

INSTITUTO FEDERAL

Mato Grosso do Sul

Mapeamento de Objetos para o Modelo Relacional

Prof. Luiz F. Picolo
luiz.picolo@ifms.edu.br

Análise e Projeto Orientado a Objetos 2
Técnico em Informática para Internet

Roteiro

- Introdução
- Projeto de banco de dados
- Mapeamento de objetos para o modelo relacional

Introdução

Introdução

Relevância do mapeamento de objetos para o modelo relacional:

- A tecnologia OO como forma usual de desenvolver sistemas de software.
- Sem dúvida os SGBDR dominam o mercado comercial.

Os princípios básicos do paradigma da orientação a objetos e do modelo relacional são bastante diferentes. No modelo de objetos, os elementos (objetos) correspondem a abstrações de comportamento.

No modelo relacional, os elementos correspondem a dados no formato tabular.

Introdução

Os objetos de um sistema podem ser classificados em:

- **Objetos transientes:** existem somente na memória principal.
 - Objetos de controle e objetos de fronteira.
- **Objetos persistentes:** têm uma existência que perdura durante várias execuções do sistema.
 - Precisam ser **armazenados** quando uma execução termina, e **restaurados** quando uma outra execução é iniciada.
 - Tipicamente objetos de entidade.

Introdução

Para objetos persistentes, surge o problema de conciliar as informações representadas pelo estado de um objeto e pelos dados armazenados em registros de uma tabela.

O **descasamento de informações** (Impedância) é um termo utilizado para denotar o problema das diferenças entre as representações do modelo OO e do modelo relacional.

Uma proporção significativa do esforço de desenvolvimento recai sobre a solução que o desenvolvedor deve dar a este problema.

Projeto de Banco de Dados

Projeto de Banco de Dados

Uma das primeiras atividades do **projeto detalhado** de um SSOO é o desenvolvimento do banco de dados a ser utilizado, se este não existir. Essa atividade corresponde ao projeto do banco de dados.

As principais tarefas no projeto do banco de dados são:

- Construção do esquema do banco de dados;
- Criação de índices;
- Armazenamento físico dos dados;
- Definição de visões sobre os dados armazenados;
- Atribuição de direitos de acesso;
- Políticas de backup dos dados.

Conceitos de Modelo Relacional

Conceitos de Modelo relacional

O modelo relacional é fundamentado no **conceito de relação**.

- Cada coluna de uma relação pode conter apenas **valores atômicos** (números e cadeias de caracteres).
- **Uma chave primária:** colunas cujos valores podem ser utilizados para identificar unicamente cada linha de uma relação.
- **Associações entre linhas:** valores de uma coluna fazem referência a valores de uma outra coluna. (chave estrangeira).
- Uma **chave estrangeira** também pode conter valores nulos, representados pela constante NULL.
- O NULL é normalmente usado para indicar que um valor não se aplica, ou é desconhecido, ou não existe.

Bancos	
Codigo	Nome
001	Banco do Brasil
033	Santander
237	Bradesco
341	Itaú

Tabela 1: Bancos Comerciais.

Pessoas	
CPF	Nome
86277635697	José da Silva
88208811874	Manoel da Silva
66516764743	Maria dos Santos

Tabela 2: Pessoas.

Conta_Corrente		
Banco	Pessoa	Numero
033	86277635697	98876788
237	86277635697	96645727
341	66516764743	9102947
001	88208811874	8120938

Tabela 3: Contas Correntes.

Mapeamento de Objetos para o Modelo Relacional

Mapeamento de Objetos para o Modelo Relacional

- Utilização de um SGBDR:** necessidade do mapeamento dos valores de atributos de objetos persistentes para tabelas.
- É a partir do modelo de classes que o mapeamento de objetos para o modelo relacional é realizado.
 - Semelhante ao de mapeamento do MER.
 - Diferenças em virtude de o modelo de classes possuir mais recursos de representação que o MER.
 - **Importante:** o MER e o modelo de classes não são equivalentes.
 - Esses modelos são frequentemente confundidos.
 - O MER é um modelo de dados; o modelo de classes representa objetos (dados e comportamento).

