

Projeto de Software

Mapeamento Objeto-Relacional

Lesandro Ponciano

2024

Objetivos da Aula

- Discutir o Mapeamento Objeto - Relacional
- Discutir exemplos de mapeamento
- Contextualizar padrões arquiteturais associados ao mapeamento

Relevância

- Por que Mapeamento Objeto-Relacional?
 - 1) Orientação a objetos tornou-se a forma usual para desenvolvimento de sistemas
 - 2) Banco de dados relacionais tiveram êxito na computação e dominam o mercado

No entanto, as duas abordagens são baseadas em princípios teóricos bem diferentes: abstração de comportamento *versus* tabulação

Tipos de Objetos

- **Objetos transientes**
 - Existe somente em tempo de execução
 - Objetos de controles e objetos de fronteiras, ex.: controladores, e interfaces
- **Objetos persistentes**
 - Perdura durante várias sessões se uso do sistema
 - Precisam ser armazenados quando a execução do sistema termina, e são restaurados quando uma outra execução é iniciada
- Nesta aula, sempre que falarmos de “objeto” nos referimos aos persistentes

Projeto de Banco de Dados

- Construção do esquema do banco de dados
- Criação de índices para agilizar o acesso aos dados armazenados
- Definição das estruturas de dados a serem utilizadas no armazenamento físico dos dados
- Definição de visões sobre as dados armazenados
- Atribuição de direitos de acesso (que usuários podem acessar que recursos)
- Definição de políticas de backup dos dados

Mapeamento

- Como se faz o mapeamento do modelo de classes no modelo relacional (do projeto de banco de dados)?
- Há ferramentas *Computer-Aided Software Engineering* CASE que fazem esse mapeamento das classes para o banco
- Há ferramentas que fazem o mapeamento do banco para as classes
- Como elas funcionam? Quais princípios seguem?

Exemplo

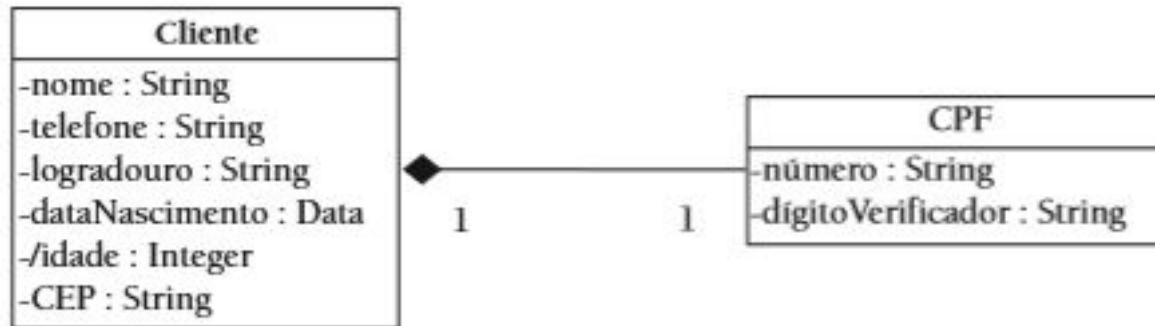
Departamento			
<u>id</u>	sigla	nome	<u>idGerente</u>
13	RH	Recursos Humanos	5
14	INF	Informática	2
15	RF	Recursos Financeiros	6

Empregado						
<u>id</u>	matricula	CPF	nome	endereço	CEP	<u>idDepartamento</u>
1	10223	038488847-89	Carlos	Rua 24 de Mato,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. Uruguatana, 50	NULL	14
6	11345	0254647888-67	Marcelo	NULL	NULL	15

