



# Curso de **Java8** para **Web**

Professor  
**Antonio Benedito Coimbra Sampaio Jr**

abc  | Treinamentos

[www.abctreinamentos.com.br](http://www.abctreinamentos.com.br)

# Terceira Disciplina

JEE - Persistência de Dados com JDBC e Hibernate

- **UNIDADE 1:** Arquitetura JEE
- **UNIDADE 2:** Introdução a Banco de Dados com Oracle
- **UNIDADE 3:** Persistência de Dados com JDBC
- **UNIDADE 4: Framework Hibernate**
- **UNIDADE 5:** Introdução ao JPA

# UNIDADE 4

# **FRAMEWORK HIBERNATE**

# Introdução ao Hibernate

# Framework Hibernate

## Motivação

- A utilização de código SQL dentro de uma aplicação agrava o problema da dependência de plataforma de banco de dados e complica, em muito, o trabalho de mapeamento entre classes e o banco de dados relacional.

## O que é um Framework?

- É um conjunto de componentes de software que provê uma arquitetura e estrutura básica para o desenvolvimento de uma aplicação.

## O que é o Hibernate?

- É um framework para o mapeamento objeto relacional (ORM) escrito na linguagem Java.
- Existe uma versão para a plataforma .Net conhecida como NHibernate.

# Framework Hibernate

## O que é o Hibernate?

- É um framework consolidado para fazer persistência (mundo OR ↔ mundo OO).
- É mantido por uma comunidade muito ativa (<http://hibernate.org>).
- O maior esforço para usá-lo está na construção e manutenção dos mapeamentos.
- O overhead é extremamente satisfatório.
- Existem muitas ferramentas de apoio ao desenvolvimento usando Hibernate, tais como os *plug-ins* para o Eclipse.
- O uso adequado dessas ferramentas simplifica o trabalho do desenvolvedor.

## Quais os outros Java ORM Frameworks?

- Enterprise JavaBeans, Java Data Objects.
- Castor, TopLink, Spring DAO, etc.

# Framework Hibernate

## Principais Vantagens

- **Produtividade:**
  - Elimina muito código repetido. Foco na lógica de negócio;
  - O esquema do banco de dados é gerado automaticamente.
- **Mantém:**
  - Poucas linhas de código;
  - Fácil de entender e administrar mudanças em modelos de objetos.
- **Performance:**
  - *Lazy Loading* – As associações só são chamadas quando for necessário;
  - Faz uso de *caching*.
- **Independência de BD:**
  - O banco de dados é abstraído;
  - Pode ser configurado fora da aplicação.

# Framework Hibernate

## Produtos

### Hibernate ORM



Domain model persistence for relational databases

[More ↗](#)

### Hibernate Search



Full-text search for your domain model

[More ↗](#)

### Hibernate Validator



Annotation based constraints for your domain model

[More ↗](#)

### Hibernate OGM



Domain model persistence for NoSQL datastores

[More ↗](#)

### Hibernate Tools



Command line tools and IDE plugins for your Hibernate usages

[More ↗](#)

### Others



We like the symmetry, everything else is here

[Even more ↗](#)

## Versão Atual

5.2.6.Final



2016-12-20 **stable**

Maven gav: `org.hibernate:hibernate-core:5.2.6.Final`

Java 8, JCache support, hibernate-entitymanager consolidation

[More on this release ↗](#)