Passos para o mapeamento

Passo 1

Classes e seus atributos

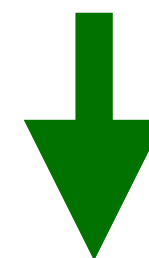
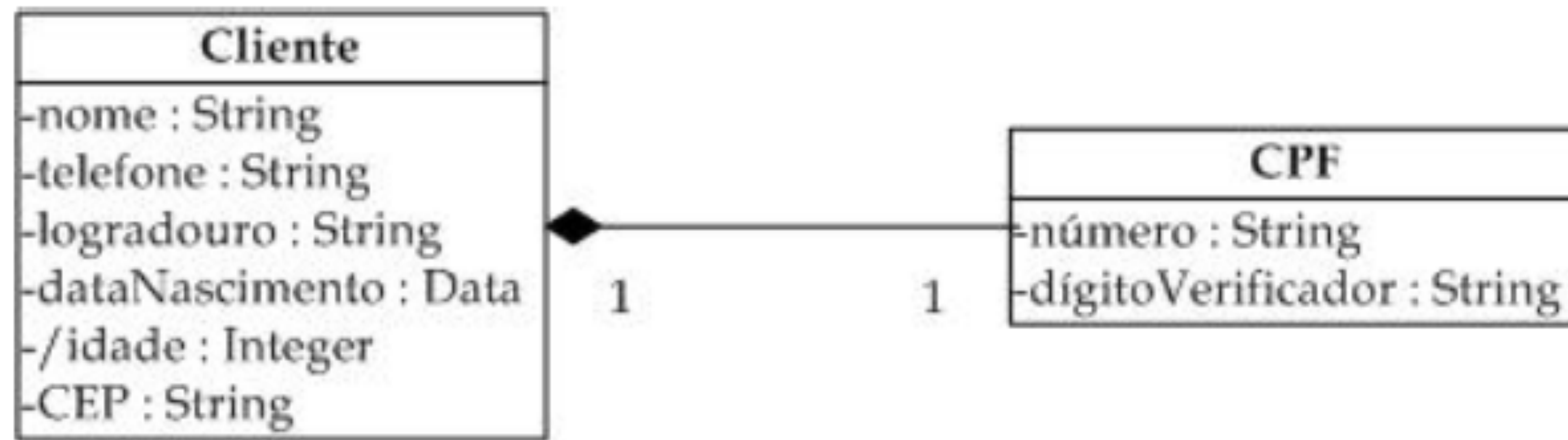
1 - Classes e seus atributos

Classes são mapeadas para relações.

- **Caso mais simples:** mapear cada classe como uma relação, e cada atributo como uma coluna.
- No entanto, pode não haver correspondência unívoca entre classes e relações.

Para atributos o que vale de forma geral é que **um atributo será mapeado para uma ou mais colunas.**

Nem todos os atributos de uma classe são persistentes (**atributos derivados**).



```
Cliente(id, CPF, nome, telefone, logradouro, dataNascimento, idCEP)
CPF(id, número, sufixo)
```

```
Cliente(id, nome, telefone, logradouro, dataNascimento, CPF, CEP)|
```

