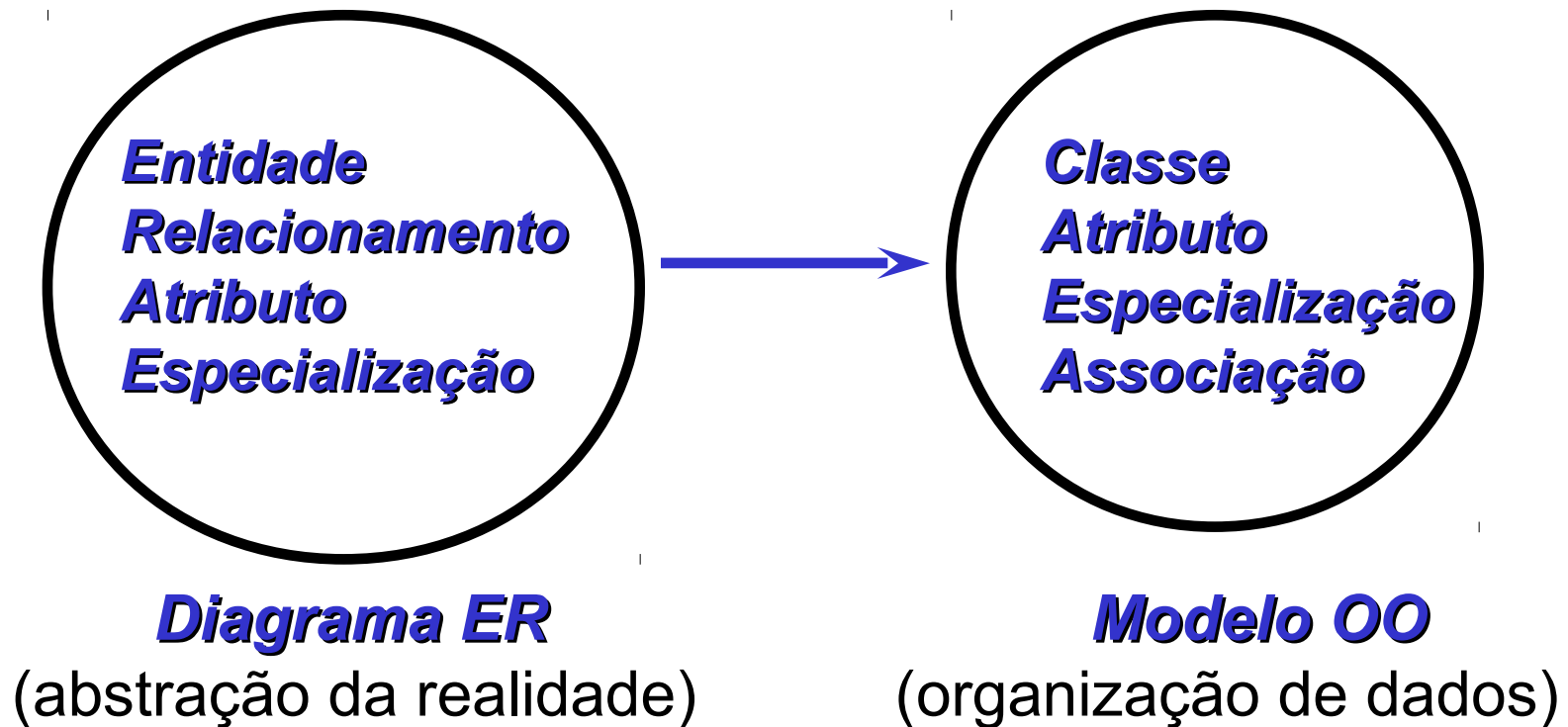
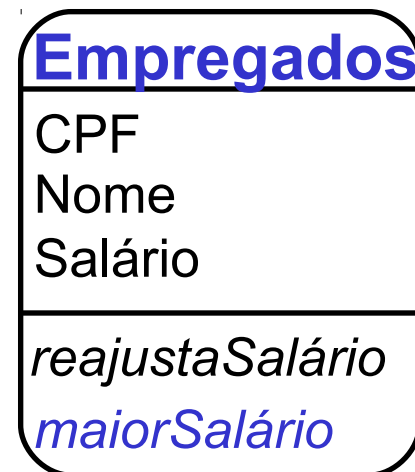


Projeto Lógico de BDOO



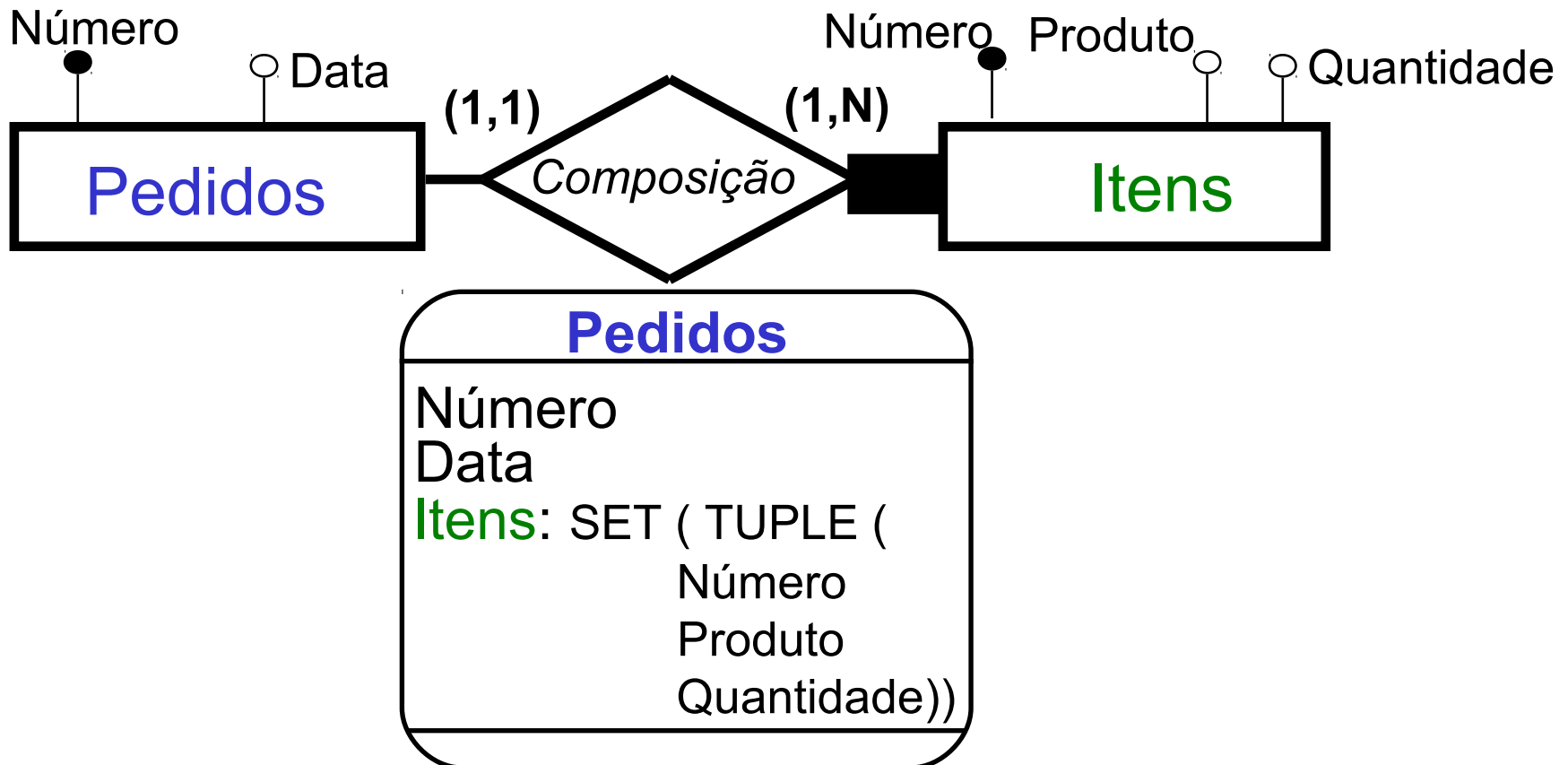
Mapeamento de Entidades

- Entidades tornam-se classes
 - controle de unicidade de atributos identificadores (*CPF*, p.ex.) deve ser definido
- Métodos relevantes em nível de instâncias e da classe podem ser previstos



