

Análise e Projeto de Sistemas

Universidade Federal do Ceará – UFC

Campus de Quixadá

Curso de Sistemas de Informação

Prof. Lincoln Souza Rocha (lincolnrocha@ufc.br)

“Na época, Nixon estava normalizando as relações com a China. Eu pensei que se ele podia normalizar relações, eu também podia.”

(E.F. Codd)

MAPEAMENTO DE OBJETOS PARA O MODELO RELACIONAL

Esses slides são uma adaptação das notas de aula do professor Eduardo Bezerra autor do livro Princípios de Análise e Projeto de Sistemas com UML

Índice

- Introdução
- Projeto de banco de dados

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 ***persistentes*** e ***transientes***
 - 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*** (*impedance mismatch*) é um termo utilizado para denotar o problema das diferenças entre as representações do modelo OO e do modelo relacional

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

Conceito do Modelo Relacional

Alguns Conceitos

- ✓ O modelo relacional é fundamentado no conceito de **relação**.
- ✓ Cada coluna de uma relação pode conter apenas **valores atômicos**.
- ✓ 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.

Conceito do Modelo Relacional

Departamento			
<u>id</u>	sigla	nome	<u>idGerente</u>
13	RH	Recursos Humanos	5
14	INF	Informática	2
15	RF	Recursos Financeiros	6

Alocação		
<u>id</u>	<u>idProjeto</u>	<u>idEmpregado</u>
100	1	1
101	1	2
102	2	1
103	3	5
104	4	2

Projeto		
<u>id</u>	nome	Verba
1	PNADO	R\$ 7.000,00
2	BMMO	R\$ 3.000,00
3	SGILM	R\$ 6.000,00
4	ACME	R\$ 8.000,00

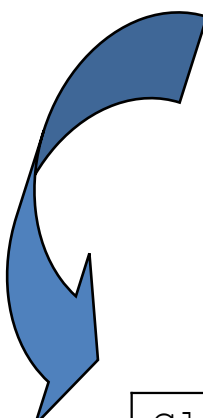
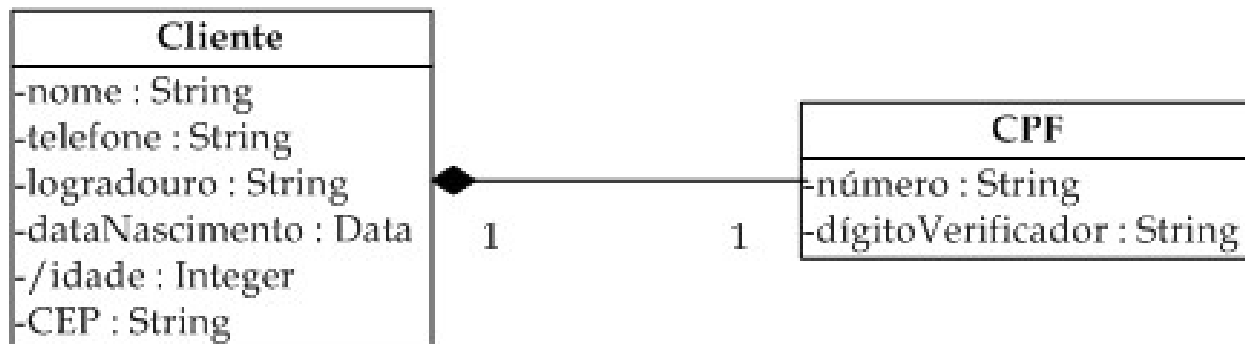
Conceito do Modelo Relacional

Empregado						
<u>id</u>	matrícula	CPF	nome	endereço	CEP	<u>idDepartamento</u>
1	10223	038488847-89	Carlos	Rua 24 de Maio, 40	22740-002	13
2	10490	024488847-67	Marcelo	Rua do Bispo, 1000	22733-000	13
3	10377	NULL	Adelci	Av. Rio Branco, 09	NULL	NULL
4	11057	0345868378-20	Roberto	Av. Apicás, 50	NULL	14
5	10922	NULL	Aline	R. Uruguaína, 50	NULL	14
6	11345	0254647888-67	Marcelo	NULL	NULL	15

Mapeamento: 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)

