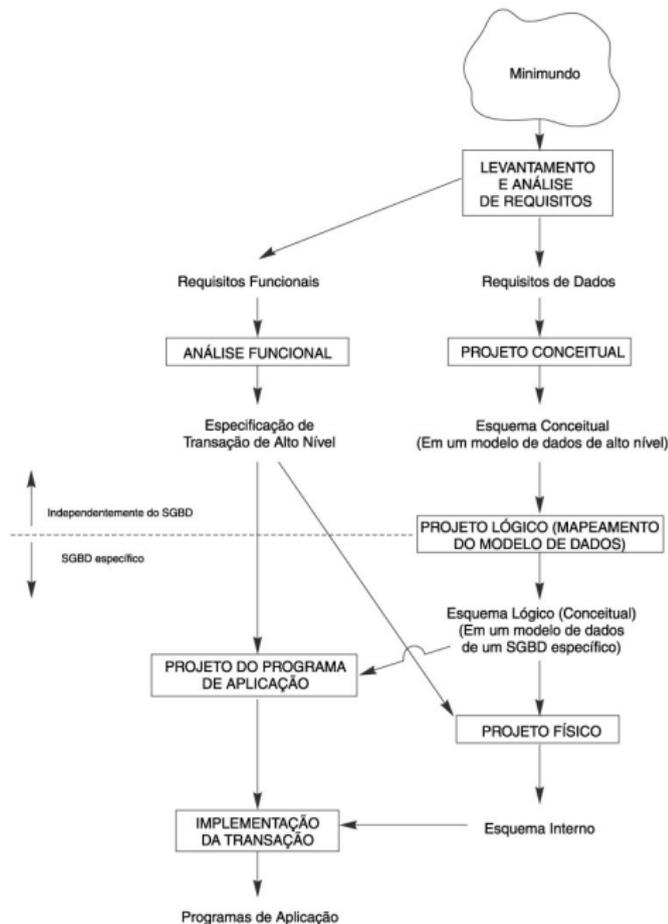
- ▶ O refinamento é uma etapa **opcional** no projeto lógico
- ▶ Ele identifica problemas em potencial no modelo lógico criado e aplica técnicas para melhorar o modelo
- ▶ Exemplo: no modelo relacional, é necessária uma etapa de **normalização** das relações
  - ▶ A normalização tem como objetivo evitar esquemas de BDs em que anomalias de alterações dos dados possam ocorrer

## Fase 4: Projeto físico

Fase na qual são definidas as estruturas de armazenamento interno, índices, caminhos de acesso, organizações de arquivos para os arquivos do BD e outros ajustes finos.

- ▶ Finalidade: otimizar o desempenho das operações de consulta e manipulação dos dados
- ▶ Pode até mesmo modificar o projeto de BD resultante das fases anteriores, a fim de satisfazer critérios de desempenho desejados
- ▶ Exemplo: “denormalização” das relações em *Data Warehouses* (bancos de dados analíticos, em que o desempenho para os operações de consulta é um requisito importante)

## Principais fases do projeto de um banco de dados



# Abstrações de dados

A modelagem conceitual de dados está relacionada à representação dos dados por meio de um conjunto de abstrações.

Abstrações têm:

- ▶ **Estrutura:** do ponto de vista estrutural, abstrações são detalhes que deliberadamente se omitem numa dada visão de implementação para facilitar a visão global do tópico em questão.
- ▶ **Comportamento:** do ponto de vista comportamental, abstrações determinam restrições, regras e operações que devem ser tomadas (automaticamente) na manipulação do conjunto de elementos.

# Modelo Entidade-Relacionamento (ER)

- ▶ Criado por Peter Chen em 1976
- ▶ É simples e de interpretação intuitiva (mesmo para usuários não-especialistas)
- ▶ É modelo mais usado para a modelagem conceitual de BDs
- ▶ Descreve os dados com base em três conceitos principais:
  - ▶ **entidades**
  - ▶ **relacionamentos**
  - ▶ **atributos**

# Entidades e Atributos

- ▶ **Entidade** – “algo” do mundo real, com uma existência independente. Pode ter existência física (ex.: uma pessoa, uma casa, um carro, etc.) ou conceitual (ex.: um projeto, uma disciplina).
- ▶ **Atributos** – propriedades particulares que descrevem uma entidade (ex.: nome da pessoa, endereço, função, etc.). Um dada entidade terá um valor para cada um de seus atributos. Esses valores constituem a maior parte dos dados armazenados em um BD.

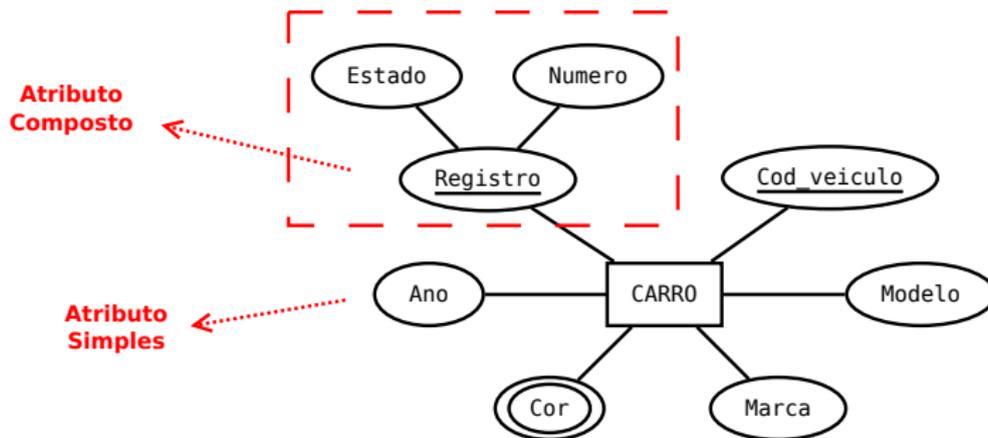
# Tipos de atributos no modelo ER

## Simplex (atômicos) × Compostos

- ▶ **Atributos compostos** – podem ser divididos em partes menores. Ex.: o atributo endereço pode ser dividido em Rua, Cidade, Estado e CEP.
  - ▶ O seu valor é dado pela concatenação dos valores dos atributos simples que os compõem.
  - ▶ Devem ser usados quando o usuário algumas vezes se refere ao atributo como um grupo, mas em outras se refere a um de seus componentes.  
(Caso contrário, o endereço, por exemplo, poderia ser definido como um atributo simples).
- ▶ **Atributos simples (ou atômicos)** – os atributos que não são divisíveis.