Passo 2

Associações

2 - Associações

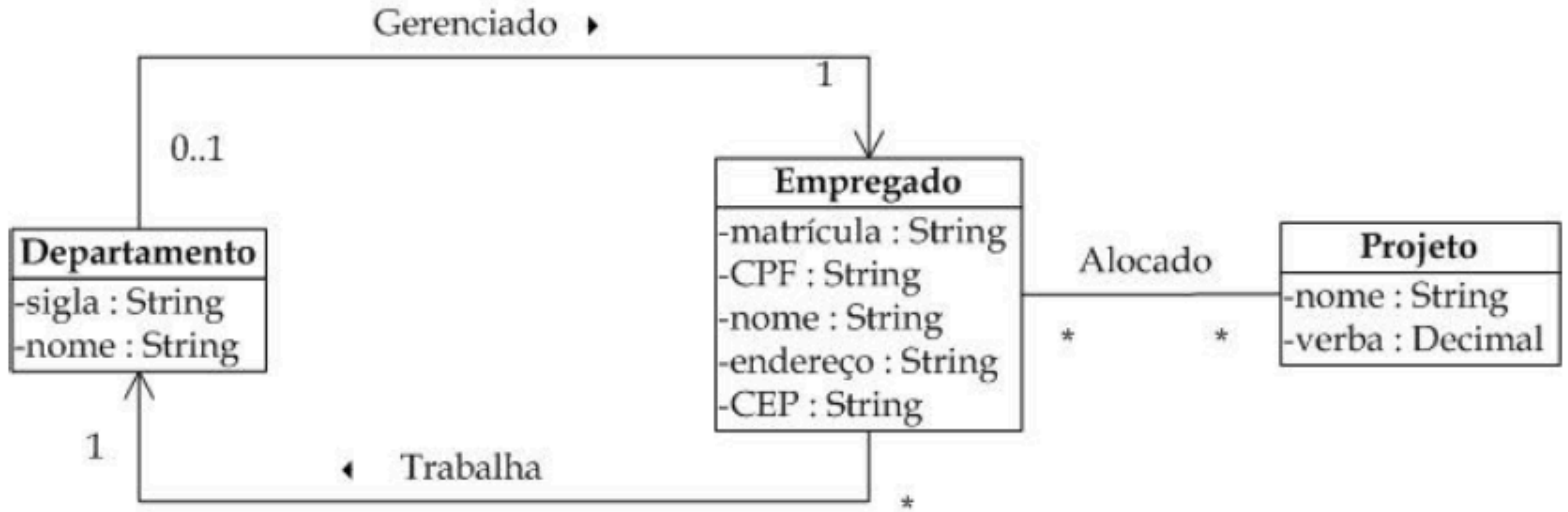
O procedimento utiliza o conceito de chave estrangeira.

Há três casos, cada um correspondente a um tipo de **conectividade**.

Nos exemplos dados a seguir, considere, sem perda de generalidade, que:

- há uma associação entre objetos de duas classes, **Ca** e **Cb**.
- **Ca** e **Cb** foram mapeadas para duas relações separadas, **Ta** e **Tb**.

