

# Transformação Conceitual x Lógico Relacional

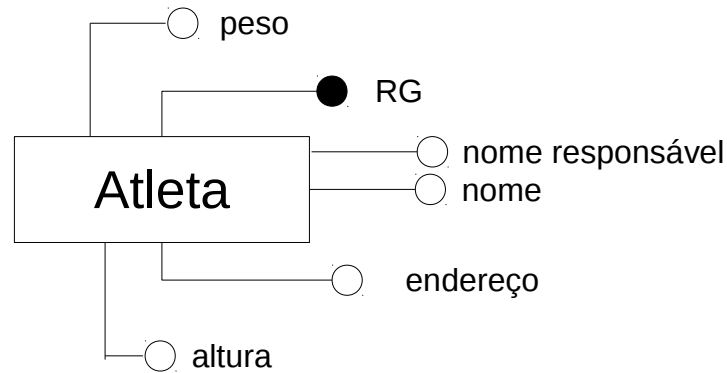
Baseado na apresentação do professor Denio Duarte

# Introdução

- Etapa onde o projetista escolhe o modelo de dados do SGBD alvo (no nosso caso relacional):
  - Evitar a criação de tabelas desnecessárias: **menos junções**
  - Evitar atributos opcionais = desperdício de espaço nas tabelas
  - Evitar controle de restrições no BD: restringir-se às chaves e a obrigatoriedade dos atributos

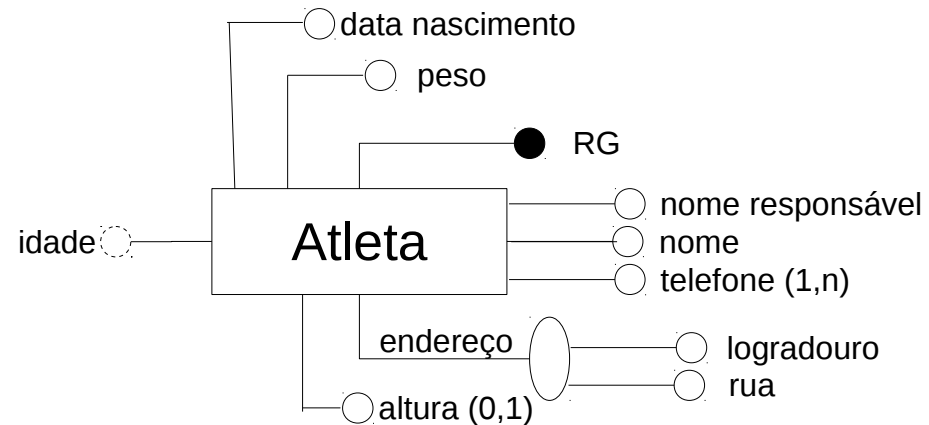
# Entidade

- Quase sempre é transformada em uma tabela



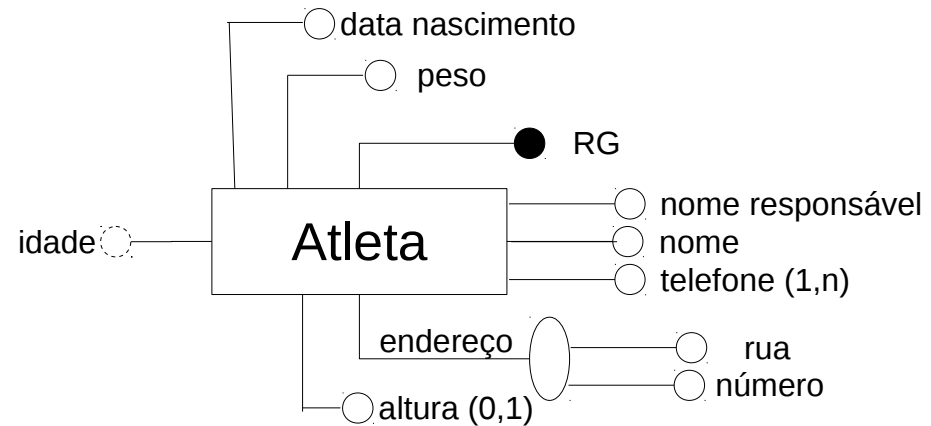
atleta (nrg, nome, peso, alt, nomresp, ender)