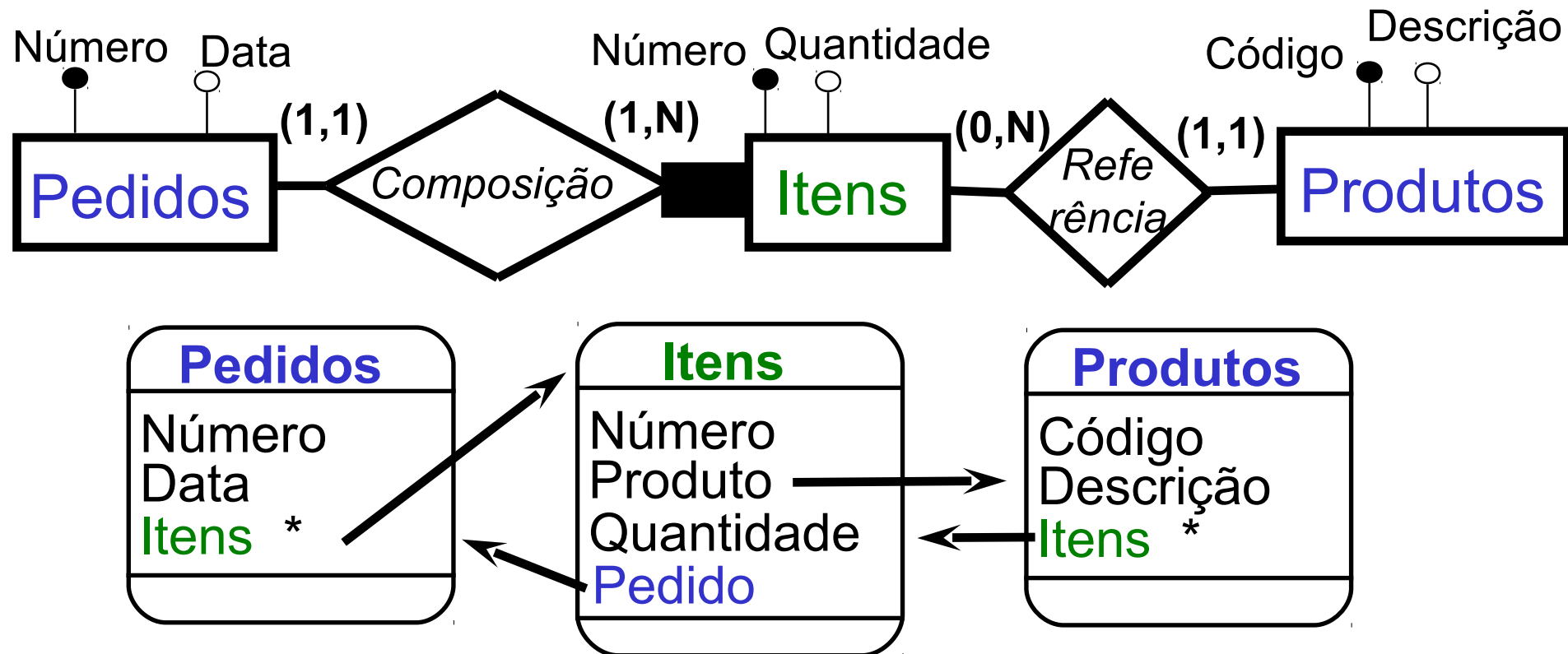
Mapeamento de Entidades Fracas

- Opção 1: atributo composto e multivalorado
 - entidade fraca relaciona-se apenas com a entidade forte



Mapeamento de Entidades Fracas

- Opção 2: classe
 - entidade fraca relaciona-se também com outras entidades que desejam referenciá-la

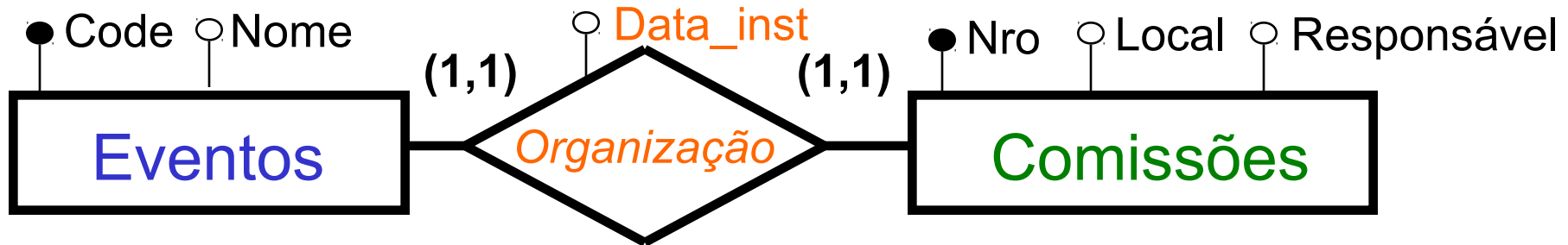


Relacionamentos

- Análise de 3 casos
 - 1:1
 - 1:N
 - M:N
- Participação *obrigatória/opcional* da entidade no relacionamento
 - se o SGBDOO não dá suporte explícito a estas RIs na ODL, então
 - definir métodos de RI