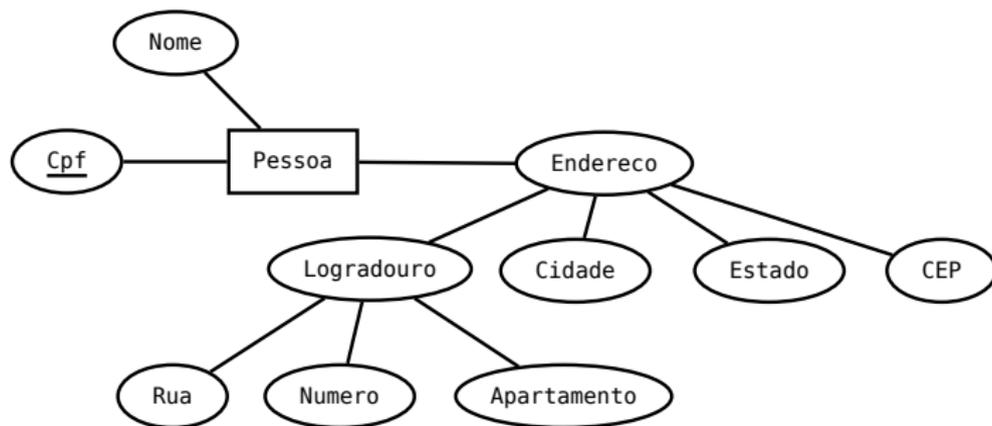
# Notação para tipos de entidades e atributos no modelo ER

## Exemplo de atributos simples e compostos



# Tipos de atributos no modelo ER

## Exemplo de hierarquia de atributos compostos



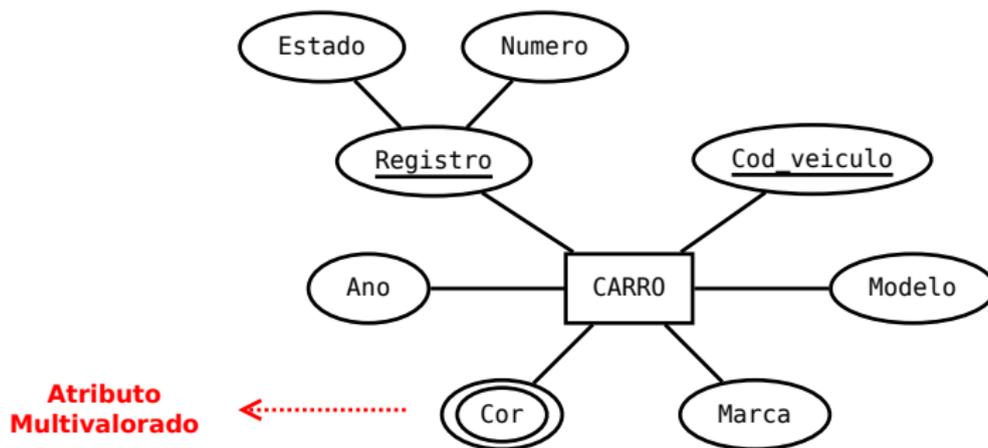
# Tipos de atributos no modelo ER

## Monovalorados × Multivalorados

- ▶ **Monovalorado** – atributo que tem um único valor para uma dada entidade.
  - ▶ **Ex.:** o atributo idade para uma pessoa.
- ▶ **Multivalorado** – atributo para o qual diferentes entidades podem ter diferentes quantidades de valores.
  - ▶ **Ex.:** atributo titulação para uma pessoa; algumas pessoas não possuem título algum, outras possuem 1, ou 2, ...
  - ▶ Um atributo multivalorado deve ter um **limite inferior e superior** pra restringir o número de valores permitidos a cada entidade individual.

# Notação para tipos de entidades e atributos no modelo ER

## Exemplo de atributo multivalorado



# Tipos de atributos no modelo ER

## Armazenados × Derivados

- ▶ **Atributo derivado** – é derivado a partir de outro(s) atributo(s) ou entidade(s) relacionado(s). Exemplos:
  - ▶ o atributo **idade** é derivado a partir da data de nascimento e da data corrente
  - ▶ o atributo **número de empregados** de um departamento pode ser derivado da contagem do número de empregados relacionados (que trabalham) nesse departamento
- ▶ **Atributo armazenado** – é um atributo que não é derivado.

# Notação para tipos de entidades e atributos no modelo ER

## Exemplo de atributo derivado



# Tipos de atributos do modelo ER

## Atributos Complexos

São atributos compostos e multivalorados, que podem ser aninhados de modo arbitrário.