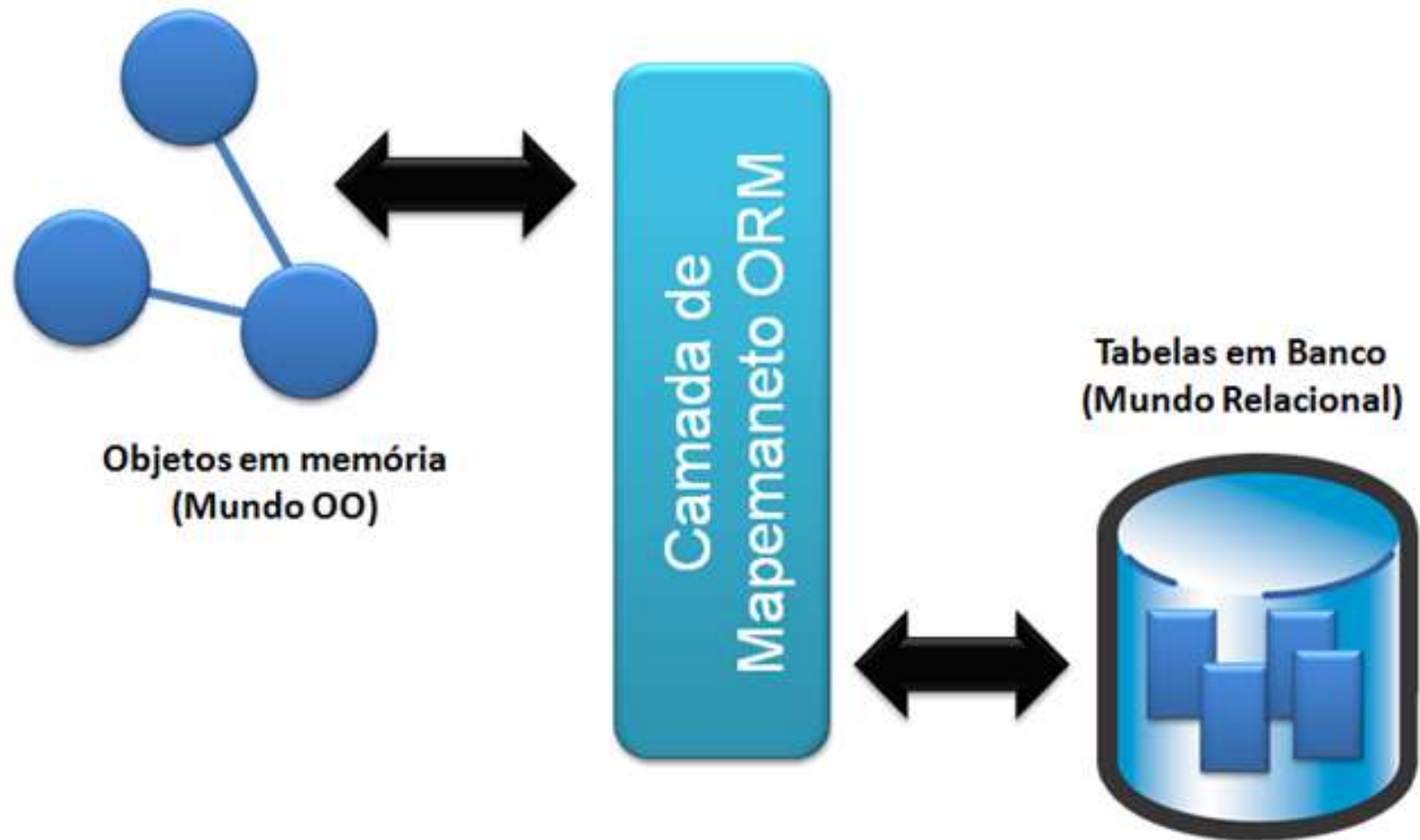


# Mapeamento ORM

## O que é o Mapeamento Objeto-Relacional

- É uma técnica utilizada para reduzir a impedância da programação orientada aos objetos utilizando bancos de dados relacionais.
- As tabelas do banco de dados são representadas através de classes (DAOs) e os registros de cada tabela são representados como instâncias das classes (VOs).
- Com esta técnica, o desenvolvedor não se preocupa com os comandos SQL, pois o framework de ORM faz todo o trabalho de persistência.
- A ideia da persistência O/R é reunir as vantagens de se utilizar um modelo orientado a objetos para a construção de uma aplicação, com a performance e a confiabilidade dos bancos de dados relacionais.