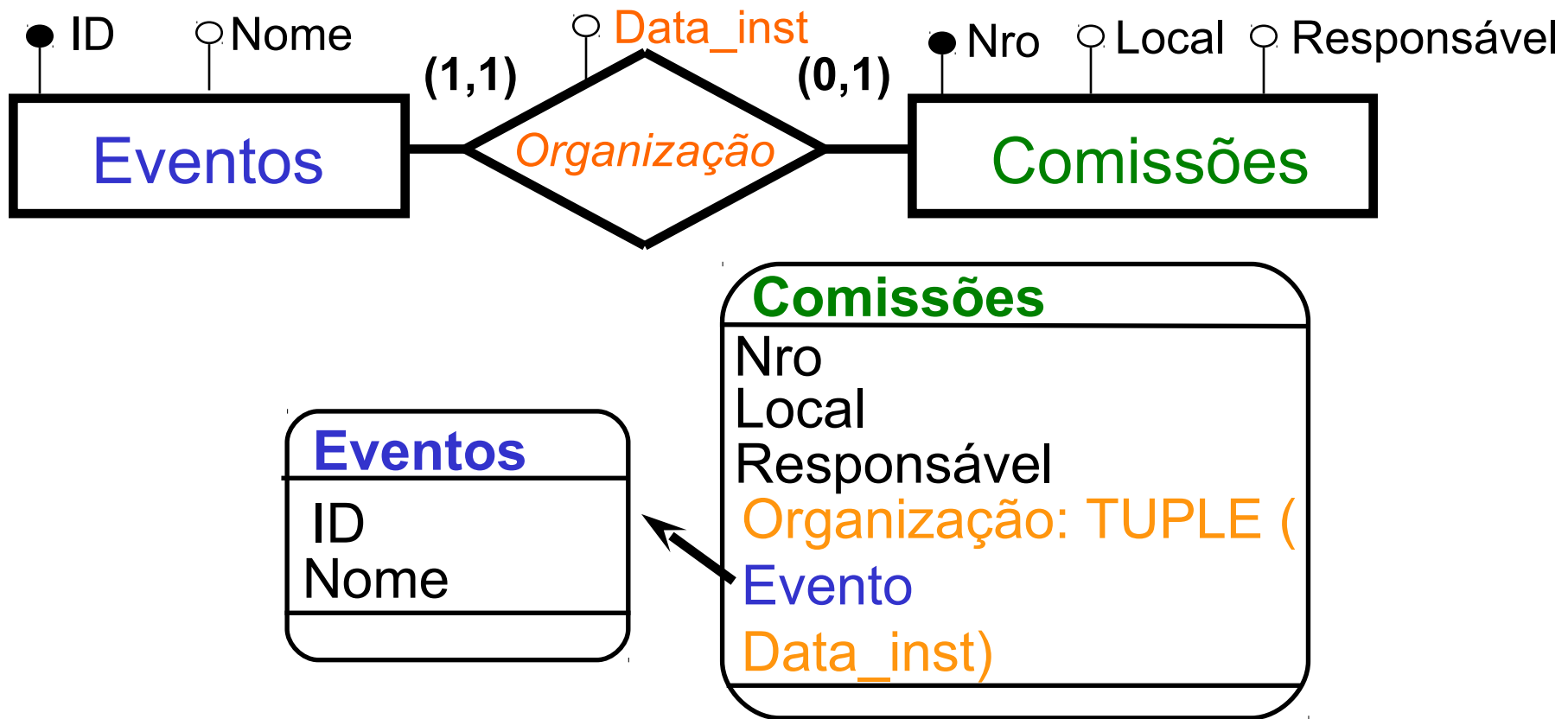
Relacionamentos 1:1

- Obrigatório em ambos os sentidos
 - fusão de entidades



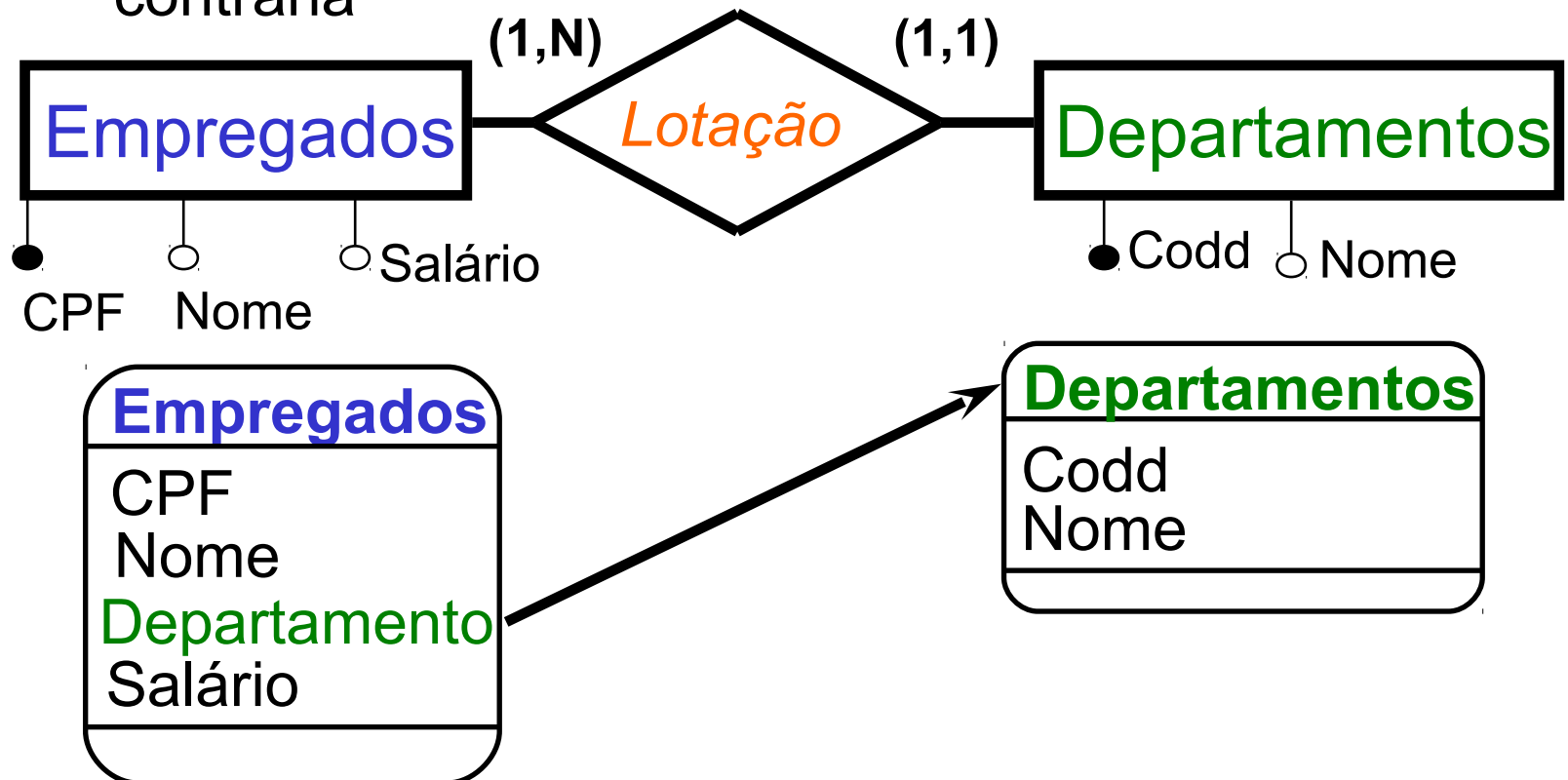
Relacionamentos 1:1

- Opcional em um ou em ambos os sentidos
 - atributo de referência
 - pelo menos na classe com obrigatoriedade de participação, se apenas um sentido é opcional



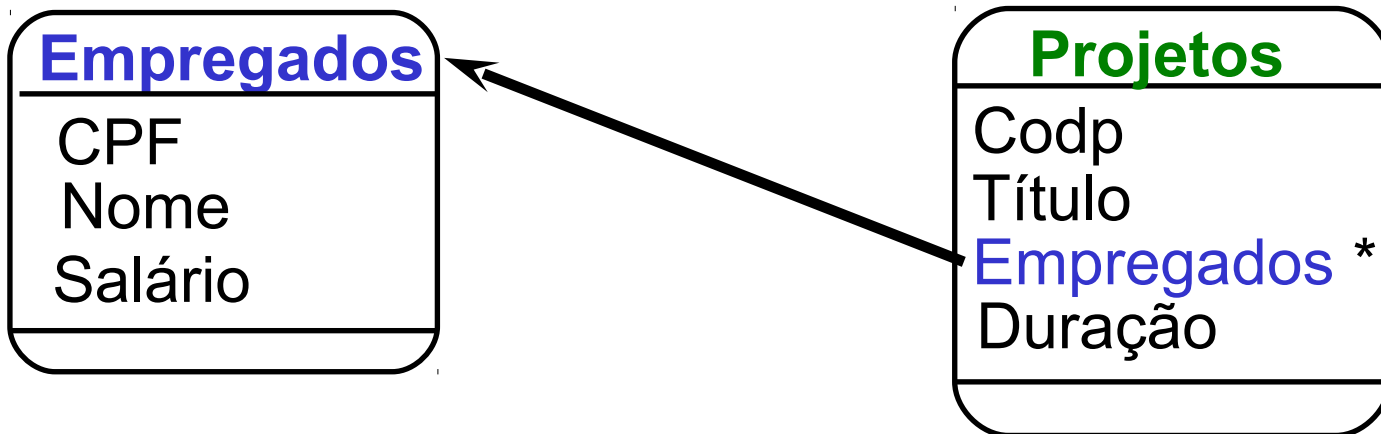
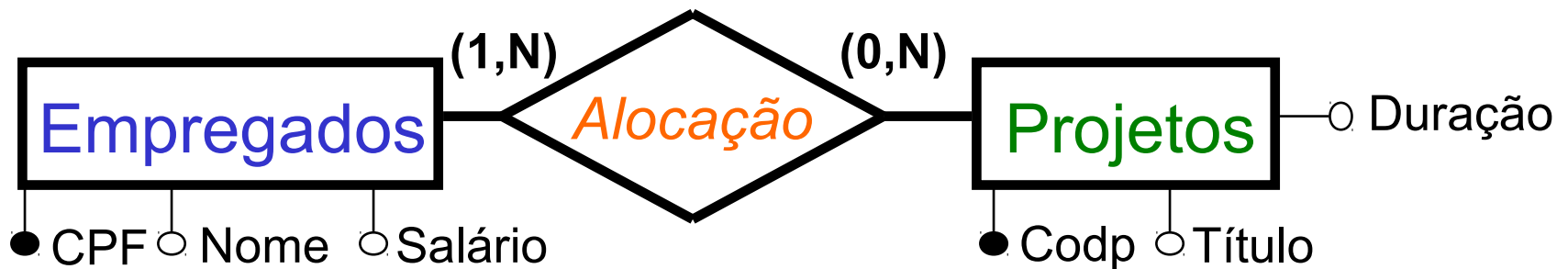
Relacionamentos 1:N