# Atributos



- Os atributos atômicos, opcionais e obrigatórios são criados diretamente
- Multivalorados: nova tabela ou criação de atributos para alguns valores
- Compostos: nova tabela ou desmembramento da composição

# Atributos



1: atleta (nrg, nome, peso, nresp, dtnasc, rua, nro, alt, tele1, tele2, tele3)

**ou**

2: atleta (nrg, nome, peso, alt, nresp, dtnasc, altura)

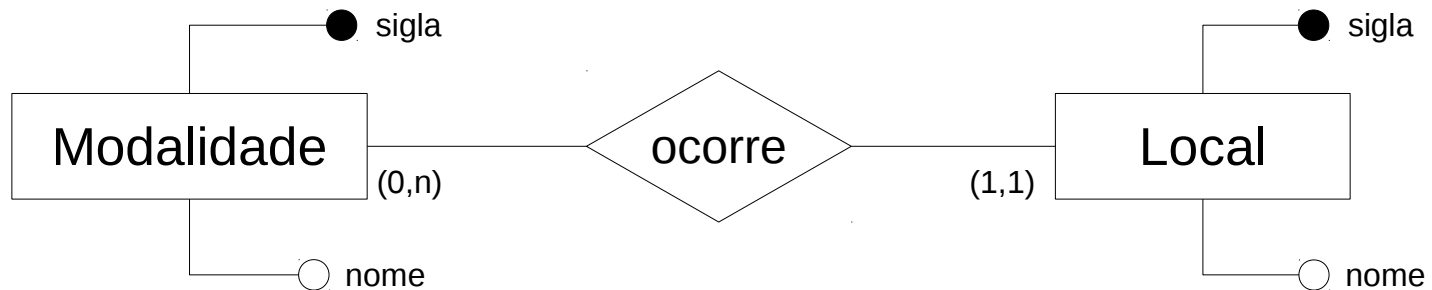
atleta\_ender(nrg(atleta), rua, nro)

atleta\_tel(nrg(atleta), nro)

# Relacionamentos

- Cardinalidade

- 1:N



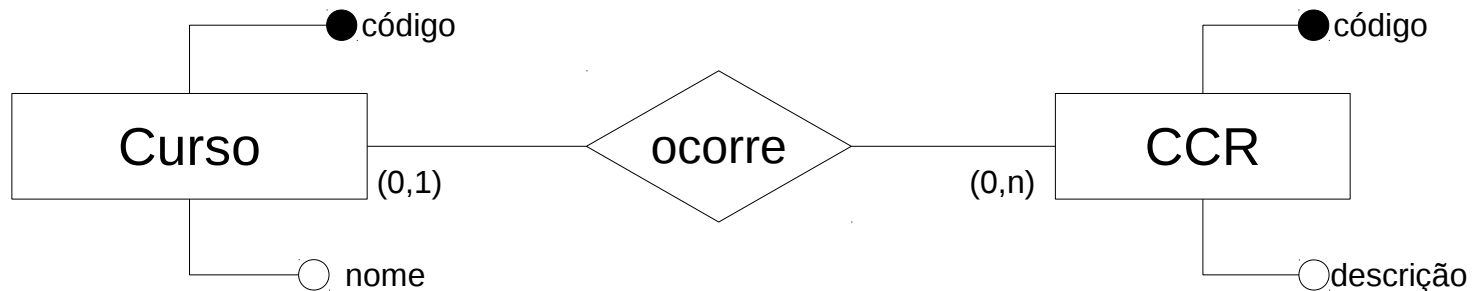
- A entidade do lado N recebe a chave primária da entidade relacionada como chave estrangeira

`local(sigla, nome)`

`modal(sigla, nome, sigl(local))`

# Relacionamentos

- Cardinalidade
  - 1:N



- A entidade do lado N recebe a chave primária da entidade relacionada como chave estrangeira opcional