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

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

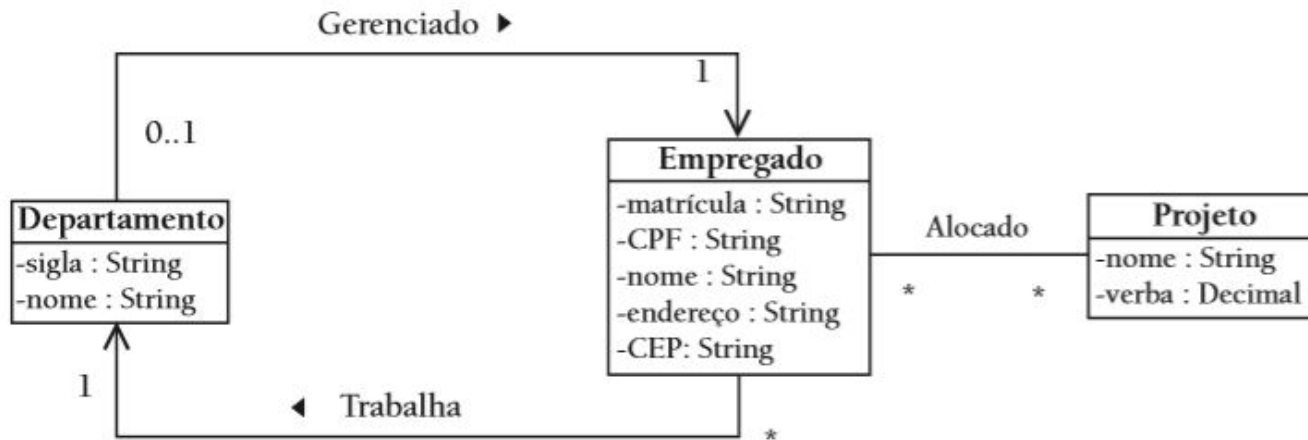
Classes e Seus Atributos



1ª	Cliente(<u>id</u> , CPF, nome, telefone, logradouro, dataNascimento, <u>idCEP</u>) CEP(<u>id</u> , número, sufixo)
2ª	Cliente(<u>id</u> , nome, telefone, logradouro, dataNascimento, CPF, CEP)