- ▶ **Ex.:** se uma pessoa pode ter mais de uma residência, e cada uma delas pode ter múltiplos telefones, um atributo `EnderecoFone` pode ser especificado como a seguir

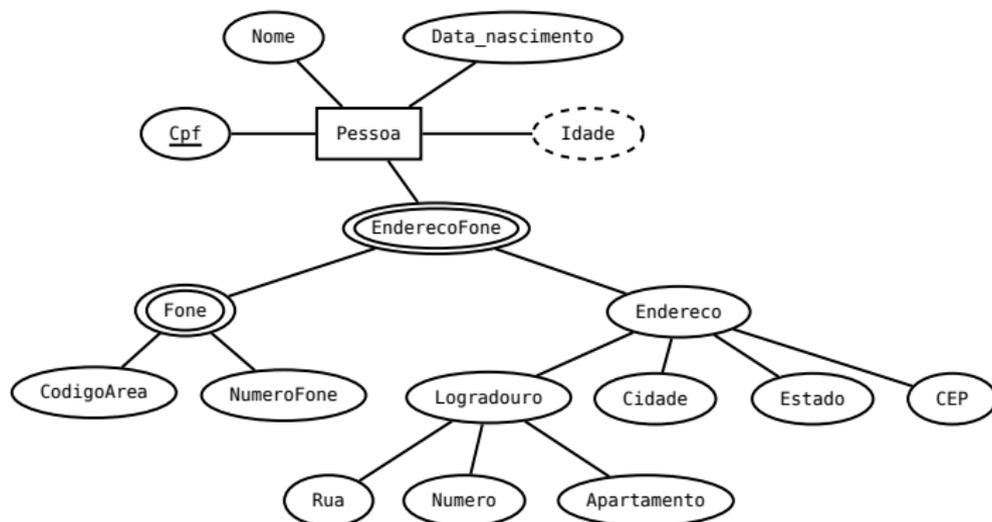
```
{EnderecoFone({Fone(CodigoArea,NumeroFone)},  
               Endereco(Logradouro(Numero,Rua,Apartamento),  
                         Cidade,Estado,CEP))}
```

## Notação

- ▶ `{atributo}` – atributo multivalorado
- ▶ `(atrib1,atrib2,atrib3,...)` – atributo composto

# Notação para tipos de entidades e atributos no modelo ER

## Exemplo de atributo complexo



# Tipos de atributos do modelo ER

## Valor NULL (nulo)

- ▶ É um valor especial, usado quando uma entidade **não possuiu um valor** para um atributo.
- ▶ O NULL serve tanto para indicar que um atributo **não se aplica** a uma dada entidade, quanto para indicar que o valor para um atributo de uma dada entidade é **desconhecido**.
- ▶ “Desconhecido” se aplica a dois casos distintos:
  - ▶ quando é sabido que existe um valor para o atributo, mas ele está faltando (**ex.:** Altura – todo mundo tem!)
  - ▶ quando não é sabido se o valor existe ou não (**ex.:** FoneResidencial – uma pessoa pode ou não ter)

# Tipo de entidade

Um **tipo de entidade** define um conjunto de entidades que possuem os mesmos atributos.

## Conceitos importantes:

- ▶ **Atributo-chave (restrição de exclusividade)** – é um atributo cujos valores são distintos para toda entidade pertencente ao conjunto de entidades do tipo.
  - ▶ Os valores do atributo-chave **identificam cada entidade univocamente**.

# Tipo de entidade

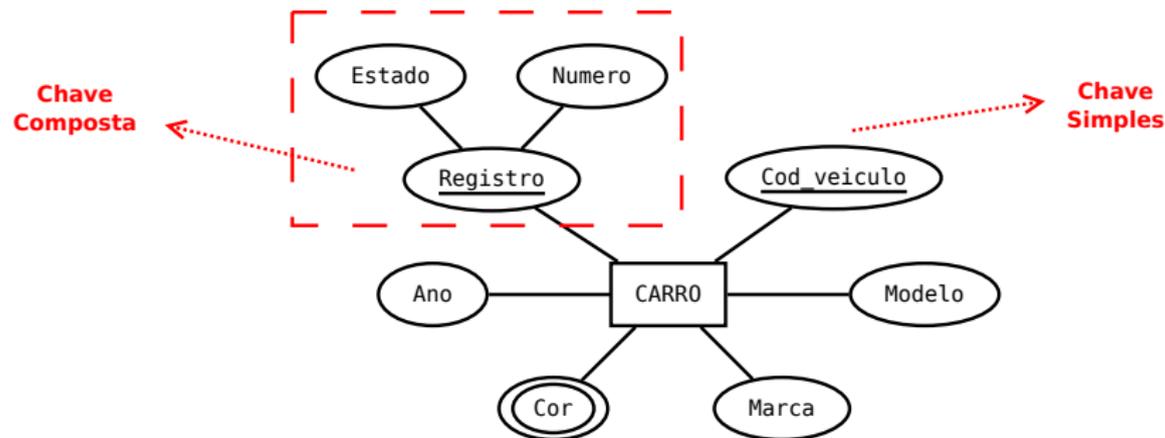
Um **tipo de entidade** define um conjunto de entidades que possuem os mesmos atributos.

## Conceitos importantes:

- ▶ **Chave composta** – é uma chave formada por diversos atributos, cuja a combinação dos valores é distinta para cada entidade. Uma chave composta deve ser mínima, no sentido de que todos os atributos componentes devem ser considerados na composição para garantir a unicidade.
- ▶ Alguns tipos de entidade têm mais de um atributo chave (ex.: NUSP e CPF). Outros, podem nem ter uma chave – são os chamados **tipos de entidade fraca**.