- **Atributo de referência**
 - pelo menos na classe com referência monovalorada (define uma estrutura menos complexa)
 - exceto se a maior frequência de pesquisa for a contrária



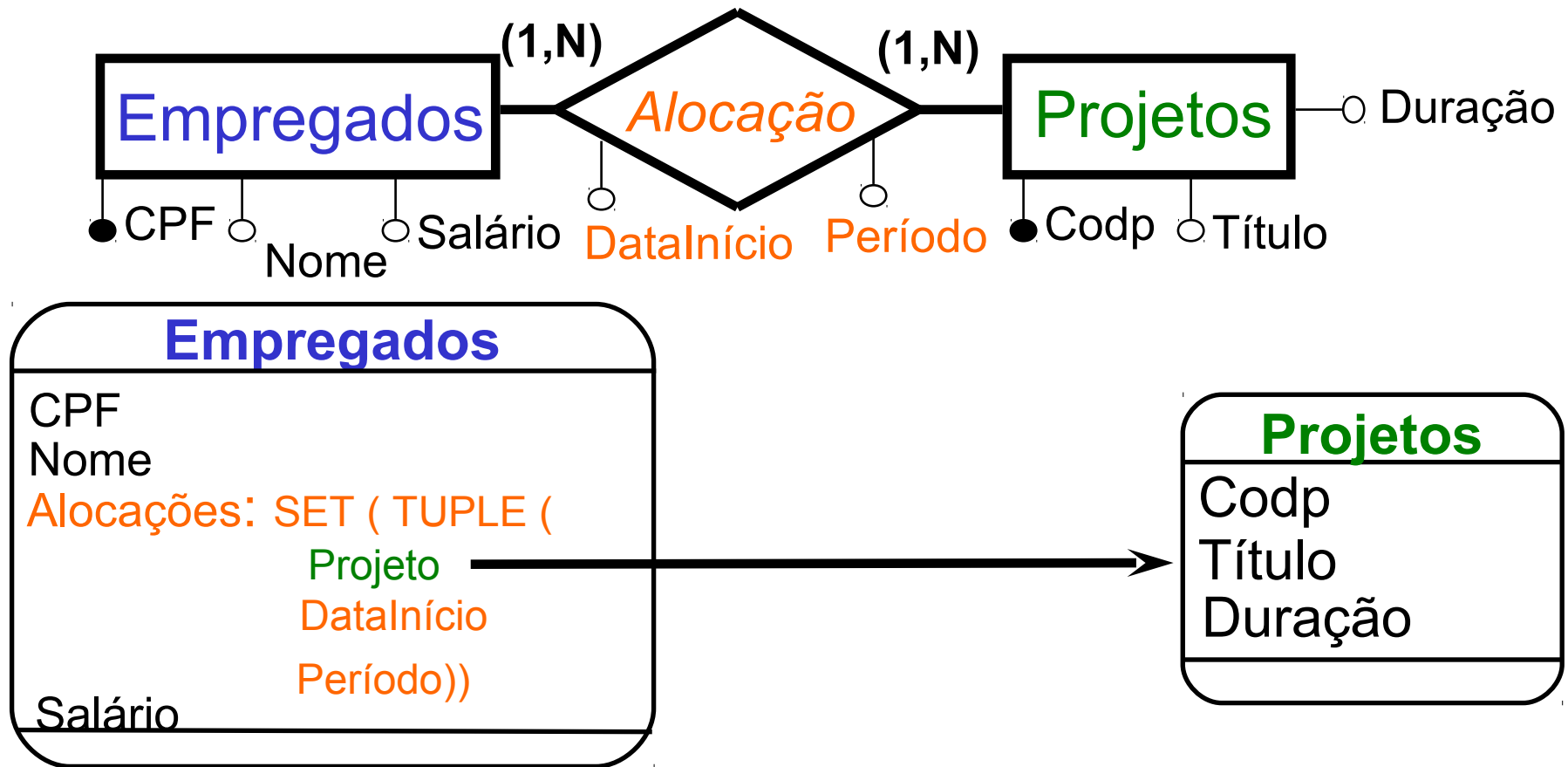
Relacionamentos M:N

- **Atributo de referência multivalorado**
 - em pelo menos uma das classes
(preferência para o relacionamento obrigatório)



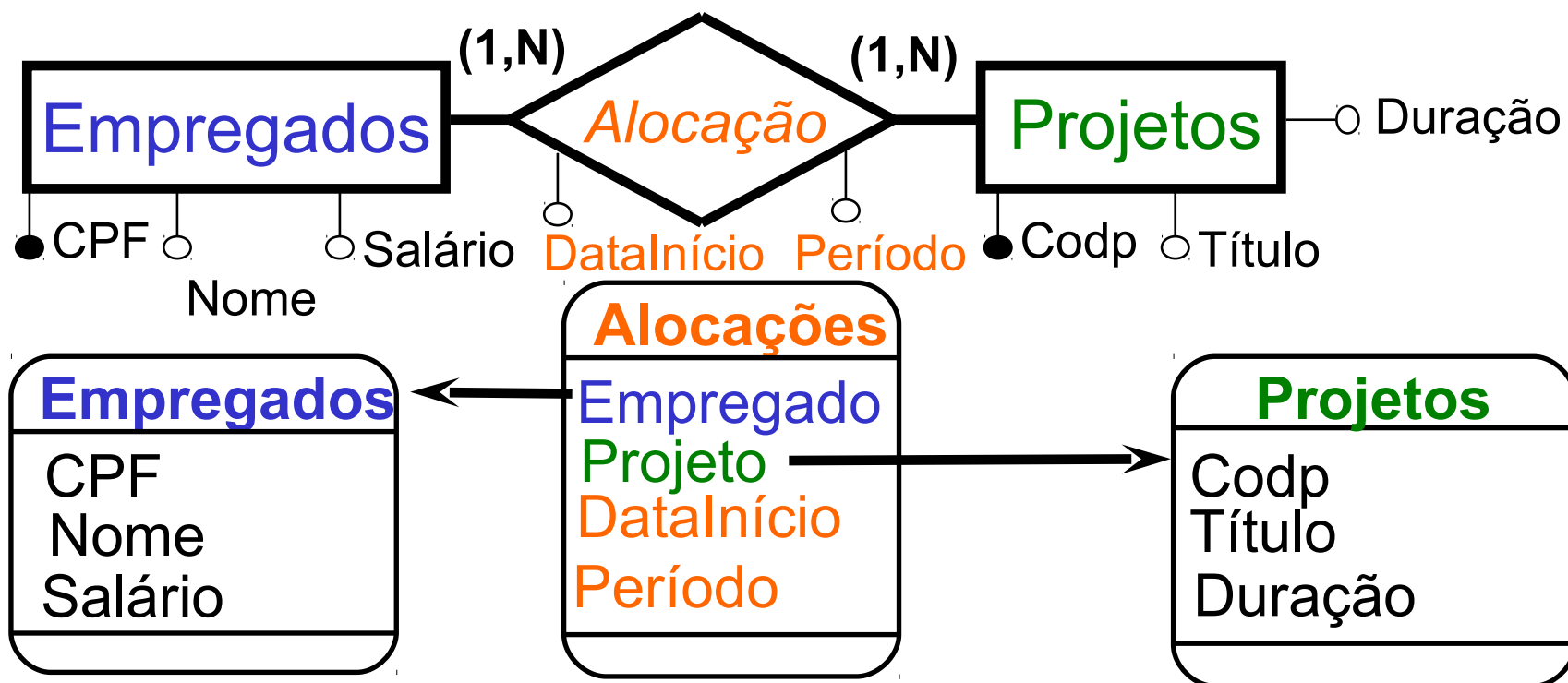
Relacionamentos M:N

- Existem atributos no relacionamento?
 - alternativa 1: atributo complexo em alguma classes
 - menos classes; certas consultas são prejudicadas