# Mapeamento ORM



<http://elderstroparo.blogspot.com>

# Mapeamento ORM

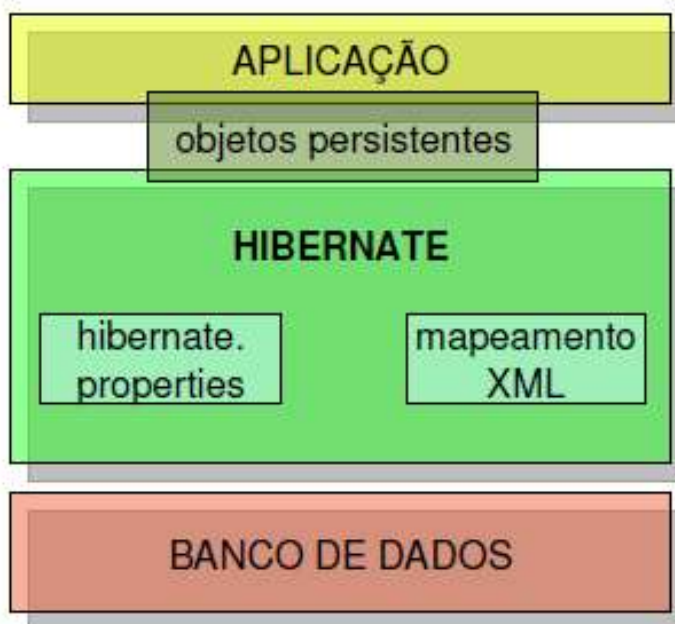
## Principais Características

- Programação OO (herança, polimorfismo, agregação, associação, composição e coleções Java).
- Sem aumento de tempo na construção da aplicação.
- Gratuito e aberto.
- Portável para todos os bancos compatíveis com o padrão SQL.
- Implementa mecanismos de mapeamento:
  - Classes Java  $\Leftrightarrow$  Tabelas em SGBDs relacionais;
  - Tipos Java  $\Leftrightarrow$  Tipos SQL.
- O Hibernate objetiva reduzir em cerca de 95% do tempo de desenvolvimento de tarefas relacionadas à persistência!
- O Hibernate persiste objetos Java comuns (POJO).

# Mapeamento ORM

## Principais Características

- Usa reflexão para acessar as propriedades persistentes de um objeto.
- As classes persistentes são definidas (descritas) em documentos de mapeamento.
- Arquivos XML são usados para descrever os campos, associações e subclasses persistentes.
- Os Mapeamentos são "compilados" na inicialização da aplicação.



# Exercícios

1) [CESPE - 2009 - SECONT-ES] O Hibernate, um framework de mapeamento objeto relacional (ORM), cria uma camada persistência na solução desenvolvida, o que permite ligar os objetos aos bancos de dados relacionais. Entre seus serviços, o Hibernate provê um meio de se controlar transações, por meio de métodos de suas interfaces session e transaction, tendo ainda suporte a herança e polimorfismo. É distribuído sob a licença LGPL, o que permite seu uso em projetos comerciais ou open source.

a) Certo   b) Errado

2) [CESPE - 2010 - TRE-BA] Quanto ao Hibernate, julgue o item abaixo.  
O Hibernate, um framework para o mapeamento objeto- relacional, é escrito na linguagem Java e, por isso, somente pode ser executado no ambiente Java.

a) Certo   b) Errado

# Exercícios

1) [CESPE - 2009 - SECONT-ES] O Hibernate, um framework de mapeamento objeto relacional (ORM), cria uma camada persistência na solução desenvolvida, o que permite ligar os objetos aos bancos de dados relacionais. Entre seus serviços, o Hibernate provê um meio de se controlar transações, por meio de métodos de suas interfaces session e transaction, tendo ainda suporte a herança e polimorfismo. É distribuído sob a licença LGPL, o que permite seu uso em projetos comerciais ou open source.

**a) Certo**   b) Errado

2) [CESPE - 2010 - TRE-BA] Quanto ao Hibernate, julgue o item abaixo.  
O Hibernate, um framework para o mapeamento objeto- relacional, é escrito na linguagem Java e, por isso, somente pode ser executado no ambiente Java.