Mapeamento: Classes e seus Atributos



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

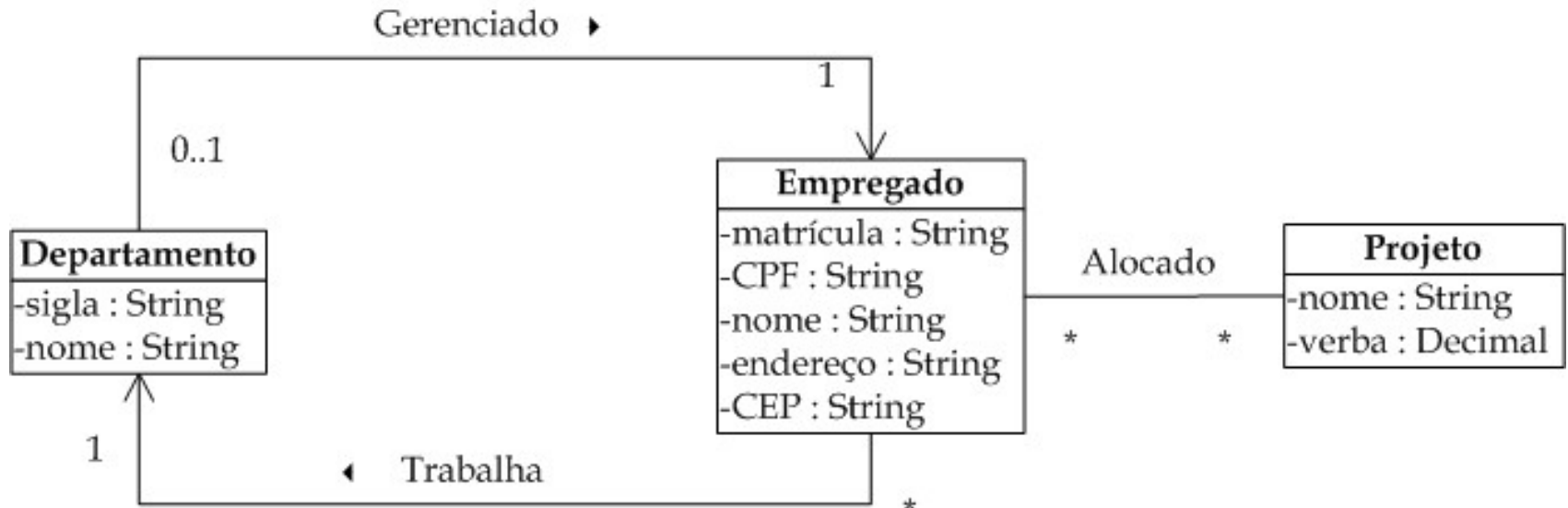
```
Cliente(id, nome, telefone, logradouro, dataNascimento, CPF, CEP)
```