# Notação para tipos de entidades e atributos no modelo ER

## Exemplo de atributos chave



# Tipo de Entidade

Exemplo de dois tipos de entidade e algumas entidades membro de cada um deles:

**NOME DO TIPO ENTIDADE:**

**EMPREGADO**

Nome, Idade, Salario

**EMPRESA**

Nome, Sede Administrativa, Presidente

**CONJUNTO DE ENTIDADE:  
(EXTENSÃO)**

$e_1$

(John Smith, 55, 80k)

$e_2$

(Fred Brown, 40, 30K)

$e_3$

(Judy Clark, 25, 20K)

⋮

$c_1$

(Sunco Oil, Houston, John Smith)

$c_2$

(Fast Computer, Dallas, Bob King)

⋮

# Notação para tipos de entidades e atributos no modelo ER

## Conjunto de entidades do tipo de entidade CARRO

CARRO  
Registro(NumeroRegistro, Estado), IDVeiculo, Marca, Modelo, Ano, {Cor}

car<sub>1</sub> •  
((ABC 123, TEXAS), TK629, Ford Mustang, conversível, 1998, {vermelho, preto})

car<sub>2</sub> •  
((ABC 123, NOVA YORK), WP9872, Nissan Maxima, 4 portas, 1999, {azul})

car<sub>3</sub> •  
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4 portas, 1995, {branco, azul})

⋮

## Domínio de valores dos atributos

- ▶ Cada atributo simples de um tipo de entidade possui um **domínio de valores**, que determina o conjunto de valores válidos que ele pode assumir em cada entidade.
- ▶ Os domínios de valores não são mostrados no diagrama ER.

## Tipo de relacionamento

Um **tipo de relacionamento**  $R$  entre  $n$  tipos de entidades  $E_1, E_2, \dots, E_n$  define um conjunto de associações (= relacionamentos) entre essas entidades.

- ▶ Matematicamente, um tipo de relacionamentos  $R$  é um conjunto de instâncias de relacionamento  $r_i$ , em que cada  $r_i$  associa-se a  $n$  entidades individuais  $(e_1, e_2, \dots, e_n)$ , e cada entidade  $e_j$  em  $r_i$  é um membro do tipo de entidade  $E_j$ ,  $1 \leq j \leq n$ .
- ▶ Cada tipo de entidade  $E_j$  é dito **participante** do relacionamento  $R$ , da mesma forma que cada entidade  $e_j$  é dita participante da instância de relacionamento  $r_i$ .
- ▶ Informalmente, cada  $r_i$  em  $R$  representa uma associação de entidades existente no minimundo, que inclui exatamente uma entidade de cada tipo de entidade participante.

## Notação para tipos de relacionamento

Exemplo: relacionamento TRABALHA\_PARA entre os tipos de entidade EMPREGADO e DEPARTAMENTO



## Propriedades de um tipo de relacionamento

- ▶ **Grau** – é o número de entidades que participam do relacionamento.
  - ▶ grau 2 = relacionamento binário (mais frequente)
  - ▶ grau 3 = relacionamento ternário
  - ▶ ...
- ▶ **Nome do papel** – indica o papel que cada tipo de entidade desempenha em um tipo de relacionamento.
  - ▶ **Ex.:** no tipo de relacionamento TRABALHA\_PARA, o EMPREGADO desempenha o papel de *trabalhador*, enquanto o DEPARTAMENTO desempenha o papel de *empregador*.

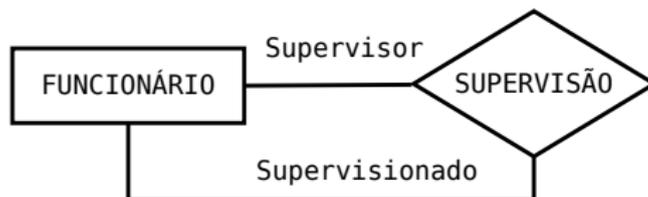


# Propriedades de um tipo de relacionamento

## Nomes dos papéis

- ▶ Só são estritamente necessários quando um mesmo tipo de entidade pode participar mais de uma vez em um mesmo tipo de relacionamento (= **relacionamento recursivo**).
  - ▶ Nesse caso, eles são fundamentais para definir o sentido de cada participação.

Ex.: tipo de relacionamento SUPERVISÃO, em que o tipo de entidade EMPREGADO participa duas vezes – uma no papel de *supervisor*, outra no papel de *supervisionado*.



# Restrições sobre tipos de relacionamento binários

Existem restrições (determinadas por situações do minimundo) que limitam as combinações de entidades que podem participar de um relacionamento binário.

Ex.: um empregado pode apenas trabalhar para um único departamento.