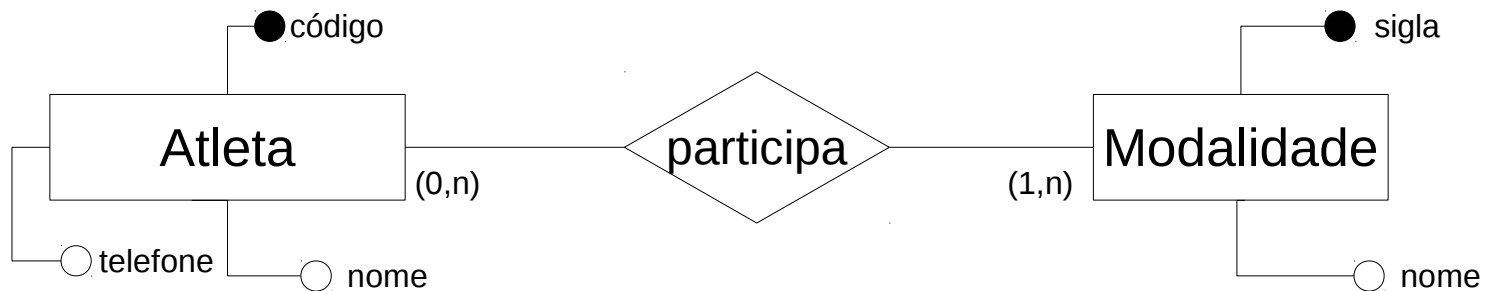
`curso(cod, nome)`

`ccr(cod, descr, codc(curso))`

# Relacionamentos

- Cardinalidade

- N:N



- O relacionamento vira uma tabela com as chaves das entidades envolvidas. Esses atributos viram chave primária.

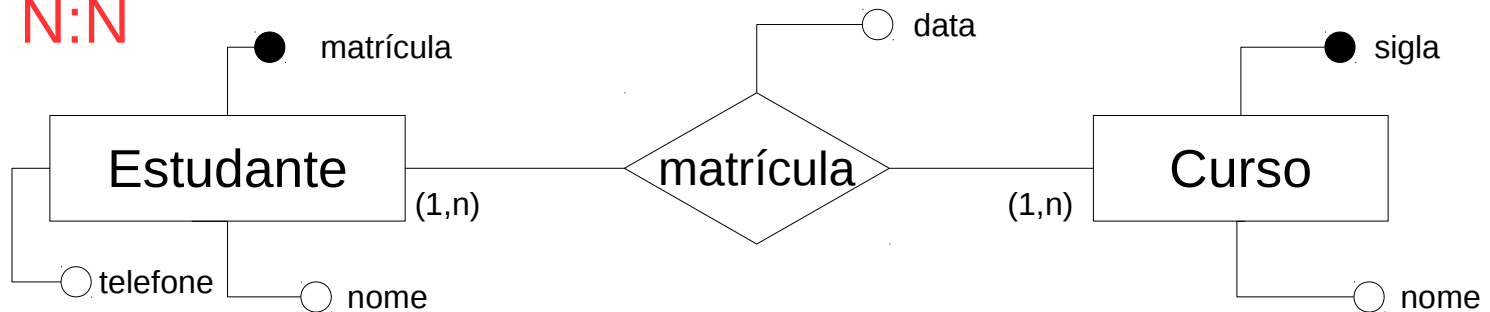
```
atleta (cod, nom, tele)    modal(sigla, nome)
      atle_modal (coda(atleta), sigm(modal))
```



# Relacionamentos

- Relacionamento
  - Cardinalidade

- **N:N**



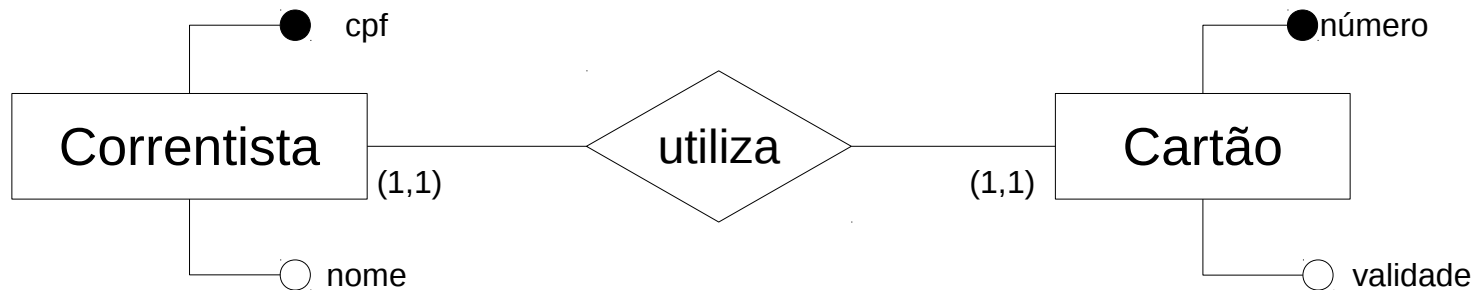
- O relacionamento vira uma tabela com as chaves das entidades envolvidas (independente da obrigatoriedade)

estudante (mat, nom, tele)    curso(sigla, nome)  
matricula (coda(atleta), sigm(modal), data)