Mapeamento de 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

Mapeamento de Associações

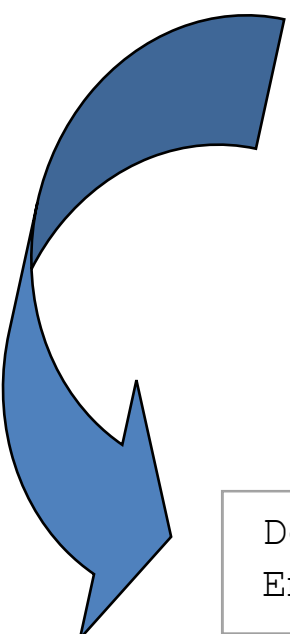
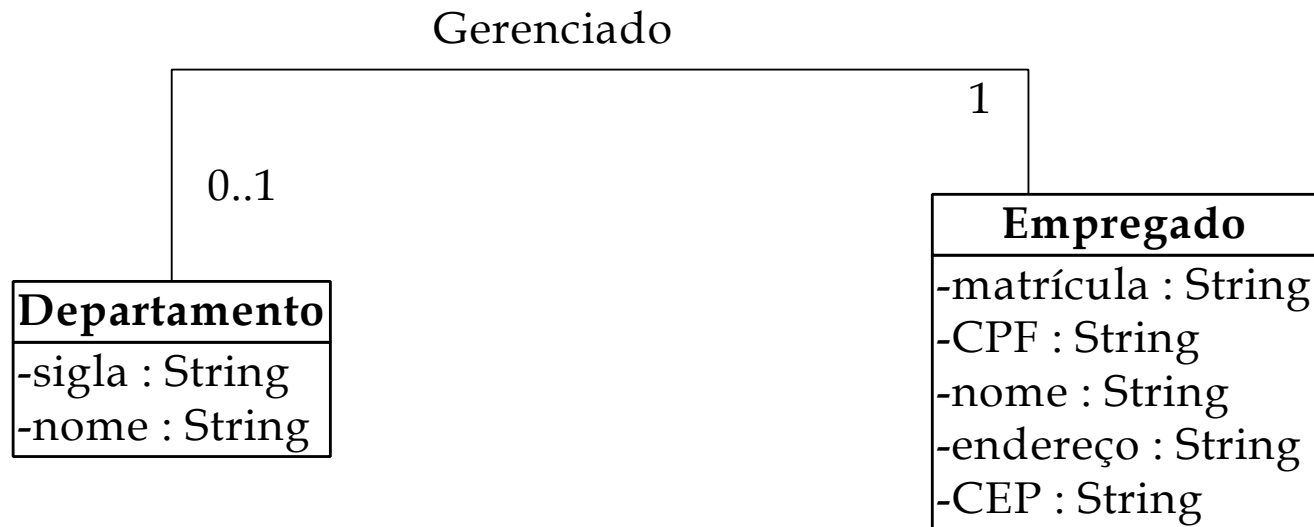
- Considere o diagrama de classes abaixo



Mapeamento de 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**
- Há três possibilidades acerca da conectividade
 - Obrigatória em ambos os extremos
 - Opcional em ambos os extremos
 - Obrigatória em um extremo e opcional no outro extremo

Mapeamento de Associações 1:1

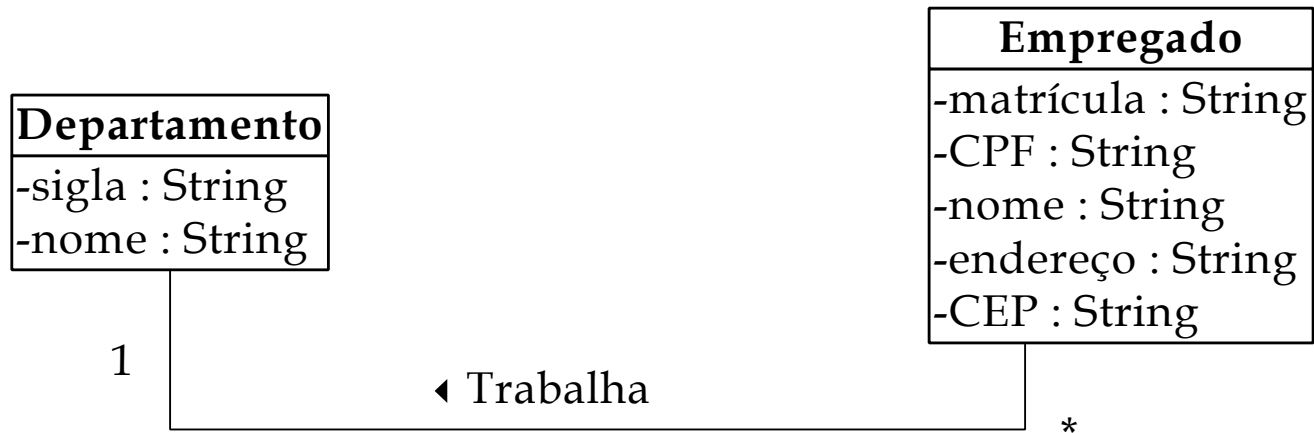


```
Departamento(id, sigla, nome, idEmpregadoGerente )  
Empregado( id, matrícula, CPF, nome, endereço, CEP )
```

Mapeamento de Associações 1:*

- 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
- Neste caso, deve-se adicionar uma chave estrangeira em T_a para referenciar a chave primária de T_b