## Restrições possíveis para relacionamentos binários:

- ▶ Razão de cardinalidade
- ▶ Restrição de participação

# Restrições sobre tipos de relacionamento binários

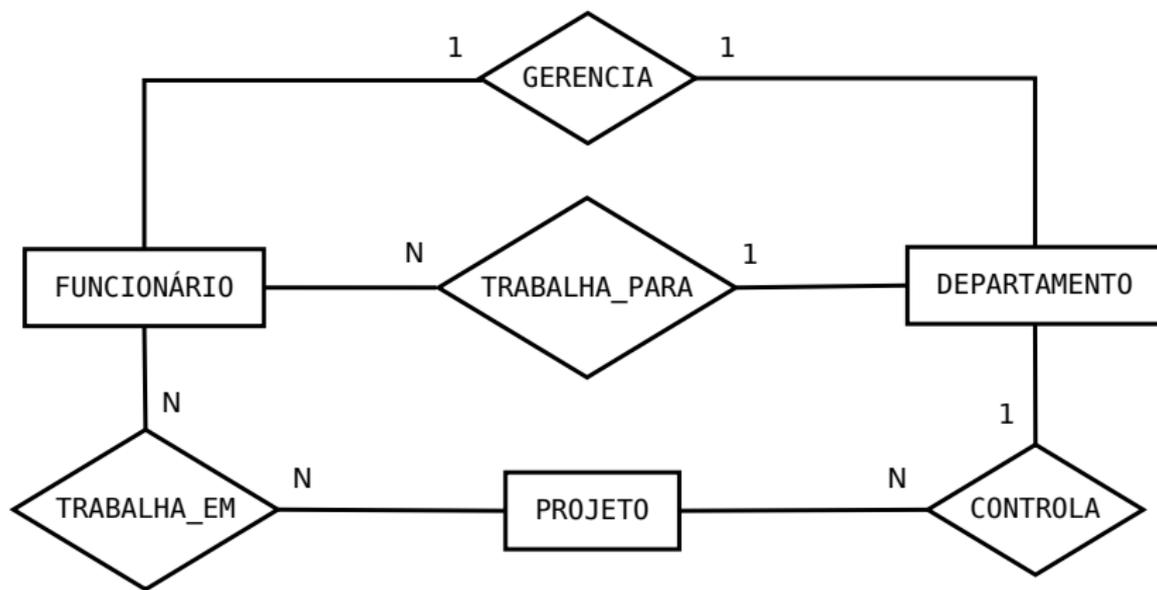
## Razão de cardinalidade

- ▶ Especifica o **número máximo** de instâncias do relacionamento em que uma entidade pode participar.
- ▶ As razões de cardinalidade possíveis são 1:1, 1:N, N:1 e M:N.

Ex.: no relacionamento TRABALHA\_PARA, DEPARTAMENTO:EMPREGADO tem razão 1:N, ou seja, um departamento pode empregar qualquer número (= 0 ou mais) de empregados, mas um empregado pode trabalhar para apenas 1 departamento.

## Notação da razão da cardinalidade

Exemplos de tipos de relacionamento com diferentes cardinalidades



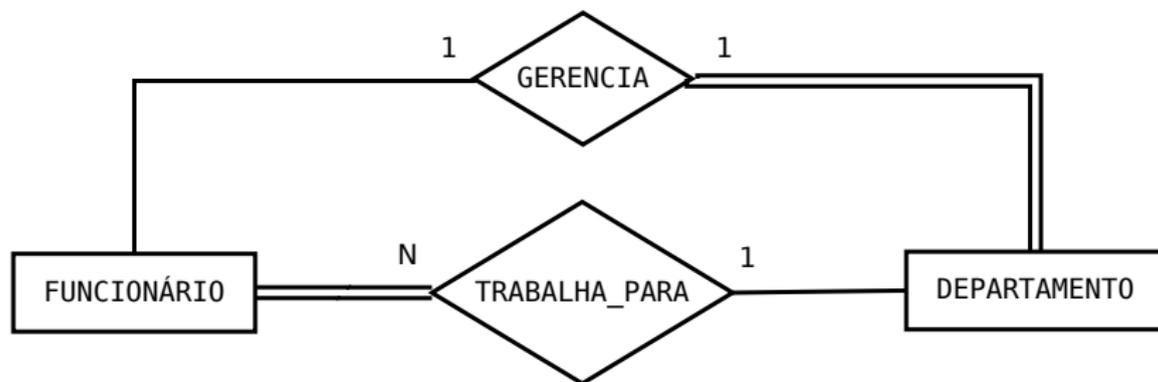
# Restrições sobre tipos de relacionamento binários

## Restrição de participação

- ▶ Determina o **número mínimo** de instâncias de relacionamento em que uma entidade deve participar.
  - ▶ Também é chamada de *restrição de cardinalidade mínima*
  - ▶ **total** – ex.: todo empregado deve trabalhar para um departamento, ou seja, um empregado só pode existir se estiver associado a um departamento. Assim, EMPREGADO tem participação total no relacionamento. Esse tipo de participação também é conhecido como **dependência de existência**.
  - ▶ **parcial** – ex.: nem todo empregado gerencia um departamento. Assim, EMPREGADO tem uma participação parcial no tipo de relacionamento GERENCIA.

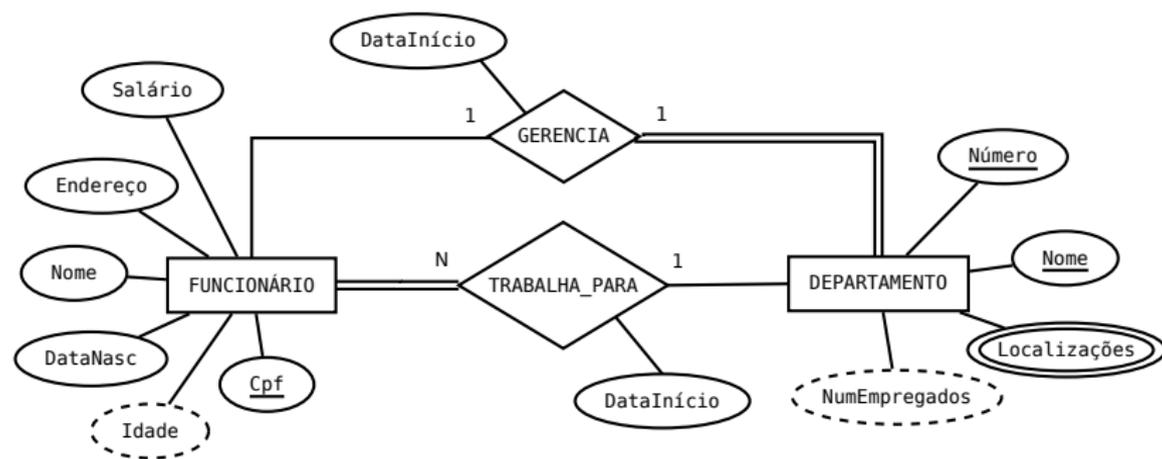
## Notação da restrição de participação

Exemplos de tipos de relacionamento envolvendo diferentes restrições de participação



## Atributos de tipos de relacionamento

Tipos de relacionamento podem ter atributos, de forma similar aos tipos de entidade.