# Relacionamentos

- Cardinalidade

- 1:1



- Fusiona: `corren(cpf, nome, nrocart, valcart)`

**OU**

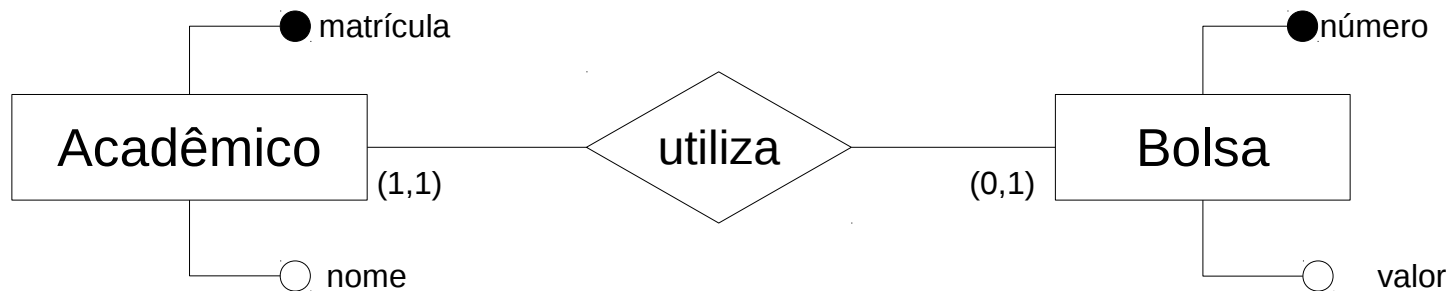
- Cria: `corren(cpf, nome)`

`cartao(cpf(corren), nrocart, valcart)`

# Relacionamentos

- Cardinalidade

- 1:1



- Fusiona: `aluno(mat, nome, nbolsa, valbol)`

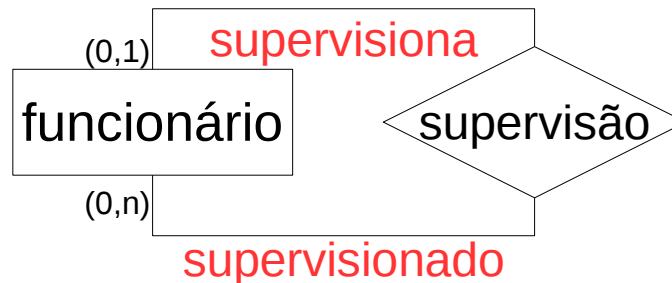
**OU**

- Cria: `aluno(mat, nome)`

`bolsa(mat(aluno), nbolsa, valbol)`

# Auto-relacionamentos

- **1:N**: a entidade recebe a chave dela mesma como estrangeira
  - `func(mat, nome, dtadm, matsup(func))`