Conversão de Associações

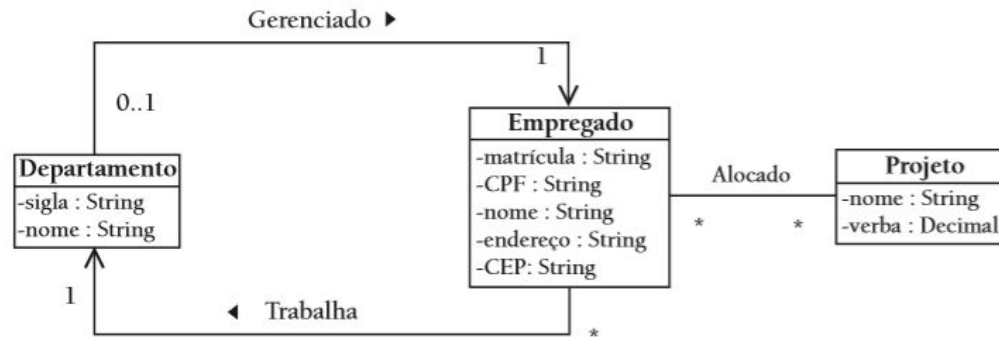
UM para UM



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

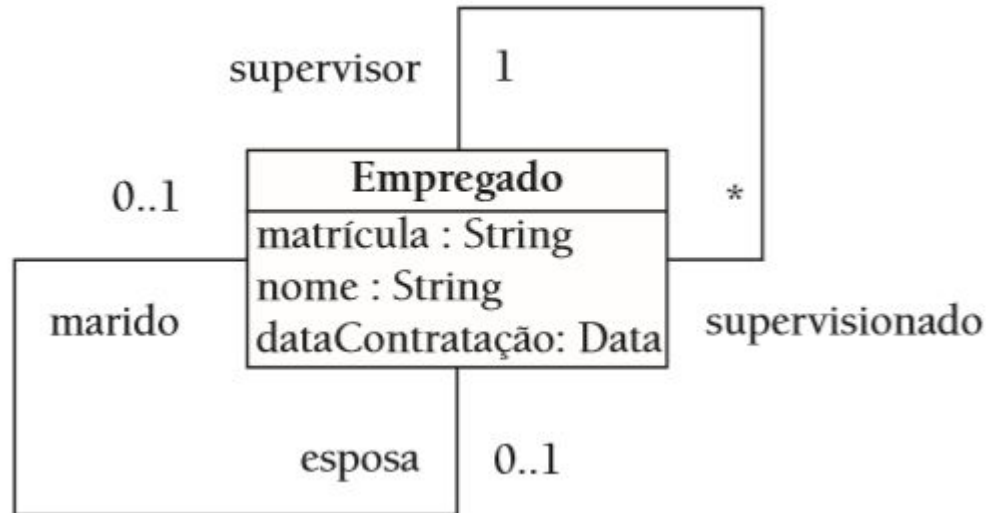
Conversão de Associações

UM para MUITOS



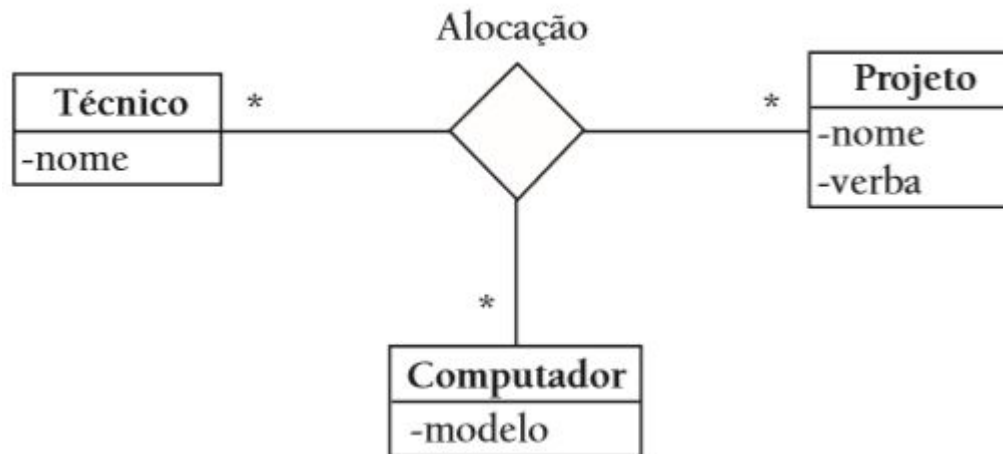
1ª alternativa	<code>Departamento(<u>id</u>, sigla, nome, <u>idEmpregadoGerente</u>)</code> <code>Empregado(<u>id</u>, matrícula, CPF, nome, endereço, CEP, <u>idDepartamento</u>)</code> <code>Alocação(<u>idProjeto</u>, <u>idEmpregado</u>, nome, verba)</code> <code>Projeto(<u>id</u>, nome, verba)</code>
2ª alternativa	<code>Departamento(<u>id</u>, sigla, nome, <u>idEmpregadoGerente</u>)</code> <code>Empregado(<u>id</u>, matrícula, CPF, nome, endereço, CEP, <u>idDepartamento</u>)</code> <code>Alocação(<u>id</u>, <u>idProjeto</u>, <u>idEmpregado</u>, nome, verba)</code> <code>Projeto(<u>id</u>, nome, verba)</code>

Agregações e Composições



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

Associações Ternárias



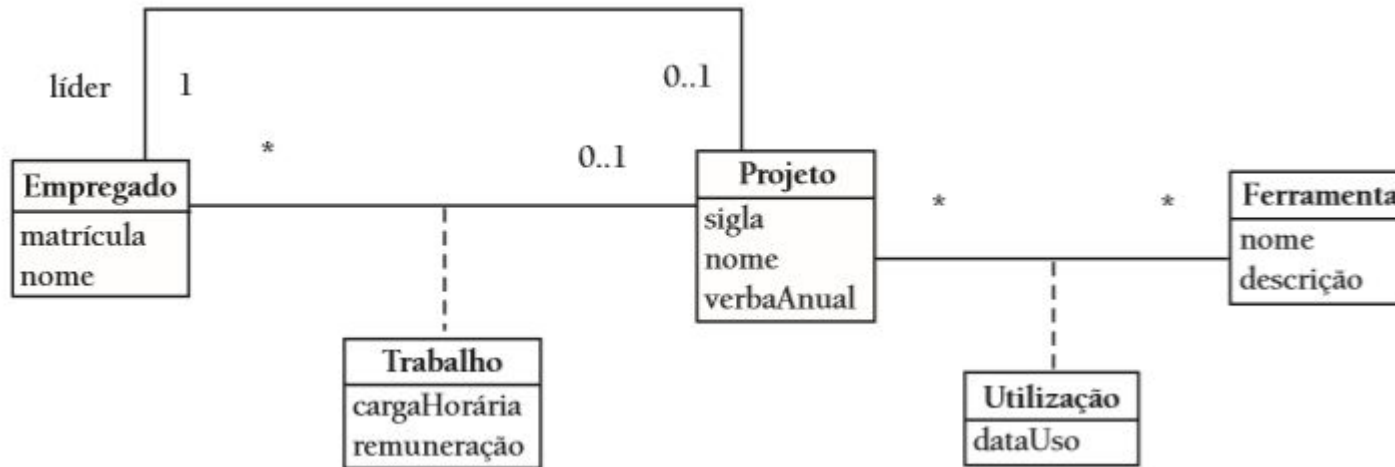
Técnico(id, nome)