Mapeamento de Associações 1:*



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

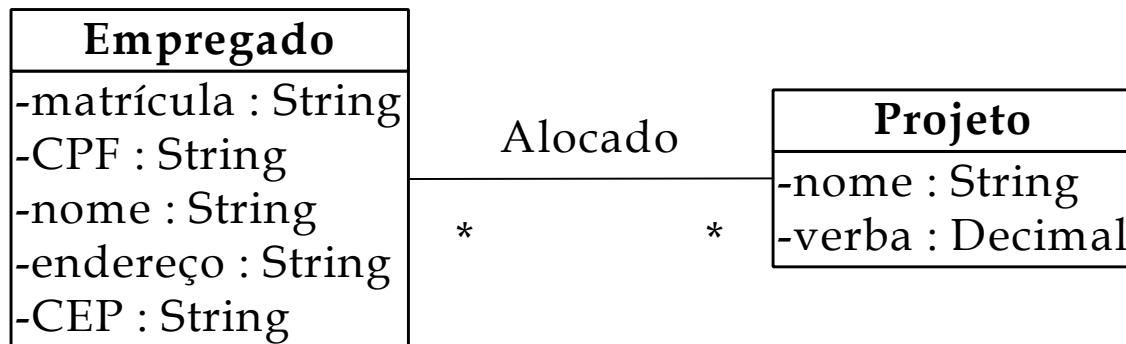
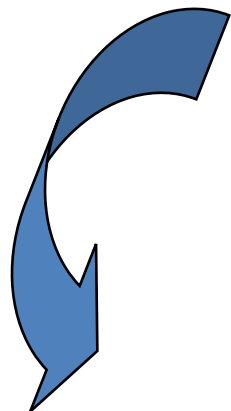
Mapeamento de Associações $^{*}:^{*}$

- 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.

Mapeamento de Associações $^{*:*}$

- Equivalente à aplicação do mapeamento um para muitos duas vezes, considerando-se os pares (Ta, Tassoc) e (Tb, Tassoc)
- Alternativas para definir a chave primária de Tassoc
 - Definir uma chave primária composta
 - Criar uma coluna de implementação que sirva como chave primária simples da relação de associação

Mapeamento de Associações *:*



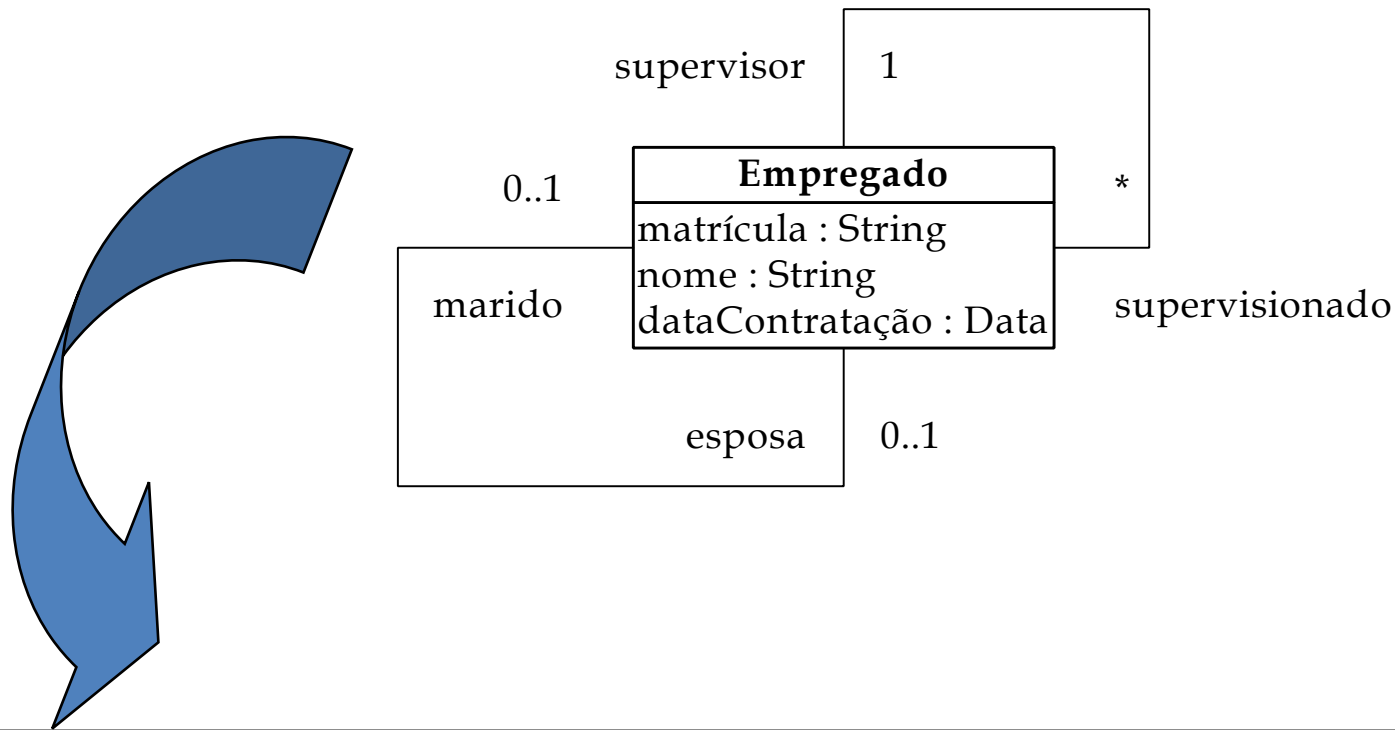
```
Departamento(id, sigla, nome, idEmpregadoGerente)
Empregado(id, matrícula, CPF, nome, endereço, CEP, idDepartamento)
Alocação(idProjeto, idEmpregado, nome, verba)
Projeto(id, nome, verba)
```

```
Departamento(id, sigla, nome, idEmpregadoGerente)
Empregado(id, matrícula, CPF, nome, endereço, CEP, idDepartamento)
Alocação(id, idProjeto, idEmpregado, nome, verba)
Projeto(id, nome, verba)
```