**a) Certo**   **b) Errado**

# Exercícios

- 3) [COPEVE-UFAL - 2012 - ALGÁS] Dadas as seguintes afirmações,
- I. Na arquitetura do JDBC, a diferença entre os tipos Statement e PreparedStatement é o fato do PreparedStatement manter os dados criptografados durante o tráfego entre o cliente e o servidor do SGBD.
  - II. Um software que utiliza o framework Hibernate, necessariamente não utiliza JDBC.
  - III. O framework Hibernate permite a implementação de transações na camada de persistência (aplicação), mesmo que o SGBD utilizado não implemente esse conceito.
  - IV. O framework Hibernate é responsável por realizar o mapeamento objeto-relacional, mas possui a restrição de só aceitar chaves primárias compostas em relacionamentos, não em entidades.

Verifica-se que está(ão) correta(s)

- a) I, II e III. b) II, apenas. c) III, apenas. d) II, III e IV, apenas. e) III e IV, apenas.

# Exercícios

3) [COPEVE-UFAL - 2012 - ALGÁS] Dadas as seguintes afirmações,

- I. Na arquitetura do JDBC, a diferença entre os tipos Statement e PreparedStatement é o fato do PreparedStatement manter os dados criptografados durante o tráfego entre o cliente e o servidor do SGBD.
- II. Um software que utiliza o framework Hibernate, necessariamente não utiliza JDBC.
- III. O framework Hibernate permite a implementação de transações na camada de persistência (aplicação), mesmo que o SGBD utilizado não implemente esse conceito.
- IV. O framework Hibernate é responsável por realizar o mapeamento objeto-relacional, mas possui a restrição de só aceitar chaves primárias compostas em relacionamentos, não em entidades.

Verifica-se que está(ão) correta(s)

- a) I, II e III. b) II, apenas. c) III, apenas. d) II, III e IV, apenas. **e) III e IV, apenas.**



# Principais Componentes do Hibernate

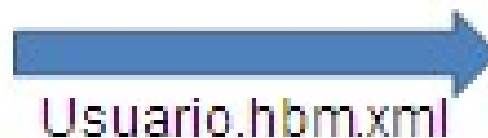
# Principais Componentes

**São 03 os Principais Componentes do Hibernate:**

- **1) Mapeamento das Tabelas em Classes**
  - **1.1) Com Arquivos .hbm.xml**
  - 1.2) Com Anotações
- 2) Arquivo de Configuração (hibernate.cfg.xml)
- 3) Classe de Gerenciamento (HibernateSessionFactory.java)

# 1) Mapeamento das Tabelas em Classes (com Arquivos)

Tabela <u>Usuario</u>
- <u>Uscod</u> : varchar2
- <u>Usnome</u> : varchar2
- <u>Ussenha</u> : varchar2
- <u>Usemail</u> : varchar2



Classe <u>Usuario</u>
- <u>Uscod</u>
- <u>Usnome</u>
- <u>Ussenha</u>
- <u>Usemail</u>

```
public class Usuario {  
    private String usCod;  
    private String usSenha;  
    private String usNome;  
    private String userEmail;  
}
```

# 1) Mapeamento das Tabelas em Classes (com Arquivos)

## USUARIO.HBM.XML

```
< ?xml version="1.0"?>
< !DOCTYPE hibernate-mapping PUBLIC "-//Hibernate Mapping DTD 3.0//EN"
"hibernate-mapping-3.0.dtd">

< hibernate-mapping>
  < class name="Usuario" table="tb_usuarios">
    < id name="UsCod" column="USCOD" type="string">
      < generator class="assigned"/>
    < /id>
    < property name="UsSenha" column="USSENHA" type="string"/>
    < property name="UsNome" column="USNOME" type="string"/>
    < property name="UsEmail" column="USEMAIL" type="string"/>
  < /class>
< /hibernate-mapping>
```