Relacionamentos M:N

- Existem atributos no relacionamento?
 - alternativa 2: classe para o relacionamento
 - acesso direto a instâncias de **Alocações**; evita estruturas complexas nas classes; mais classes
 - alternativa tb válida qdo há opcionalidade em ambos os lados do relacionamento (mesmo para casos 1-1 e 1-N)
 - evita atributos complexos opcionais em uma/ambas as classes



Atributos Especiais

- Atributo Opcional

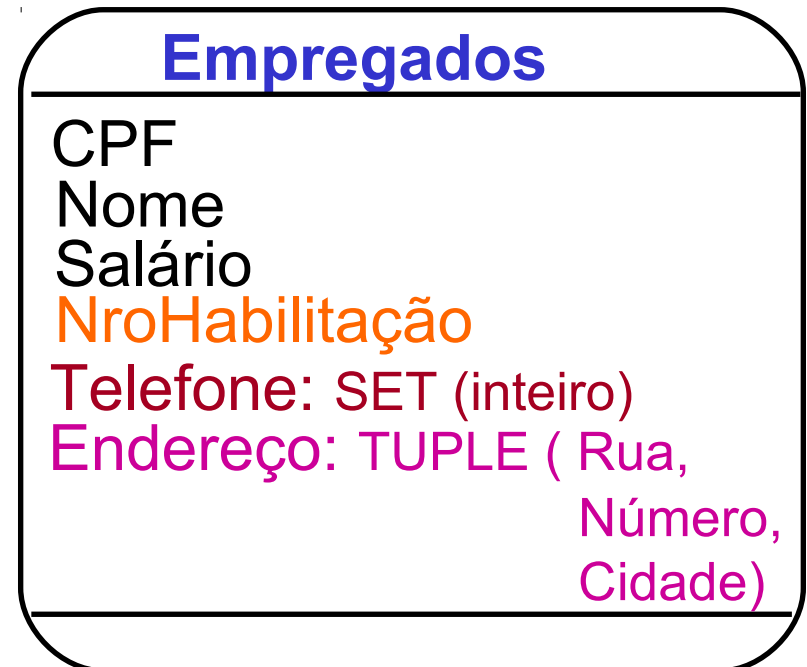
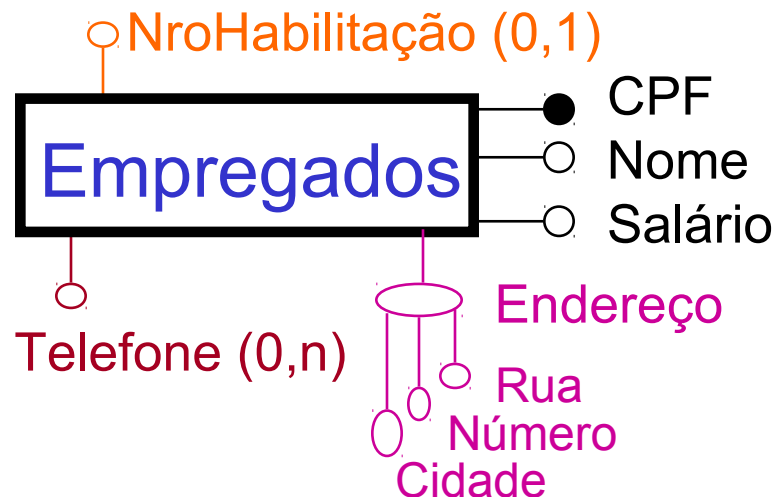
- atributo que pode assumir *null* OU atributo obrigatório em uma subclasse da entidade