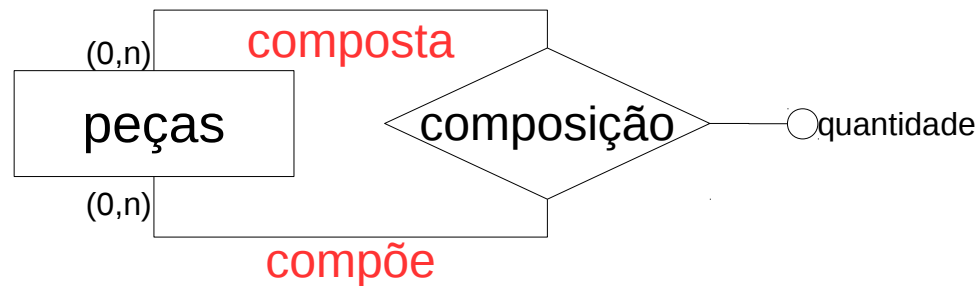


# Auto-relacionamentos

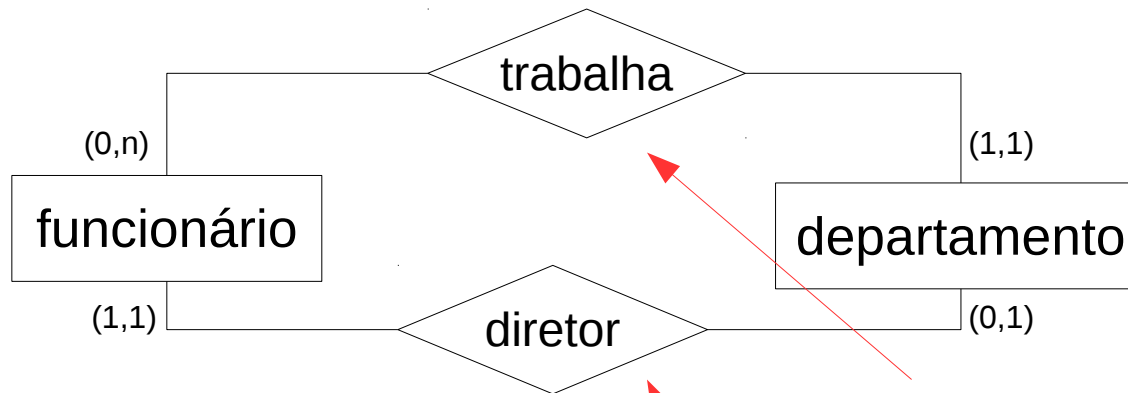
- **N:N**: cria-se uma entidade para resolver o relacionamento

peca(cod, nome, prc, qtestoq)

compoe(codp(peca), codparte(peca), qtdade)



# Relacionamentos múltiplos

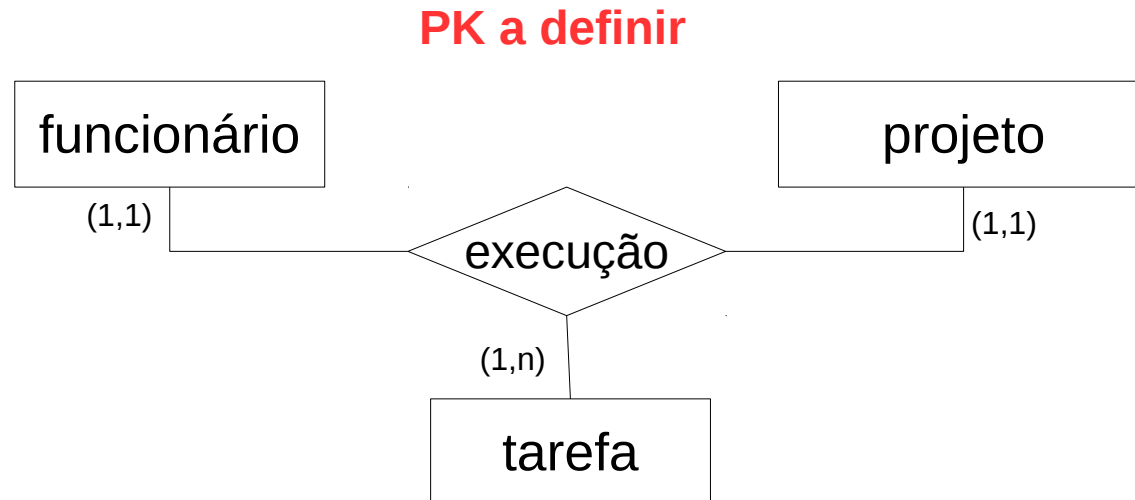


func(mat, nome, dtadm, codd(depto))

depto(cod, nome, matd(func))

# Relacionamento Ternário

- Geralmente, transforma a relação em tabela como em relacionamentos **N:N**
  - A PK é definida conforme a necessidade de cardinalidade  
`execucao(matf(func), codp(prjt), codt(tarefa))`