Considere o seguinte diagrama de classes



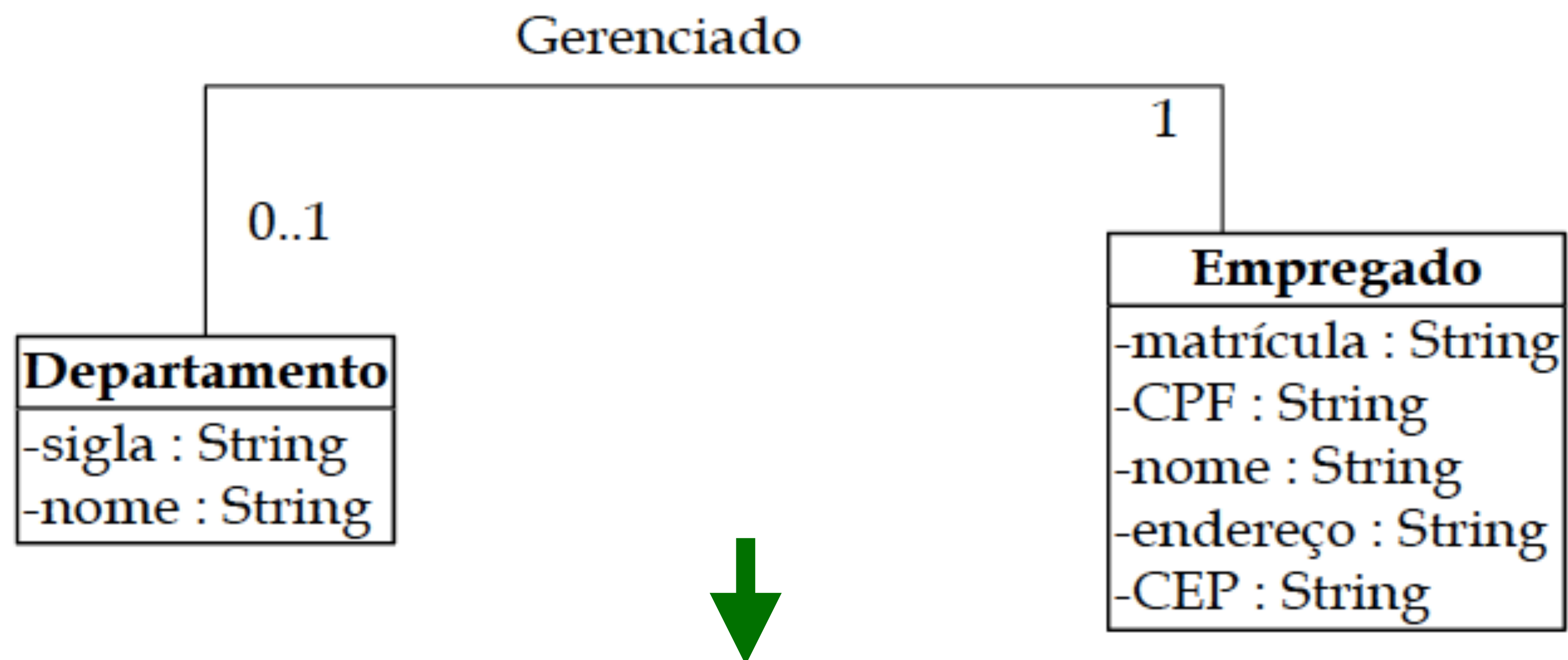
Passo 2.1

Associações 1..1

2.1 - Associações 1..1

Deve-se adicionar uma chave estrangeira em uma das duas relações para referenciar a chave primária da outra relação.

Escolha da relação na qual a chave estrangeira deve ser adicionada com base na **participação**.



```
Departamento(id, sigla, nome, idEmpregadoGerente )
Empregado( id, matrícula, CPF, nome, endereço, CEP )
```

Passo 2.2

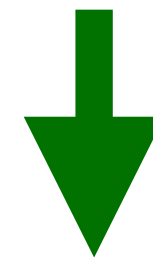
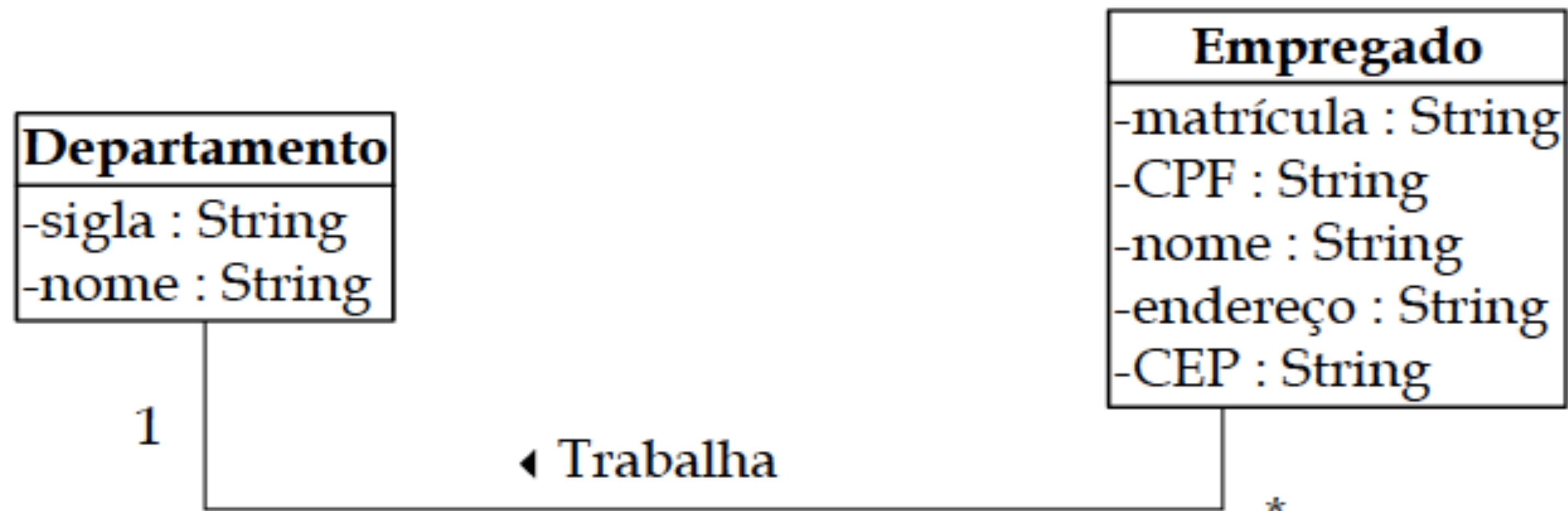
Associações 1..*