# Atributos de tipos de relacionamento

## Relacionamentos 1:1 ou 1:N

- ▶ Atributos podem ser “migrados” para um dos tipos de entidade participantes. Exemplos:
  - ▶ o atributo `DataInicio` para o tipo de relacionamento `GERENCIA` pode ser um atributo tanto de `EMPREGADO` quanto de `DEPARTAMENTO`, porque esse é um relacionamento 1:1.
  - ▶ o atributo `DataInicio` do tipo de relacionamento 1:N `TRABALHA_PARA` pode apenas ser migrado para o tipo de entidade `EMPREGADO` (ou seja, o lado N do relacionamento), porque um empregado trabalha apenas para um departamento.

# Atributos de tipos de relacionamento

## Relacionamentos M:N

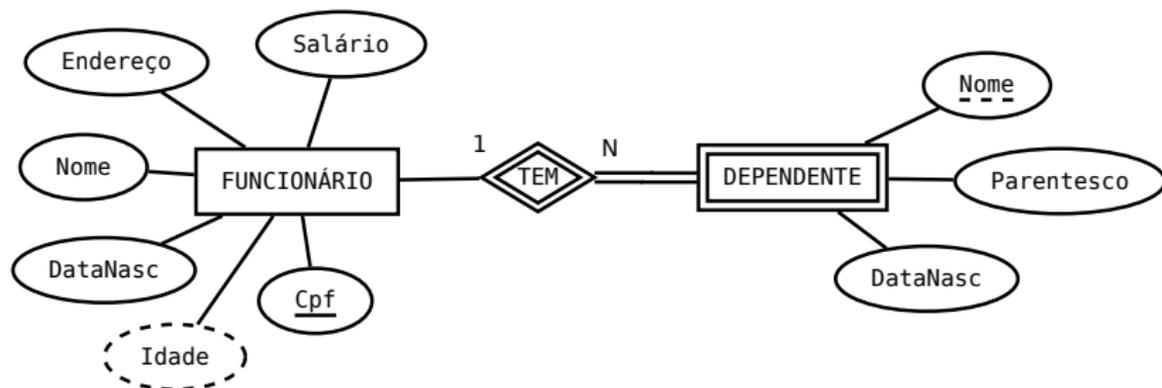
Os atributos são determinados pela combinação das entidades participantes de uma instância do relacionamento, e não por uma entidade única. Por essa razão, eles só podem ser especificados como atributos de relacionamento.

## Tipo de entidade fraca

- ▶ **Tipo de entidade forte** – tipo de entidade que possui um atributo-chave.
- ▶ **Tipo de entidade fraca** – tipo de entidade que não possui um atributo-chave.
  - ▶ Entidades de tipos de entidade fraca são identificadas por estarem relacionadas (associadas) a entidades de um outro tipo de entidade (chamado de **tipo de entidade identificador** ou **tipo de entidade proprietária**).
  - ▶ Esse tipo de relacionamento é chamado de **relacionamento identificador** do tipo de entidade fraca.

## Notação dos tipos de entidade fraca

Exemplo: Tipo de entidade fraca DEPENDENTE em um relacionamento identificador com FUNCIONÁRIO



## Tipo de entidade fraca

- ▶ Um tipo de entidade fraca sempre tem uma restrição de participação total em relação ao seu relacionamento identificador.
- ▶ Um tipo de entidade fraca normalmente tem uma **chave parcial**, que é um conjunto de atributos que identifica univocamente as entidades fracas que estão relacionadas a uma mesma entidade proprietária.
  - ▶ No pior caso, a chave parcial será a composição de todos atributos do tipo de entidade fraca.
- ▶ Quando um tipo de entidade fraca não é participante em tipos de relacionamento, então ele pode ser definido como um atributo complexo (composto, multivalorado) em seu tipo de entidade proprietária.