- Atributo Composto

- atributo com domínio *tuple*

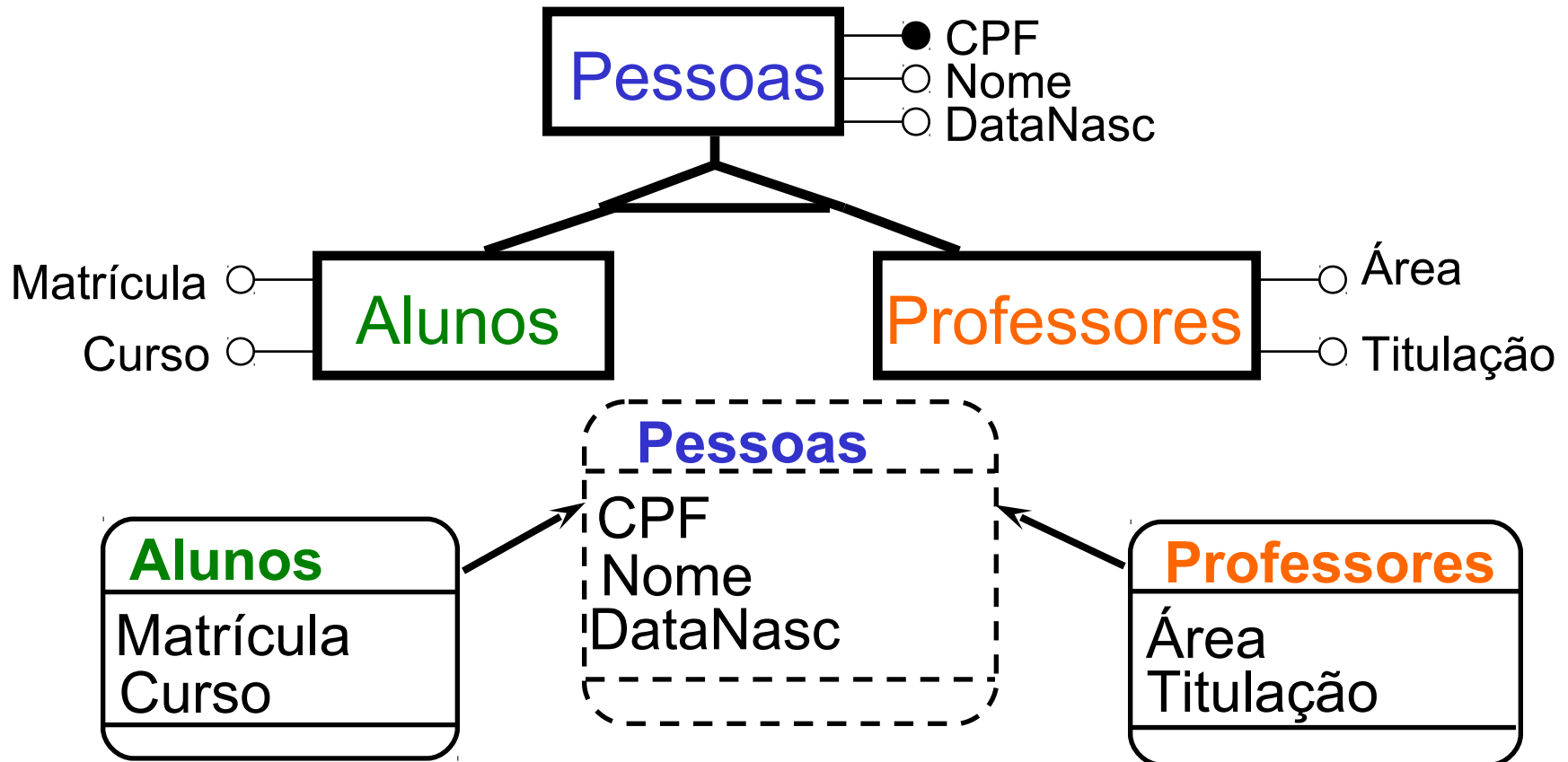
- Atributo Multivalorado

- atributo com domínio *set* ou *list*



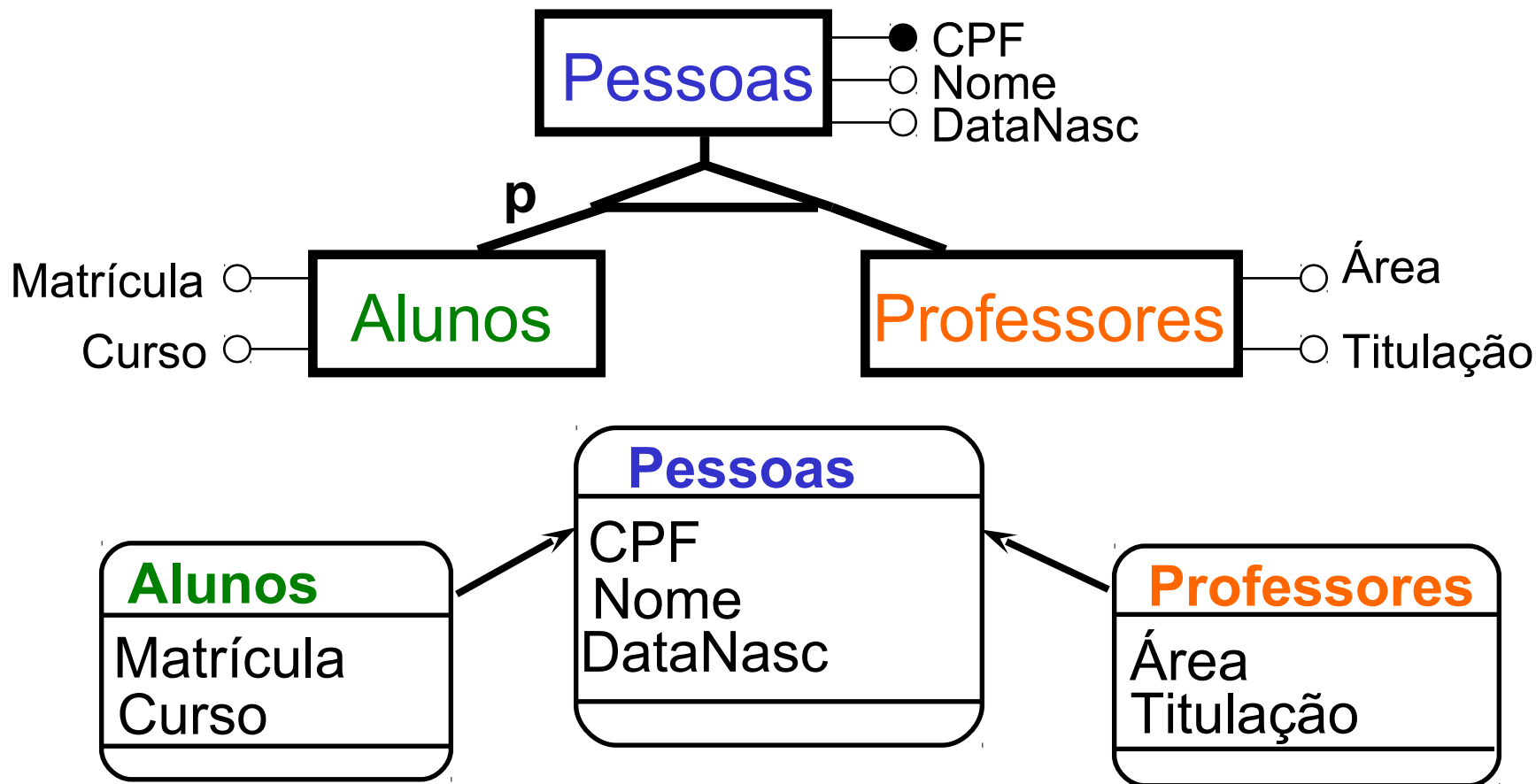
Herança

- Com exclusão mútua e totalidade
 - gera hierarquia de classes
 - RI: instâncias apenas nas subclasses
 - classe genérica é metaclassa



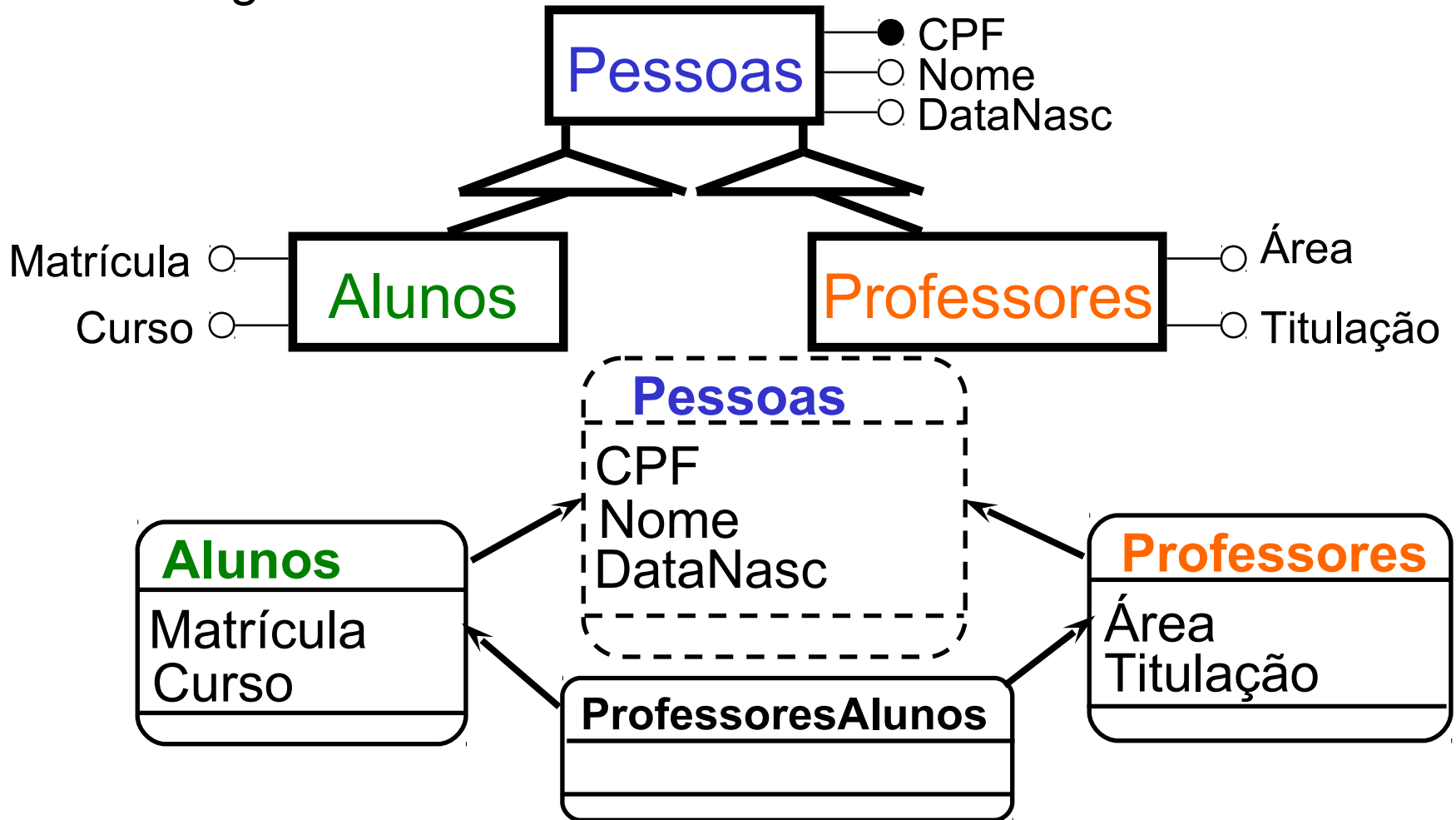
Herança

- Com exclusão mútua e parcialidade
 - idêntico ao caso anterior
 - instâncias podem existir na classe genérica