2.2 - Associações 1..*

Seja **Ca** a classe na qual cada objeto se associa com muitos objetos da classe **Cb**.

Sejam **Ta** e **Tb** as relações resultantes do mapeamento de **Ca** e **Cb**, respectivamente.

Neste caso, deve-se adicionar uma chave estrangeira em **Tb** para referenciar a chave primária de **Ta**.



```
Departamento( id, sigla, nome, idEmpregadoGerente )
Empregado( id, matrícula, CPF, nome, endereço, CEP, idDepartamento)
```

Passo 2.3

Associações *..*

2.3 - Associações *..*

Seja C_a a classe na qual cada objeto se associa com muitos objetos da classe C_b .

Sejam T_a e T_b as relações resultantes do mapeamento de C_a e C_b , respectivamente.

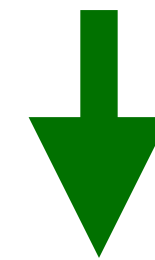
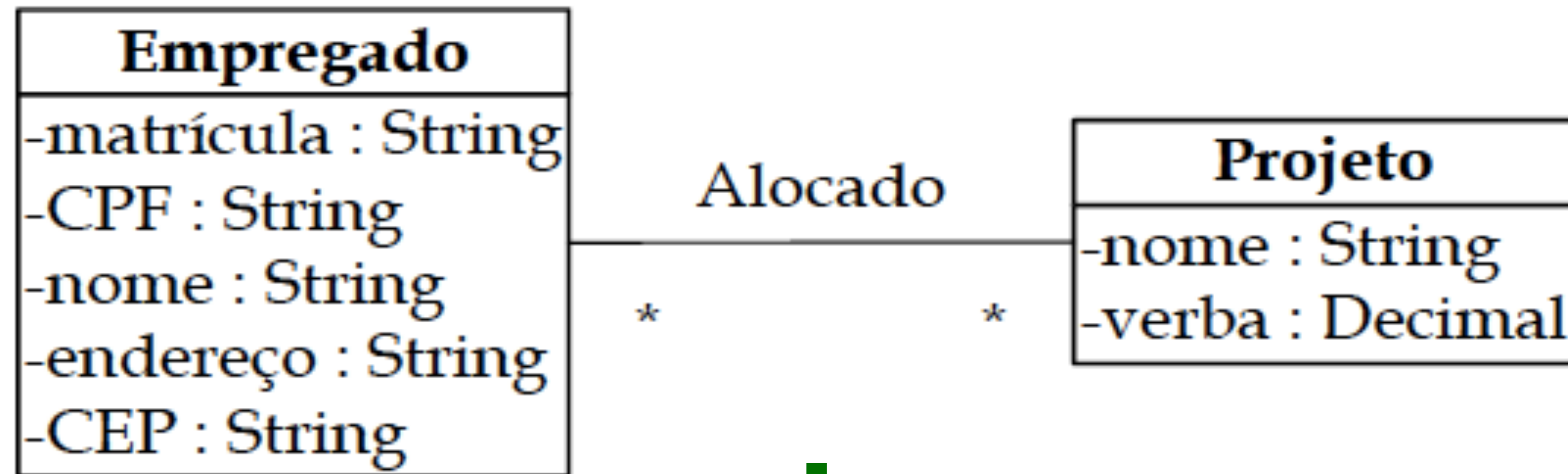
Uma **relação de associação** deve ser criada.