Projeto(id, nome, verba)

Computador(id, modelo)

Alocação(id, idProjeto, idTécnico, idComputador)

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)

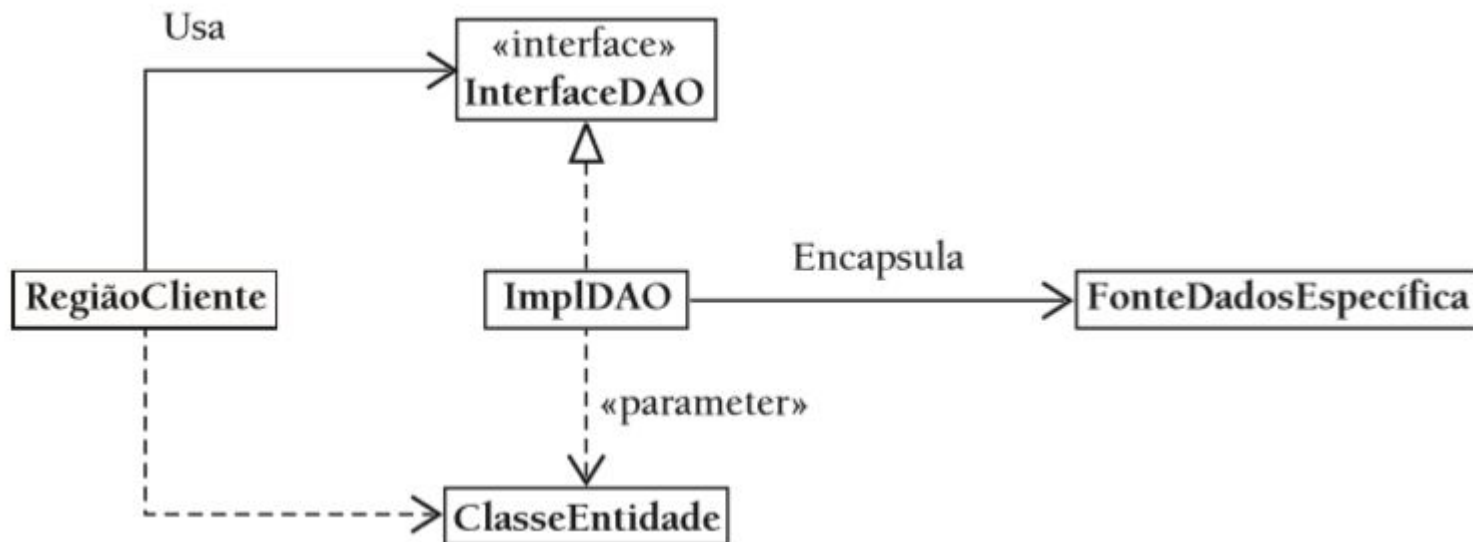
Generalização



1ª alternativa	Contribuinte(<u>id</u> , endereço) PessoaFísica(<u>id</u> , nome, dataNascimento, CPF, <u>idContribuinte</u>) PessoaJurídica(<u>id</u> , CNPJ, razãoSocial, <u>idContribuinte</u>)
2ª alternativa	Pessoa(<u>id</u> , nome, endereço, dataNascimento, CPF, CNPJ, razãoSocial, tipo)
3ª alternativa	PessoaFísica(<u>id</u> , dataNascimento, nome, endereço, CPF) PessoaJurídica(<u>id</u> , CNPJ, endereço, razãoSocial)

Construção da Persistência

Padrão DAO: uso de um objeto do acesso dos dados (DAO) para obter acesso e para armazenar dados em uma fonte. O objeto DAO controla a conexão com a fonte dos dados para obter e armazenar dados.



Construção da Persistência

- Object-Relational Mapping (ORM)
- Frameworks ORM
 - Tentam resolver o problema do descasamento de informações objeto-relacional
 - Programadores definem um arquivo de configuração, chamado de arquivo de mapeamento, com a correspondência objeto-relacional
 - Em tempo de execução, o framework mapeia as requisições

Referências

BEZERRA, E. Princípio de Análise e Projeto de Sistemas com UML, Rio de Janeiro, Elsevier, 2007. **(Capítulo 12)**

Projeto de Software

Prof. Dr. Lesandro Ponciano

<https://orcid.org/0000-0002-5724-0094>