Mapeamento de Agregações

- A diferença semântica entre agregações e associações 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*

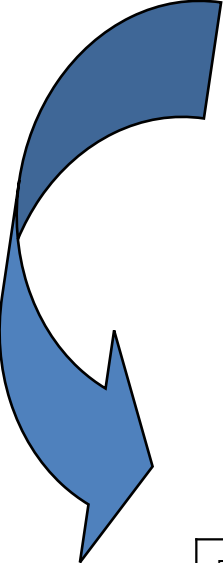
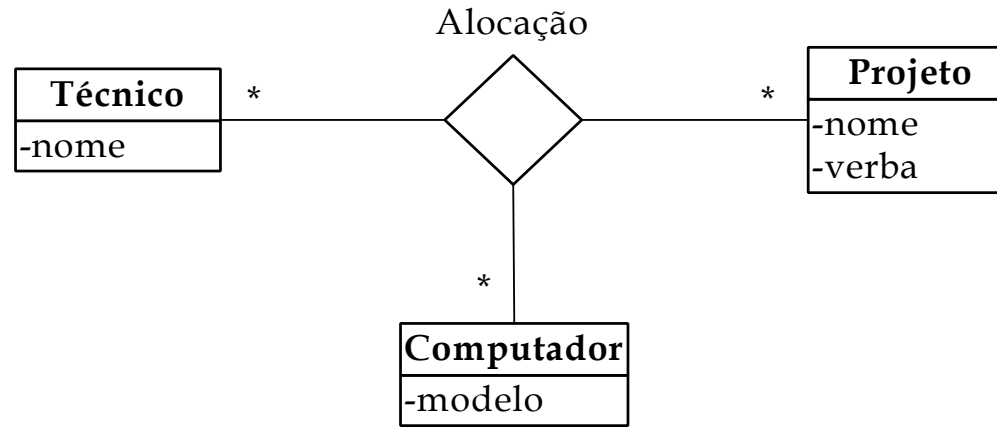
Mapeamento de Associações Reflexivas