- Uma relação de associação serve para representar a associação muitos para muitos entre duas ou mais relações.

Equivalente à aplicação do mapeamento um para muitos duas vezes, considerando-se os pares (T_a, T_{assoc}) e (T_b, T_{assoc}) .

Alternativas para definir a chave primária de T_{assoc} .

- definir uma chave primária composta.
- criar uma coluna de implementação que sirva como chave primária simples da relação de associação.



Departamento(<u>id</u> , sigla, nome, idEmpregadoGerente) Empregado(<u>id</u> , matrícula, CPF, nome, endereço, CEP, idDepartamento) Alocação(<u>idProjeto</u> , <u>idEmpregado</u> , nome, verba) Projeto(<u>id</u> , nome, verba)
Departamento(<u>id</u> , sigla, nome, idEmpregadoGerente) Empregado(<u>id</u> , matrícula, CPF, nome, endereço, CEP, idDepartamento) Alocação(<u>id</u> , idProjeto, idEmpregado, nome, verba) Projeto(<u>id</u> , nome, verba)

Passo 2.4

Agregação

2.4 - Agregação

Forma especial de associação em que o mesmo procedimento para realizar o mapeamento de associações pode ser utilizado.

No entanto, a diferença semântica influi na forma como o SGBDR deve agir quando um registro da relação correspondente ao todo deve ser excluído ou atualizado.

- Remoção ou atualização em cascata.
- Pode ser implementado como gatilhos e procedimentos armazenados.

O padrão de acesso em agregações (composições) também é diferente do encontrado nas associações.

- Quando um objeto todo deve ser restaurado, é natural restaurar também os objetos parte.
- Em associações, isso nem sempre é o caso.
- Definição de índices adequados é importante para acesso eficiente aos objetos parte.