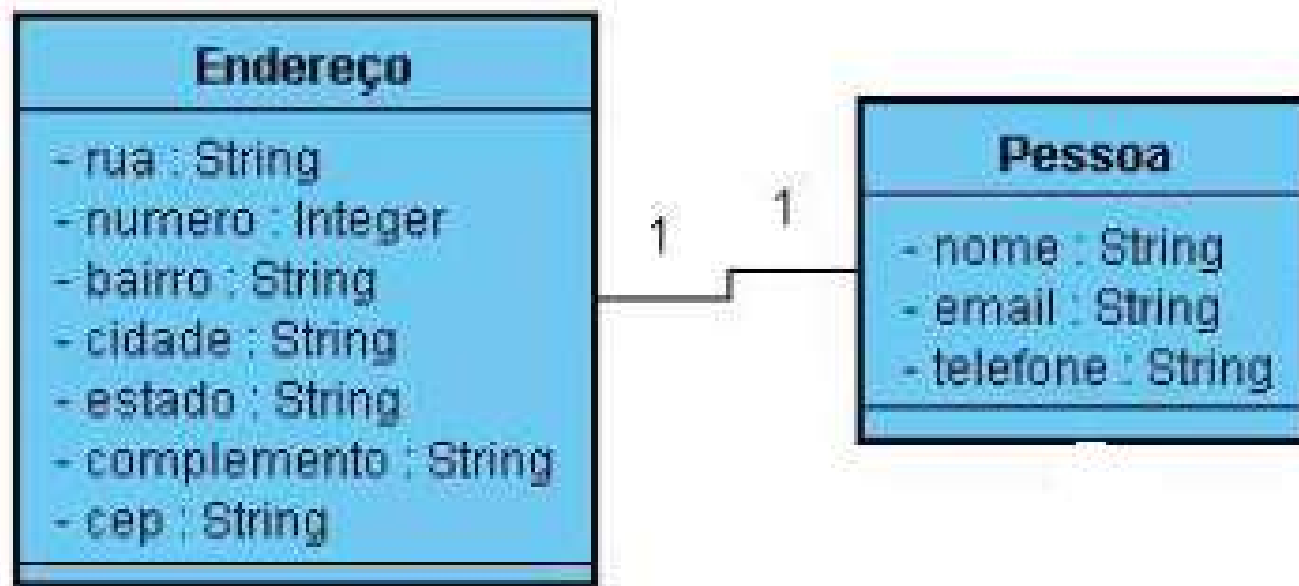
# 1) Mapeamento das Tabelas em Classes (com Arquivos)

## Principais Elementos

- **<class>** mapeia uma classe a uma tabela.
- **<id>** mapeia um atributo da classe à chave-primária da tabela.
- **<generator>** utilizado para gerar (automaticamente) os valores da chave-primária.
- **<property>** mapeia os demais atributos do objeto às colunas da tabela.
- Os arquivos de mapeamento (\*.hbm.xml) podem ser gerados automaticamente a partir da base de dados ou das classes Java.
- Esses arquivos de mapeamento (\*.hbm.xml) devem ser registrados no arquivo de configuração do Hibernate (**hibernate.cfg.xml**).

# 1) Mapeamento das Tabelas em Classes (com Arquivos)

## Mapeamento 1:1



# 1) Mapeamento das Tabelas em Classes (com Arquivos)

## PESSOA.HBM.XML

```
<hibernate-mapping>
  <class name="Pessoa">
    <id name="id">
      <generator class="increment"/>
    </id>
    <property name="nome"/>
    <property name="telefone"/>
    <property name="email"/>
    <one-to-one name="endereco"
      class="Endereco" cascade="save-update"/>
  </class>
</hibernate-mapping>
```

