

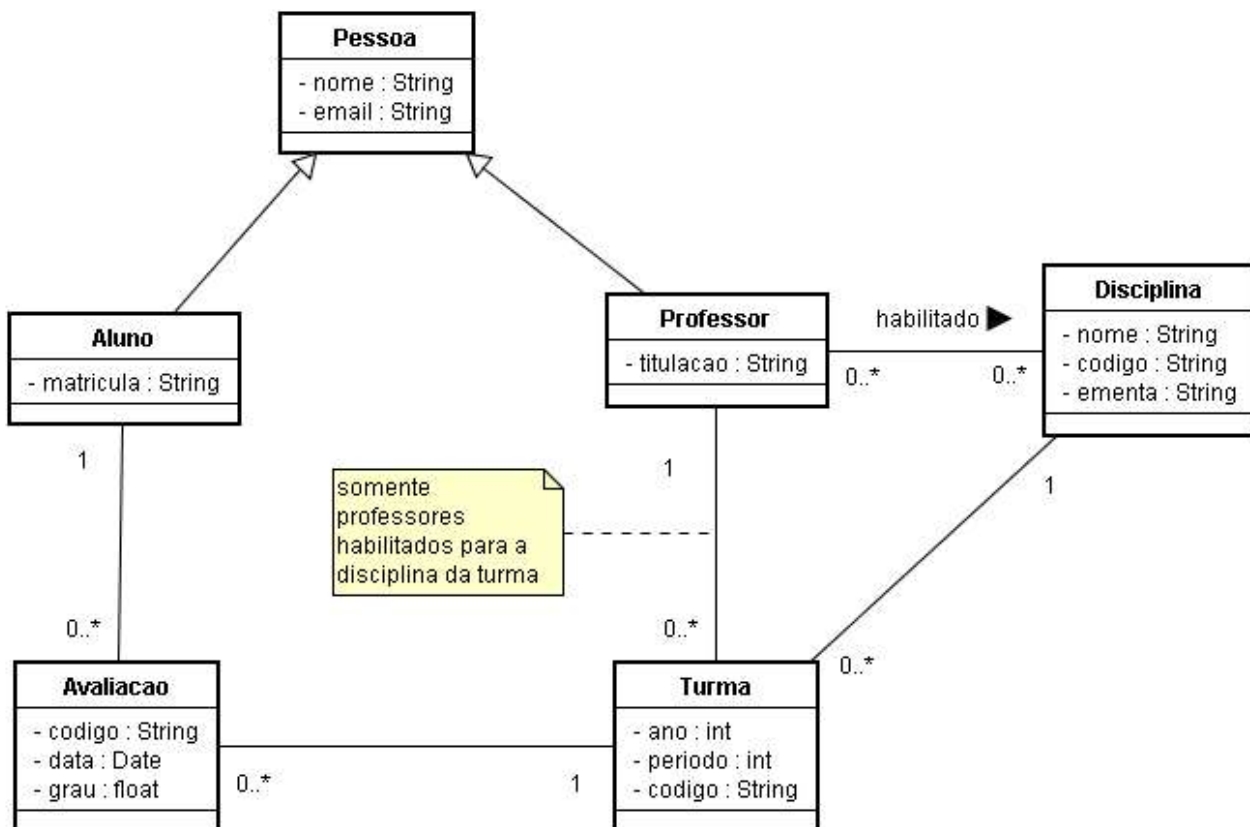
Exemplo de Mapeamento Objeto Relacional

contendo herança, relações de muitos para muitos e regras no diagrama

Hamilton Lima
www.athanazio.pro.br
 maio 2005

O fato de que, banco de dados que suportem o armazenamento de objetos não estar vastamente popularizado, nos força a criar soluções para mapear as classes projetadas em tabelas em um banco de dados relacional, este exemplo é uma sugestão de como formalizar as decisões envolvidas neste processo de mapeamento de objetos e suas associações em tabelas e um banco de dados relacional.

O diagrama



O mapeamento

| <i>Classe / associação</i> | <i>mapeamento</i> | <i>comentário</i> |
|----------------------------|---|--|
| Pessoa | Tabela PESSOA { ID int autoincremento NOME varchar(50) not null EMAIL varchar(50) } | Como haverá buscas por "pessoa" independente se é professor ou aluno, a solução de uma tabela para cada classe da herança está sendo adotada |

| <i>Classe / associação</i> | <i>mapeamento</i> | <i>comentário</i> |
|---|---|---|
| Aluno | Tabela ALUNO { ID int autoincremento PESSOA int foreign key PESSOA.ID MATRICULA varchar (10) not null } | |
| Professor | Tabela PROFESSOR { ID int autoincremento PESSOA int foreign key PESSOA.ID TITULACAO varchar (5) not null } | |
| (Habilitado) Professor – Disciplina | Tabela HABILITACAO { ID int autoincremento PROFESSOR int foreign key PROFESSOR.ID DISCIPLINA int foreign key DISCIPLINA.ID } | Como a associação é de muitos para muitos é necessário criar uma tabela para suportar a mesma |
| Disciplina | Tabela DISCIPLINA { ID int autoincremento NOME varchar (50) not null CODIGO varchar (10) not null EMENTA blob } | |
| Turma | Tabela TURMA { ID int autoincremento PROFESSOR int foreign key PROFESSOR.ID DISCIPLINA int foreign key DISCIPLINA.ID ANO int not null PERIODO int not null CODIGO varchar(15) not null } | A regra para permitir somente professores habilitados será verificada no sistema (1) |
| Avaliacao | Tabela AVALIACAO { ID int autoincremento ALUNO int foreign key ALUNO.ID TURMA int foreign key TURMA.ID CODIGO varchar(10) not null DATA date not null GRAU decimal } | |

(1) esta regra poderia ser verificada através de um trigger por exemplo, dependendo do banco de dados que estiver sendo adotado pela solução ou ainda da estratégia de usar ou não triggers, evitando a dependência de uma tecnologia específica.

Alternativas de mapeamento

somente uma tabela

| <i>Classe / associação</i> | <i>mapeamento</i> | <i>comentário</i> |
|--------------------------------|--|---|
| Pessoa, Aluno, Professor | Tabela PESSOA { ID int autoincremento NOME varchar(50) not null EMAIL varchar(50) MATRICULA varchar (10) TITULACAO varchar (5) TIPO varchar(1) not null } | Nesta estratégia todos os atributos das classes envolvidas na herança são mapeados para somente uma tabela, adicionando uma coluna de tipo que pode ter A para aluno e P para professor |

Uma tabela para cada classe filha no ultimo nivel da herança

| <i>Classe / associação</i> | <i>mapeamento</i> | <i>comentário</i> |
|--------------------------------|--|-------------------|
| Aluno | Tabela ALUNO { ID int autoincremento NOME varchar(50) not null EMAIL varchar(50) MATRICULA varchar (10) not null } | |
| Professor | Tabela PROFESSOR { ID int autoincremento NOME varchar(50) not null EMAIL varchar(50) TITULACAO varchar (5) not null } | |