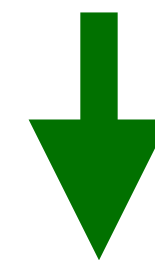
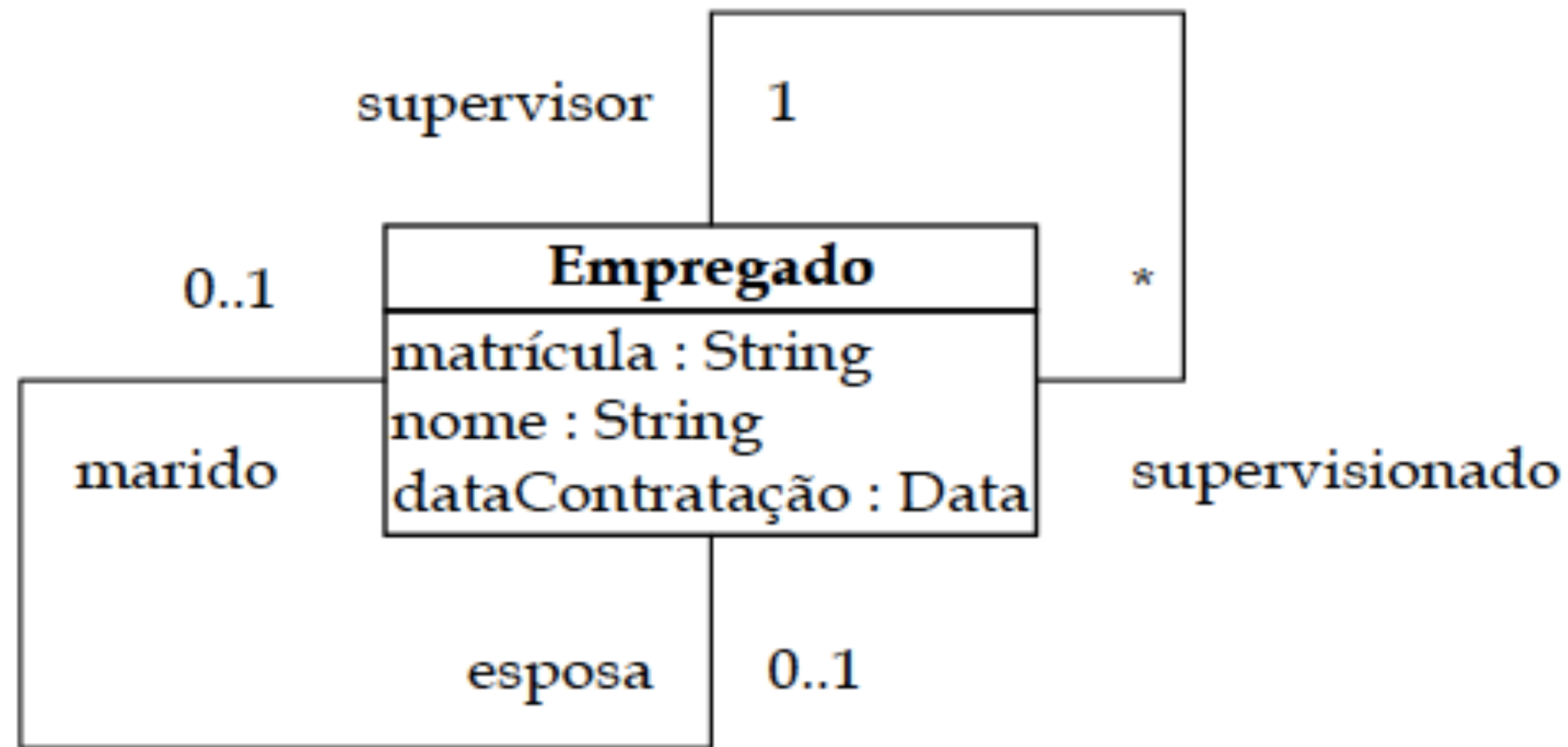
Passo 2.5

Associação Reflexiva

2.4 - Agregação

Forma especial de associação em que o mesmo procedimento para realizar o mapeamento de associações pode ser utilizado.

Em particular, em uma associação reflexiva de conectividade muitos para muitos, uma relação de associação deve ser criada.



```
Empregado(id, matrícula, nome, dataContratação, idCônjunge, idSupervisor)
```

