



# Projeto Orientado a Objetos

## Exercício

# Exercício – Usando JPA

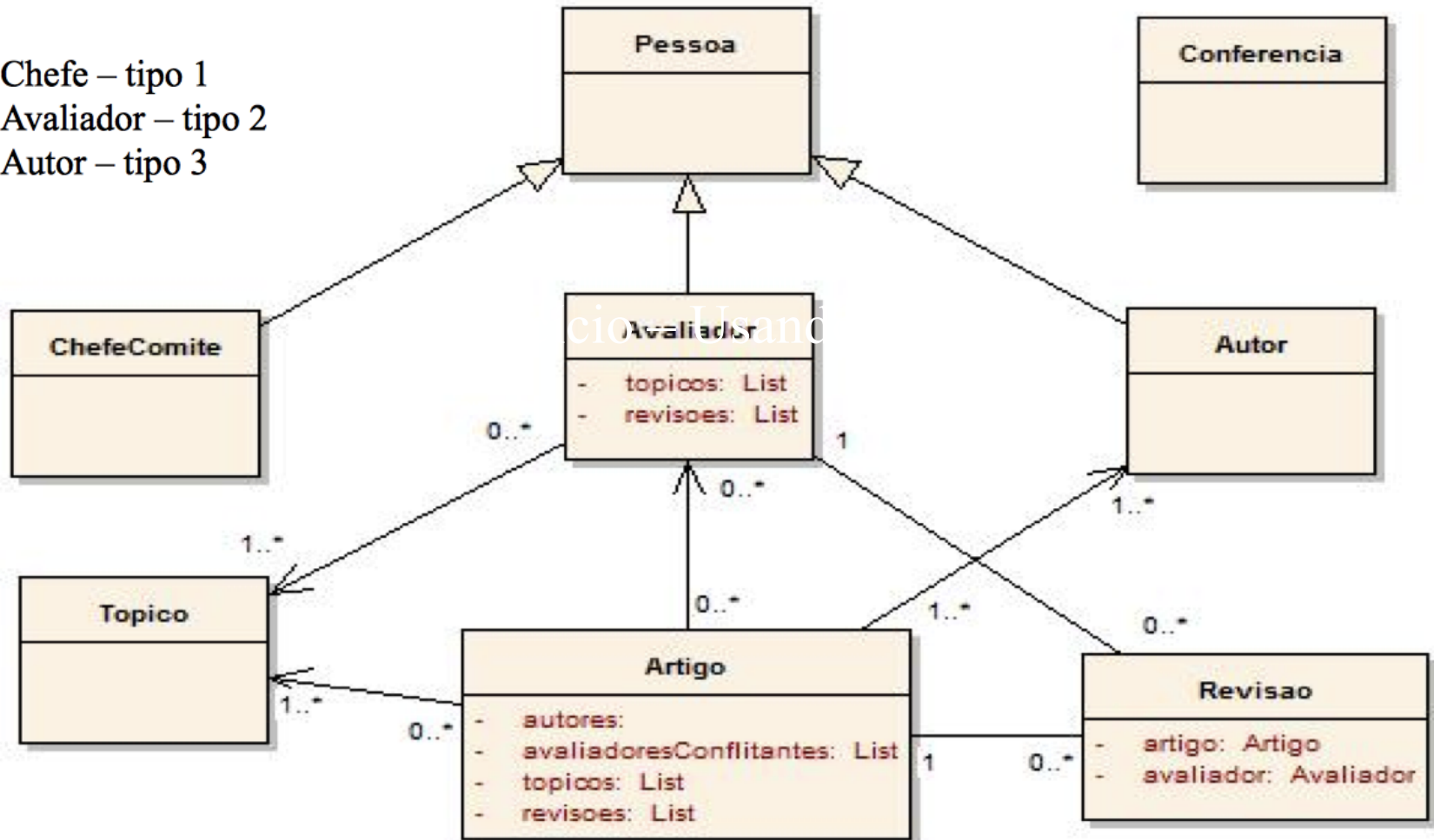
1. Descompacte o arquivo conferenciaJPAAlunos.zip. Este arquivo contém todas as classes necessárias para executar o problema da Conferencia utilizando a JPA. Complete o que está faltando:

- Faça o mapeamento das classes ChefeComite e Autor usando notações. (definição das tabelas correspondentes)
- Complete o mapeamento da classe Artigo (atributo notaFinal e associação Artigo-Autores (definida no método getAutores da classe Artigo)).
- Implemente o método distribuiArtigos da classe Conferencia.

Obs: não esqueça de criar todas as tabelas antes de executar o teste.

# Exercício – Usando JPA

Chefe – tipo 1  
Avaliador – tipo 2  
Autor – tipo 3



# JPA - Comentários

- JPA – Java Persistence API
- O Entity Manager é a fachada para o subsistema de persistência.
- O diretório META-INF contém o arquivo persistence.xml que possui as configurações do Entity Manager usado (neste exemplo o toptlink) e do Banco de Dados (neste exemplo o MySQL).
- O mapeamento é feito através de anotações ou de um arquivo .xml
- As anotações representando o mapeamento dos atributos são colocadas todas juntas aos atributos ou todas juntas aos métodos.
- Quando o atributo tem o mesmo nome da coluna da tabela, o seu mapeamento é automático, não precisando ser definido.
- @Transient – atributo que não vai para a base de dados, ou seja, não vai ser persistido.
- As classes de entidade precisam pelo menos ter 1 construtor sem parâmetro.

# JPA - Comentários

## Associações 1-1 ou 1-N

- Associações 1-1 ou 1-N: O mapeamento é definido no lado onde está a chave estrangeira (classe que será o owner da associação) (`@ManyToOne` e `@OneToOne`). Se tiver atributo inverso, na outra classe define-se o `mappedBy` (`@OneToMany` e `@OneToOne`). Exemplos: Revisao-Avaliador e Revisao-Artigo
- Associações N-N: Como as tabelas correspondentes às classes não possuem chave estrangeira, o mapeamento pode ser colocado em qualquer uma das classes. Exemplos: Artigo-Autor, Avaliador-Topico
- `@EmbeddedId`: algumas implementações da JPA, incluindo o top-link, não deixam colocar `@Id` em 2 atributos formando uma chave composta. Então é necessário criar uma nova classe com os atributos que formam a chave primária e no atributo composto correspondente é colocada a notação `@EmbeddedId`.

# JPA - Comentários

- Exemplos de comandos usados na JPA:

`em.persist(avaliador);`

`Topico topico = (Topico) em.find(Topico.class, nomeTopico);`

`List list = Conferencia.em.createQuery("select a from Conferencia a").getResultList();`

`em.merge(artigo);`

- Atualizações precisam ser feitas dentro de transações (persist, merge, remove, and refresh).

`em.getTransaction().begin()`

...

`em.getTransaction().commit()`

# JPA

Algumas dicas:

1. Colocar no CLASSPATH o postgresql-9.0-801.jdbc4.jar (se estiver usando o PostgreSQL).
  2. Para cada ferramenta, colocar no CLASSPATH os .jar necessários:
    - Toplink: toplink-essenciais.jar (usado no exemplo)  
(<http://www.oracle.com/technology/products/ias/toplink/jpa/download.html>)
    - OpenJPA  
(<http://openjpa.apache.org/>)
    - Hibernate EntityManager  
(<http://www.hibernate.org/397.html>)
  3. Fazer o mapeamento usando anotações nas classes.
- \* A documentação sobre a JPA está em ejb-3\_0-fr-spec-persistence
- \* O pacote conferenciaJPAAlunos e diretório META-INF devem estar no diretório raiz do projeto.