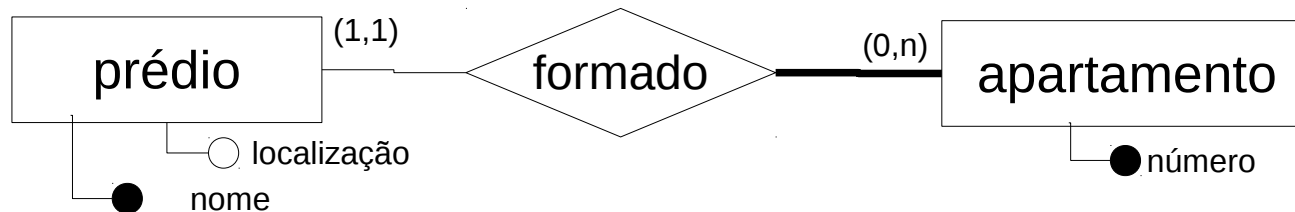


# Entidade Fraca

- Relacionamento identificador (entidade fraca)
  - Mesma estratégia de outros casos. Porém, a chave estrangeira se torna primária também

predio (nome, local)

apto (nome(predio), nro)

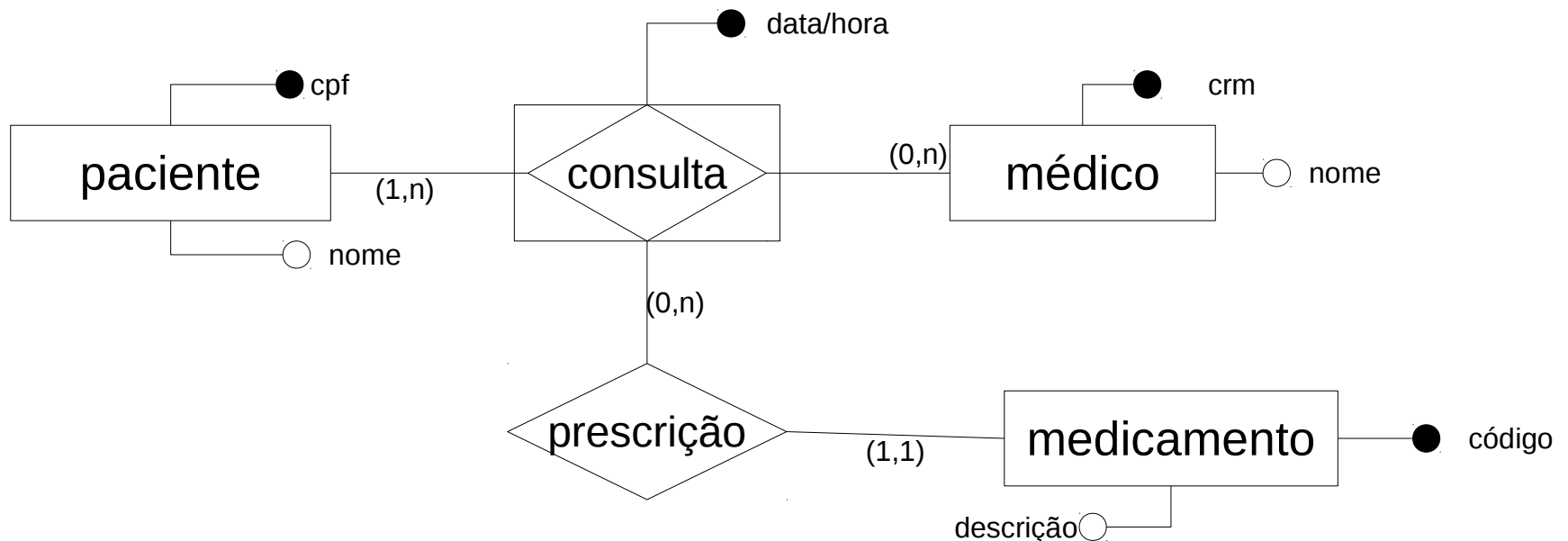




# Entidade associativa

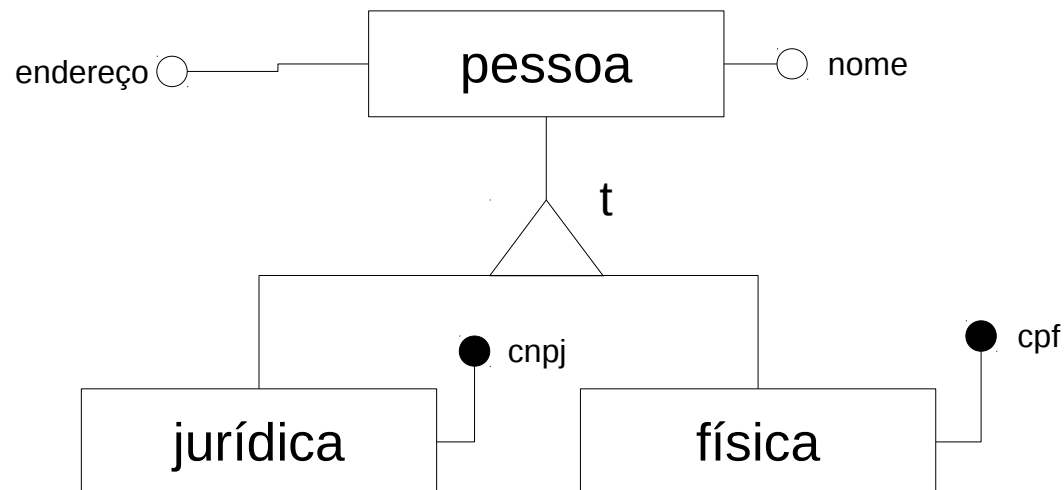
- Caso similar ao do relacionamento ternário: relacionamento se torna uma tabela