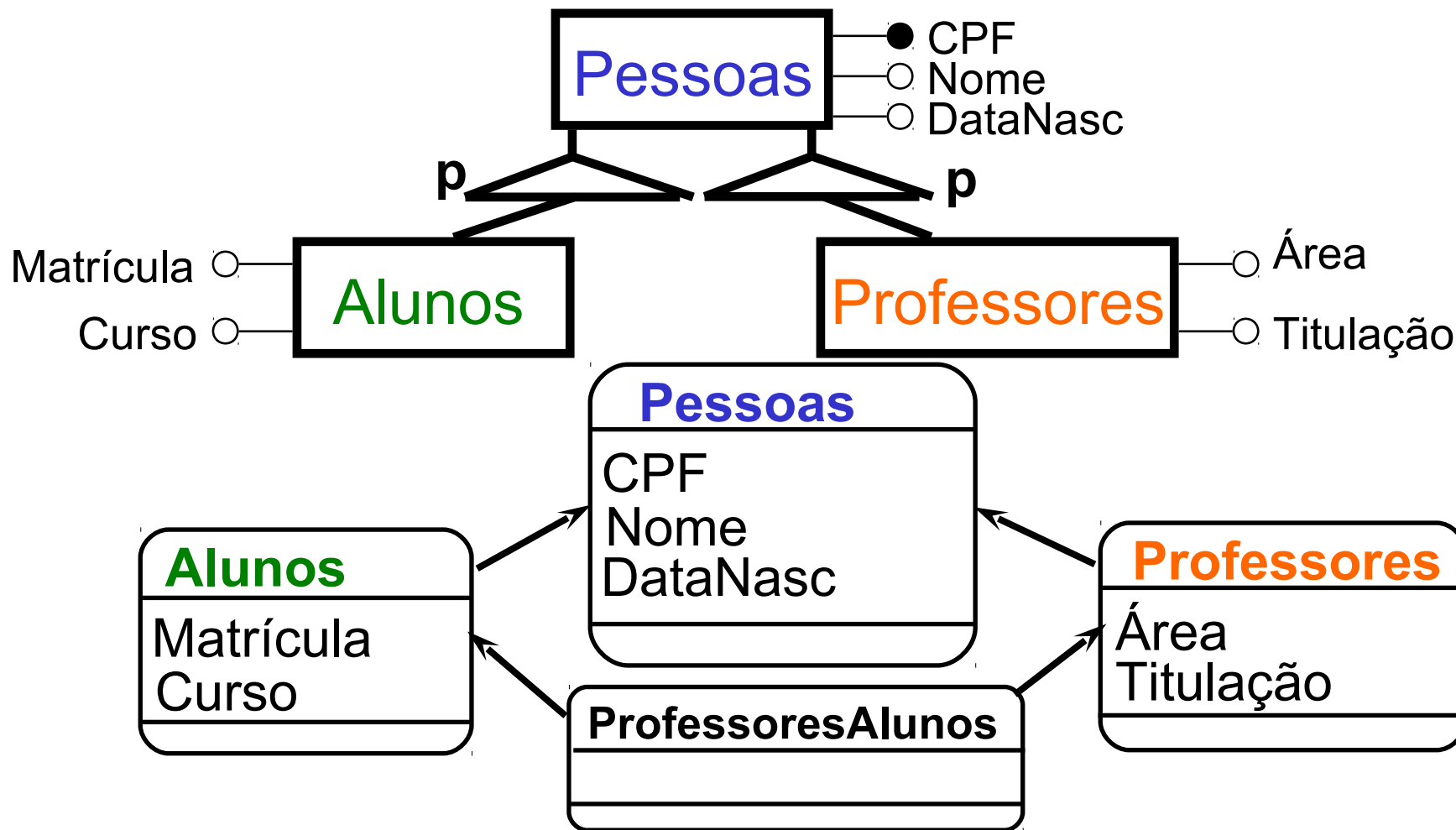
Herança

- Sem exclusão mútua e totalidade
 - mapeamento complexo...
 - prever subclasses para todos os papéis possíveis
 - classe genérica é metaclassa



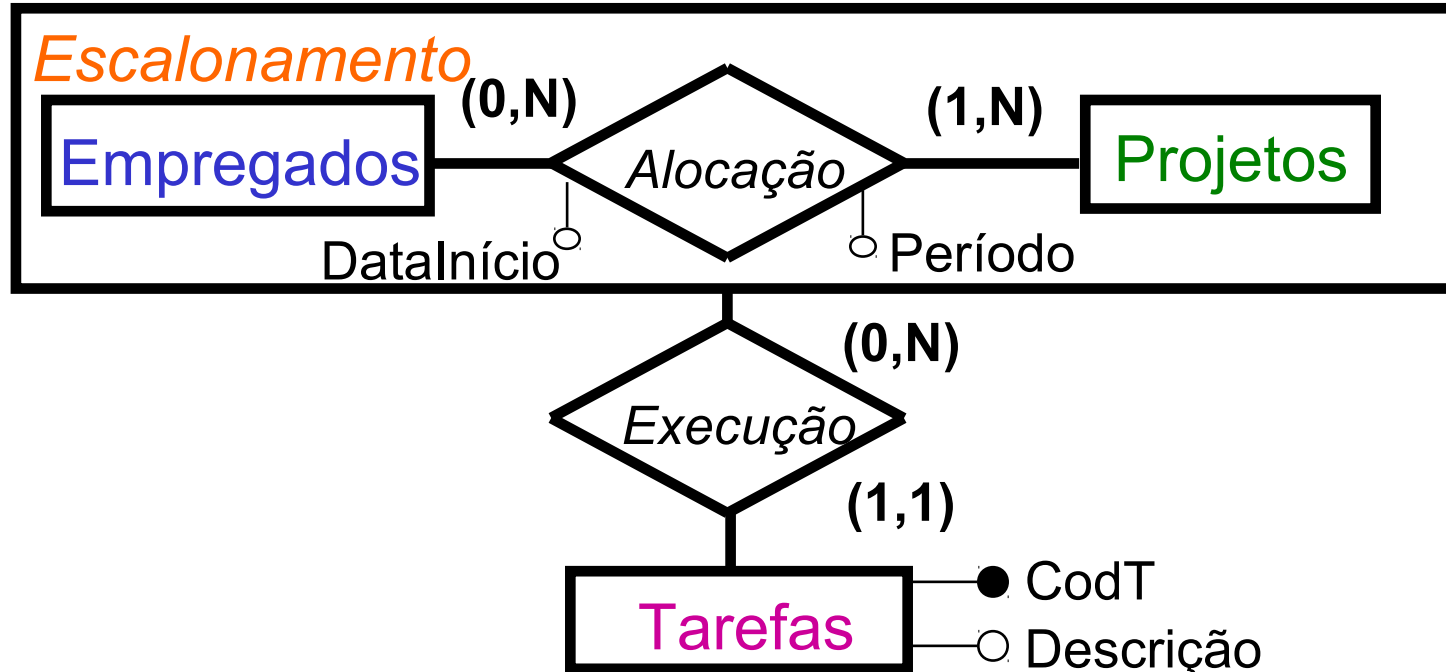
Herança

- Sem exclusão mútua e parcialidade
 - idêntico ao caso anterior
 - instâncias podem existir na classe genérica