# Diagrama ER para o esquema EMPRESA

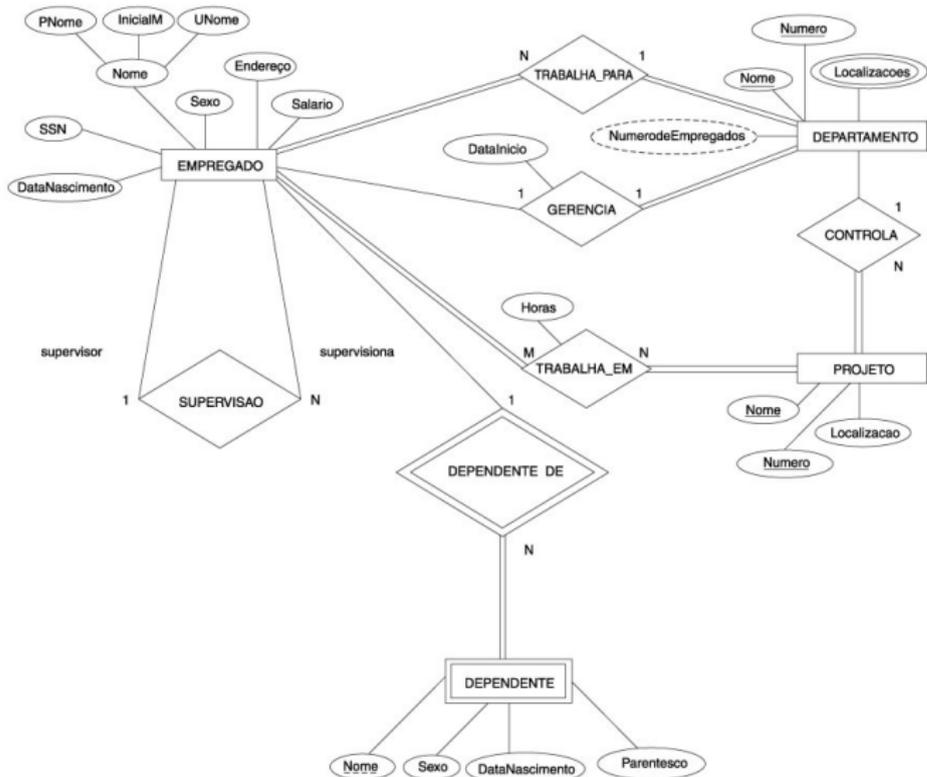


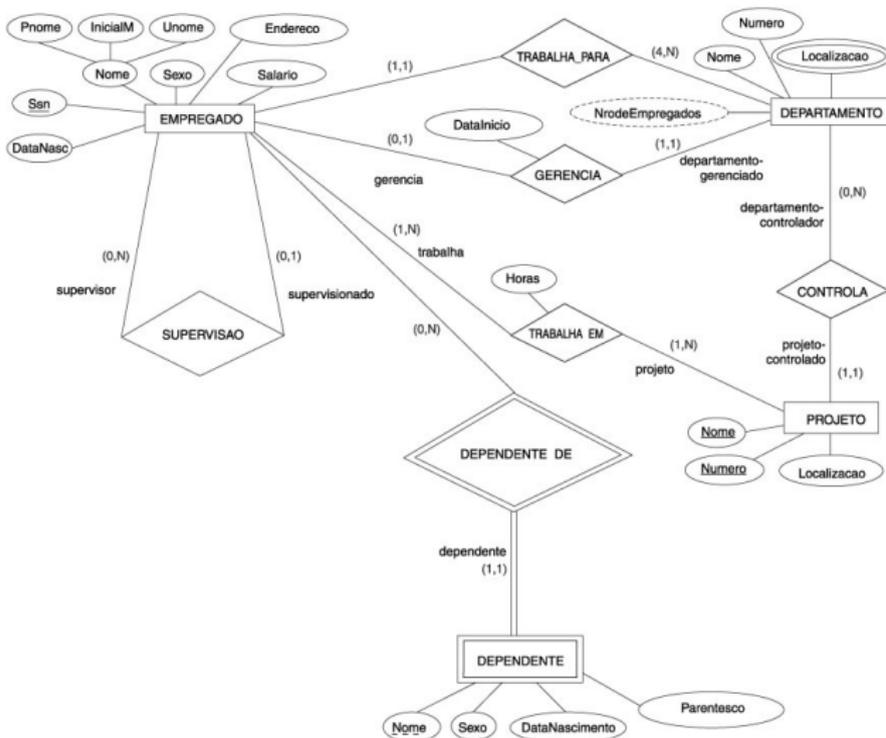
FIGURA 3.2 Um diagrama do esquema ER para o banco de dados EMPRESA.

# Notações alternativas para o modelo ER

## Notação (min,max) para a razão de cardinalidade

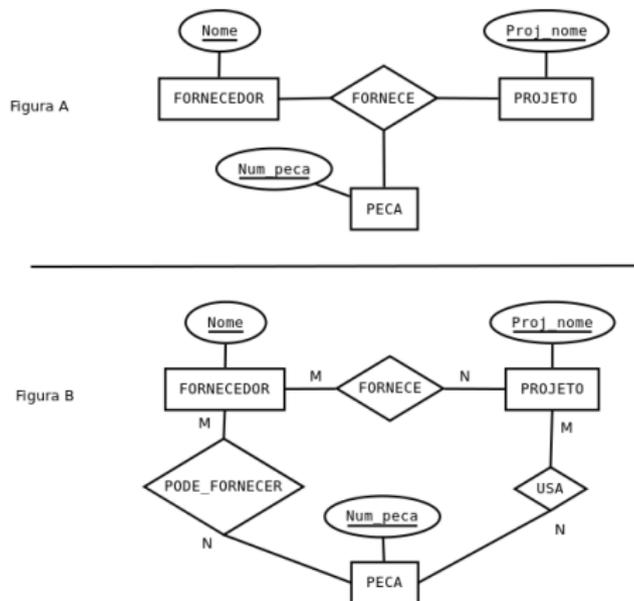
- ▶ Ideia: associar um par (min,max) a cada participação de um tipo de entidade no relacionamento
- ▶ Cada entidade do tipo de entidade deve participar de pelo menos *min* e no máximo *max* entidades do relacionamento
- ▶ *min* = 0 implica em participação parcial
- ▶ *min* > 0 implica em participação total

# Diagrama ER para o esquema EMPRESA – com notação (min,max) e o nomes de papéis



# Tipos de relacionamento de grau maior que dois

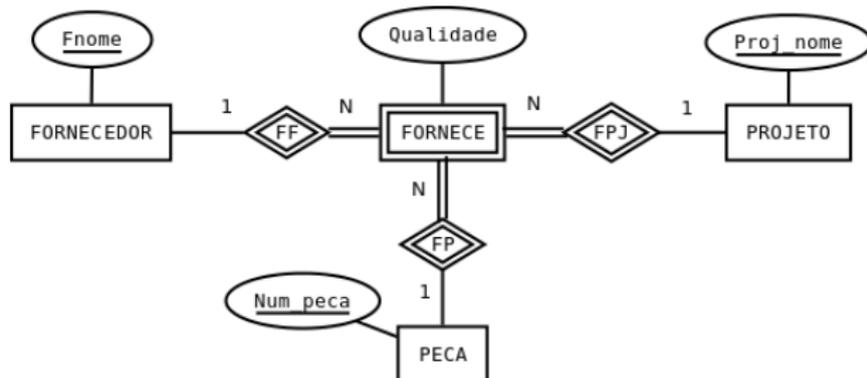
## Exemplo: tipo de relacionamento FORNECE



Os 3 relacionamentos binários (Figura B) não são equivalentes ao ternário da Figura A.