`consulta(crm(medico), cpf(paciente), dthora, codm(medic))`



# Especialização

- Caso total
  - Cria-se tabelas apenas para as especializações  
`pesj(cnpj, nome, ender)` `pesf(cpf, nome, ender)`
  - Cria-se apenas uma tabela  
`pessoa(cod, nome, ender, cpf, cnpj, tipo)`



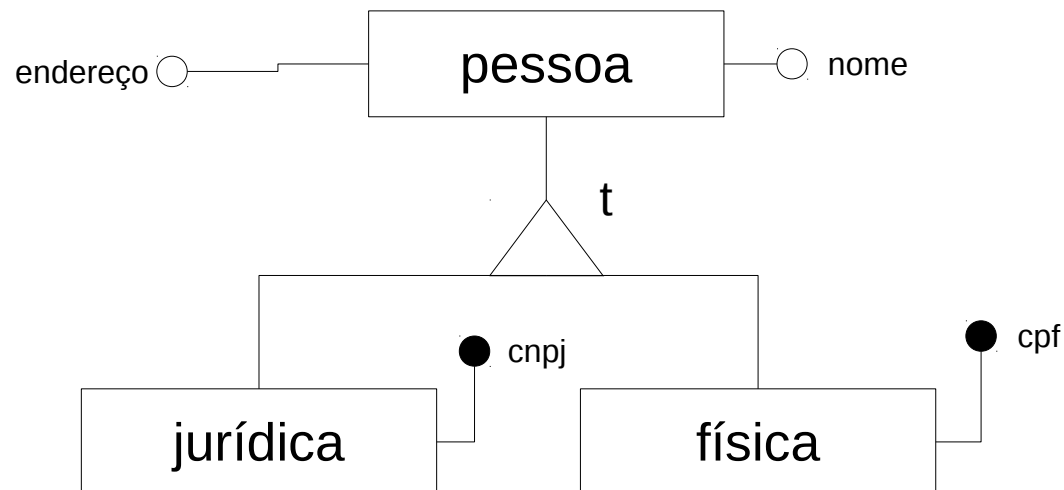
# Especialização

- Caso total

- Cria-se todas as tabelas

pesj(cod(pessoa), cnpj)   pesf(cod(pessoa), cpf)

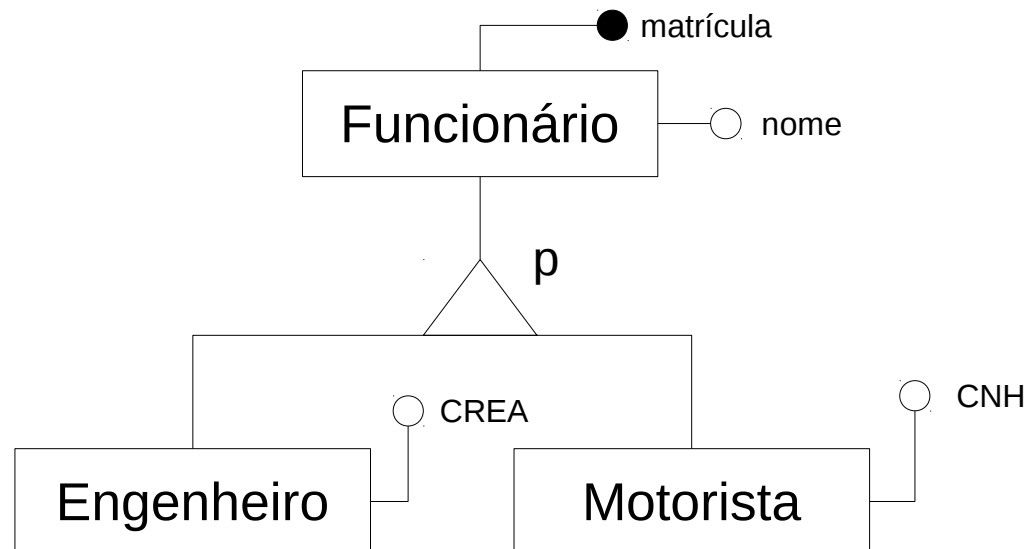
pessoa(cod, nome, ender)