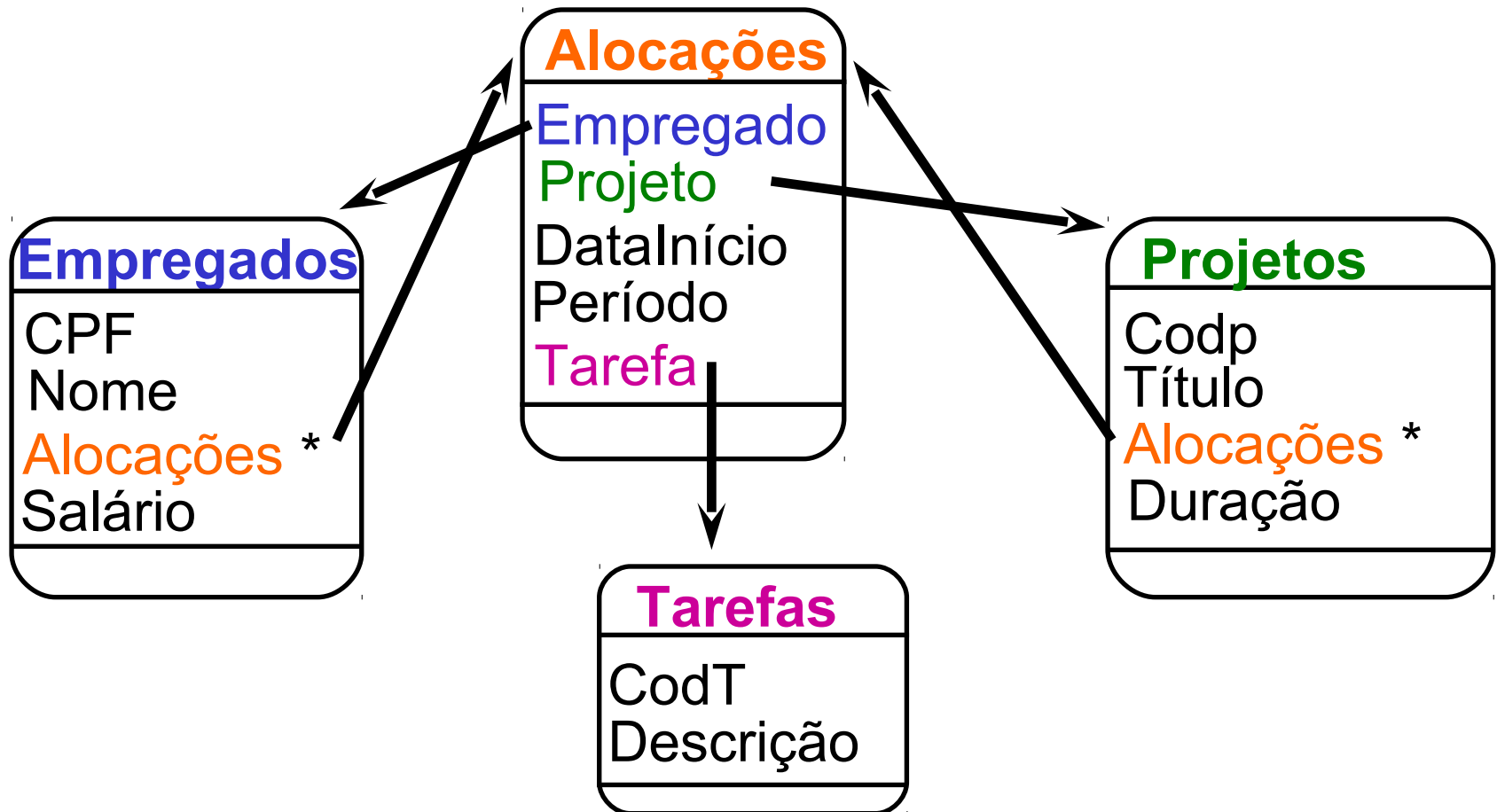
Entidade Associativa

- Mesmas diretrizes para mapeamento de relacionamentos binários
- Exemplo
 - entidade associativa *Escalonamento*



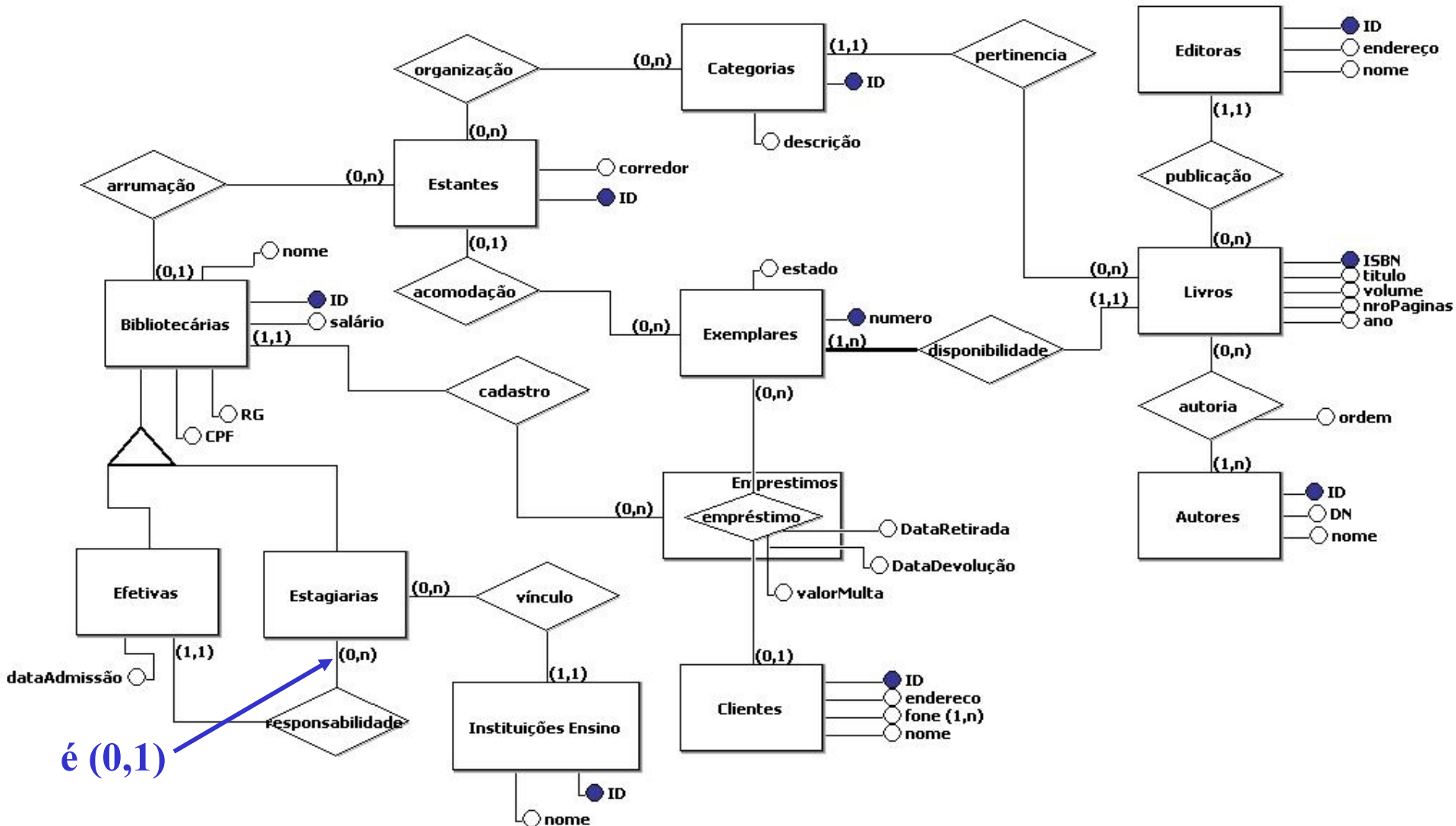
Entidade Associativa

- Possível resultado para o mapeamento



Exercício 1

- Apresente uma modelagem lógica OO para a modelagem ER abaixo (domínio de uma Biblioteca)



Exercício 2

- Apresente mapeamentos válidos para relacionamentos ternários do ER
 - considerar 4 casos (todos com cardinalidades obrigatórias)
 - M:N:P
 - 1:M:N
 - 1:1:N
 - 1:1:1
 - é possível que haja mais de uma alternativa de mapeamento? Se sim, apresente-as e descreva as vantagens de cada uma delas