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

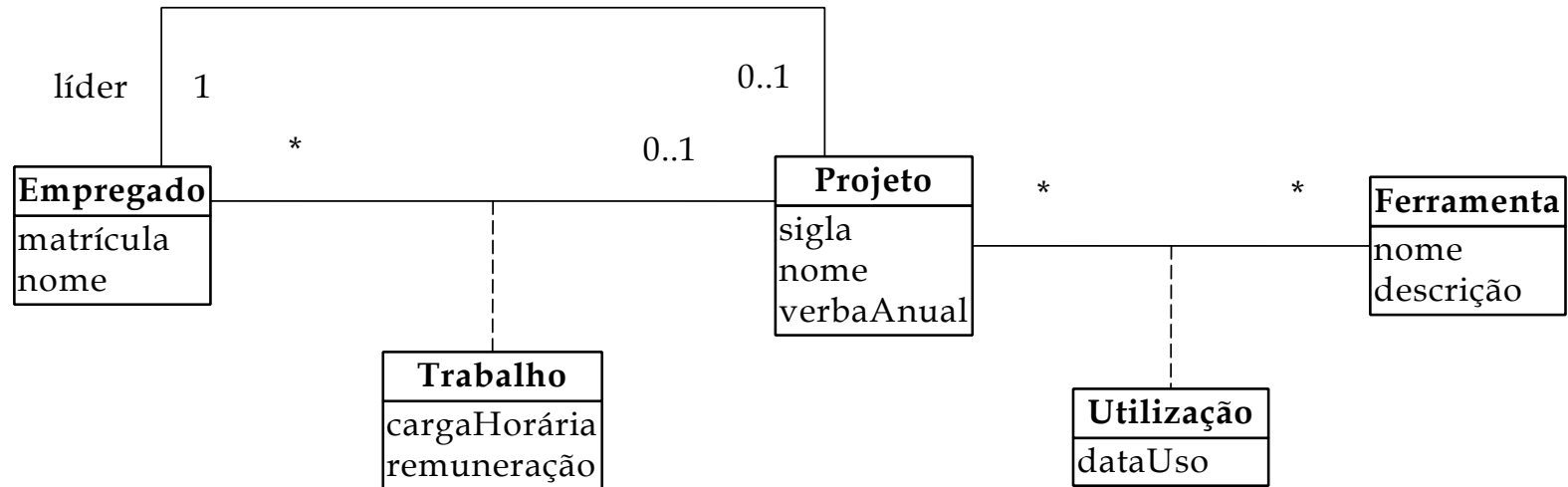
com OML - 2ª edição

Mapeamento de Associações n-áreas



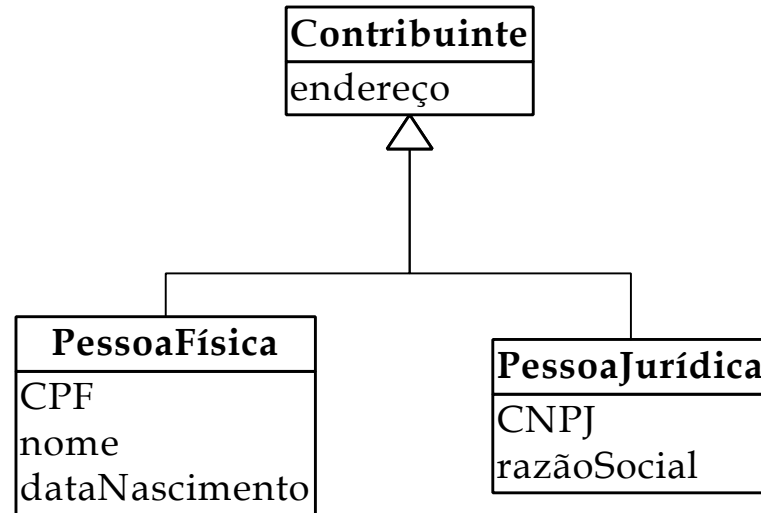
```
Técnico( id, nome )
Projeto( id, nome, verba )
Computador( id, modelo )
Alocação( id, idProjeto, idTécnico, idComputador )
```

Mapeamento de Classes Associativas



```
Empregado(id, matrícula, nome)
Projeto(id, sigla, nome, verbaAnual, idEmpregadoLíder)
Ferramenta(id, nome, descrição)
Utilização(id, idFerramenta, idProjeto, dataUso )
Trabalho(id, idEmpregado, idProjeto, cargaHorária, remuneração)
```

Mapeamento de Generalizações



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

Referências

- BEZERRA, E. Princípios de Análise e Projeto de Sistemas com UML. 2ª ed. Rio de Janeiro: Elsevier, 2007.
- FOWLER, M. 3. UML Essencial. 3. ed. Porto Alegre: Bookman, 2007.