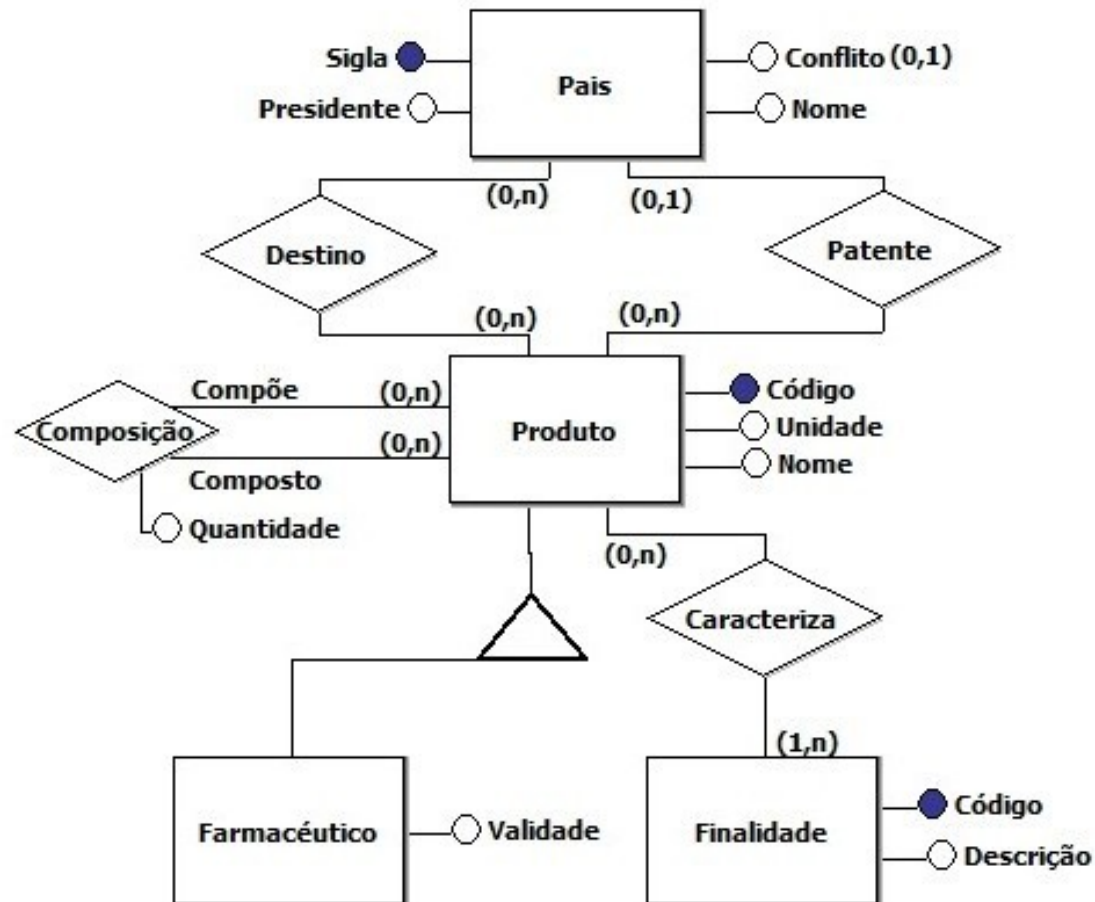
# Especialização

- Caso Parcial

- Casos similares ao total exceto que deve ser criada a tabela *mãe* sempre



# Exemplo



# Exemplo