Passo 3

Generalização

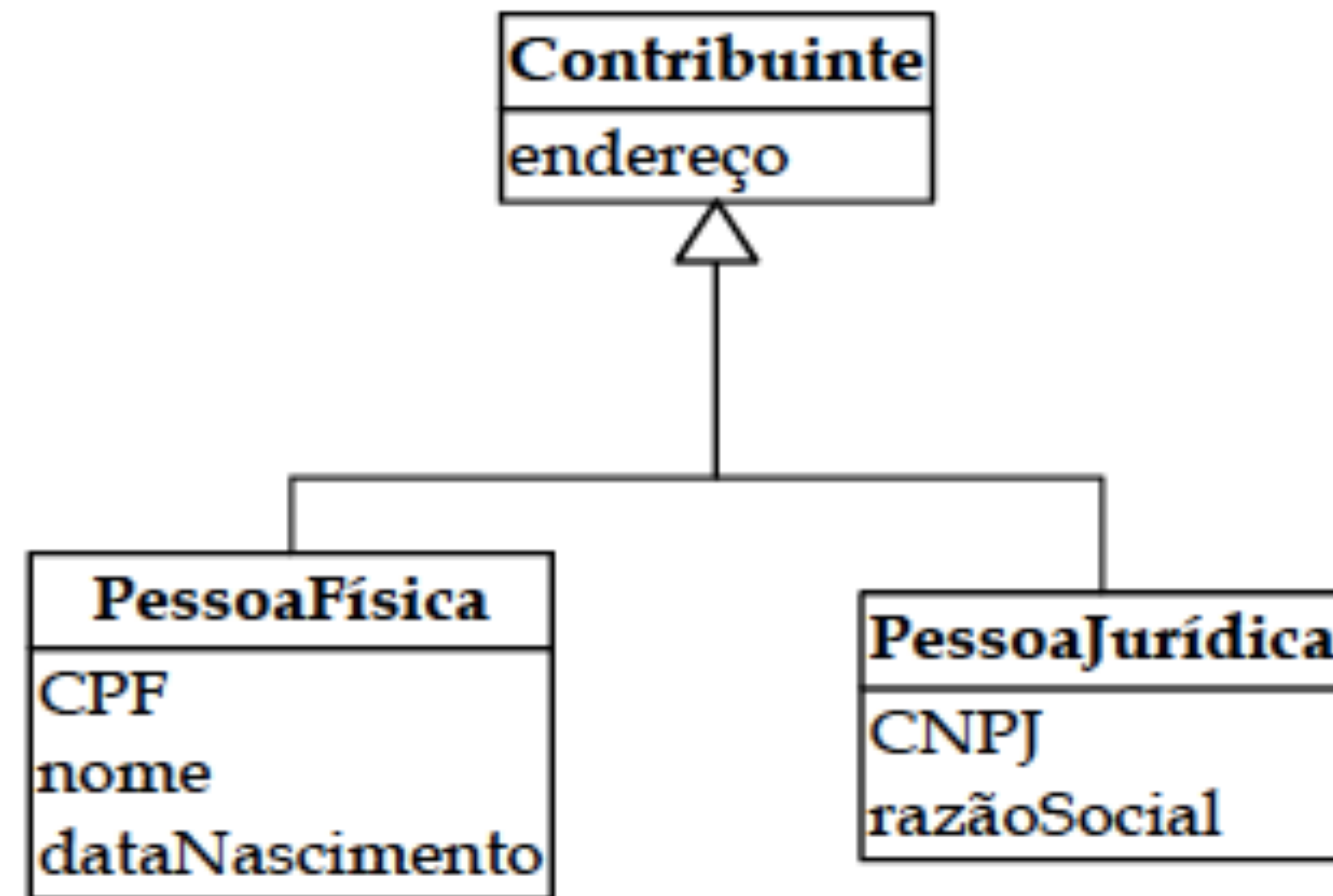
3 - Generalização

Três formas alternativas de mapeamento:

- Uma relação para cada classe da hierarquia
- Uma relação para toda a hierarquia
- Uma relação para cada classe concreta da hierarquia

Nenhuma das alternativas de mapeamento de generalização pode ser considerada a melhor dentre todas.

- Cada uma delas possui vantagens e desvantagens. Escolha de uma delas depende das do sistema sendo desenvolvido.
- A equipe de desenvolvimento pode decidir implementar mais de uma alternativa.



Contribuinte(<u>id</u> , endereço)
PessoaFísica(<u>id</u> , nome, dataNascimento, CPF, idContribuinte)
PessoaJurídica(<u>id</u> , CNPJ, razãoSocial, idContribuinte)
Pessoa(<u>id</u> , nome, endereço, dataNascimento, CPF, CNPJ, razãoSocial, tipo)
PessoaFísica(<u>id</u> , dataNascimento, nome, endereço, CPF)
PessoaJurídica(<u>id</u> , CNPJ, endereço, razãoSocial)

Vamos praticar