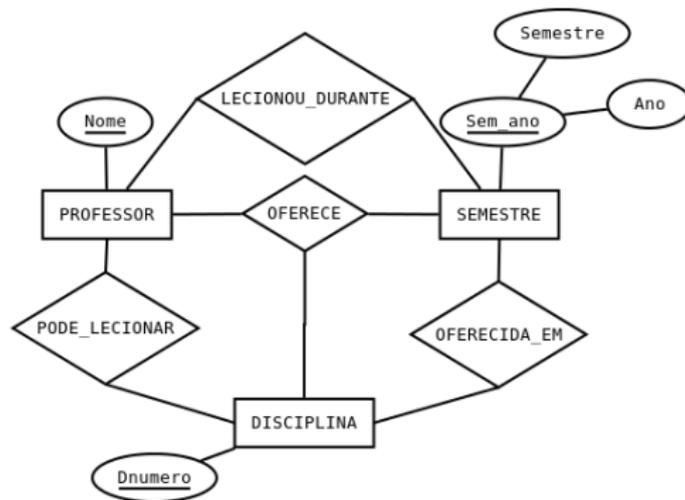
# Tipos de relacionamento de grau maior que dois

Exemplo: FORNECE como um tipo de entidade fraca



# Tipos de relacionamento de grau maior que dois

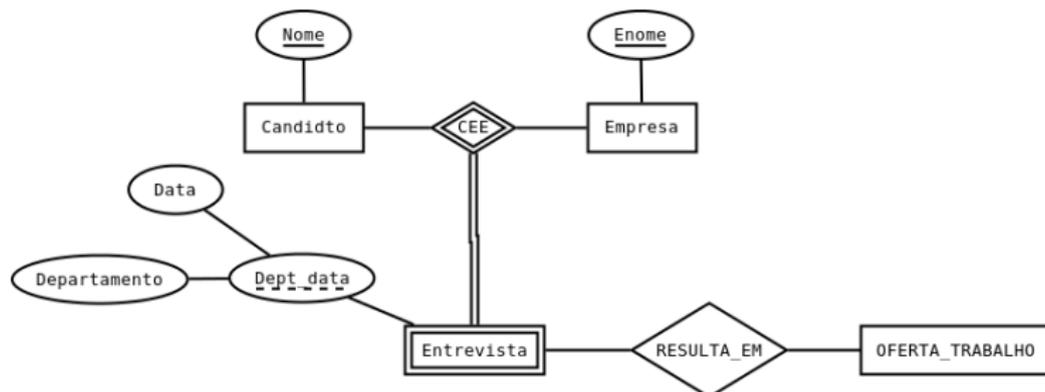
## Exemplo: tipo de relacionamento OFERECE



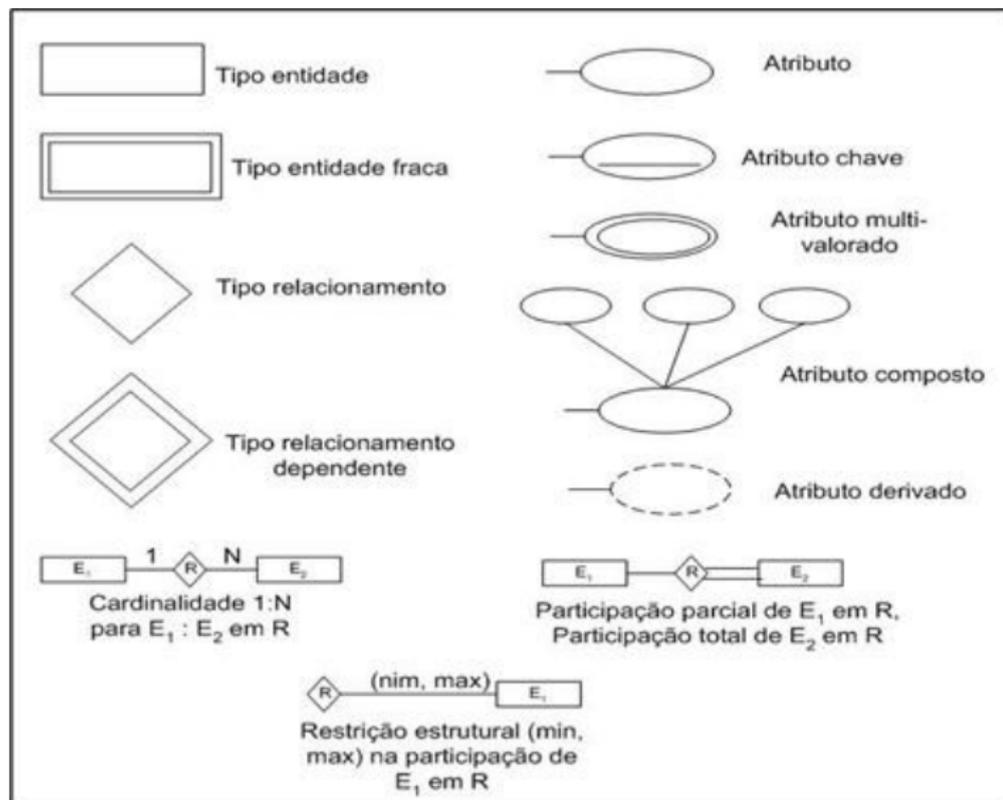
Os relacionamentos binários LECIONOU\_DURANTE e OFERECIDA\_EM são redundantes (suas instâncias podem ser deduzidas a partir das instâncias de OFERECE).

# Tipos de relacionamento de grau maior que dois

Exemplo: um tipo de relacionamento de identificação ternário



# Resumo da notação



## Referências Bibliográficas

- ▶ *Sistemas de Bancos de Dados* (6ª edição), Elmasri e Navathe. Pearson, 2010.  
Capítulo 7
- ▶ *Sistemas de Gerenciamento de Banco de Dados* (3ª edição), Ramakrishnan e Gehrke, 2008.  
Capítulo 2
- ▶ *Projeto e Modelagem de Bancos de Dados*, Teorey, Lightstone e Nadeau, 2007.  
Capítulo 2

## Cenas dos próximos capítulos...

Mais sobre projeto conceitual:

- ▶ Modelo entidade-relacionamento estendido