# 1) Mapeamento das Tabelas em Classes (com Arquivos)

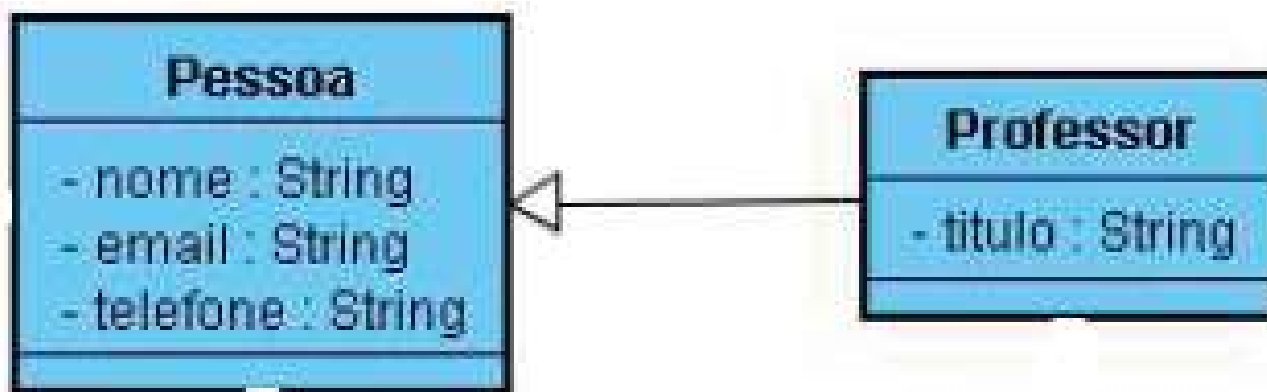
## ENDERECO.HBM.XML

```
<hibernate-mapping>
  <class name="Endereco">
    <id name="id" column="Pessoa_id">
      <generator class="foreign">
        <param name="property"> Pessoa </param>
      </generator> </id>
    <property name="bairro"/>
    <property name="cidade"/>
    <property name="complemento"/>
    <property name="estado"/>
    <property name="numero"/>
    <property name="rua"/>
    <property name="cep"/>
    <one-to-one name="Pessoa" class="Pessoa" constrained="true"/>
  </class></hibernate-mapping>
```



# 1) Mapeamento das Tabelas em Classes (com Arquivos)

## Herança



**O Hibernate suporta três estratégias de mapeamento de herança:**

- Uma tabela por hierarquia de classes
- Uma tabela por subclasse
- Uma tabela por classe concreta

# 1) Mapeamento das Tabelas em Classes (com Arquivos)

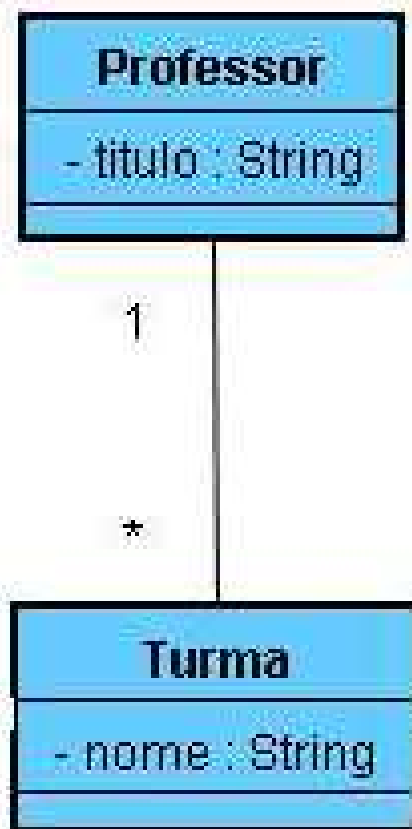
## PROFESSOR.HBM.XML

```
<hibernate-mapping>
  <joined-subclass name="Professor" extends="Pessoa">
    <key column="Pessoa_id"/>
    <property name="titulo"/>
    <set name="turmas" inverse="true">
      <key column="Pessoa_Professor_id"/>
      <one-to-many class="Turma"/>
    </set>
  </joined-subclass>
</hibernate-mapping>
```

- O elemento **<key>** indica a chave estrangeira na tabela Turma.

# 1) Mapeamento das Tabelas em Classes (com Arquivos)

## Mapeamento 1:N



- Coleções são mapeadas usando os elementos **<set>**, **<list>**, **<map>**, **<bag>**, **<array>** e **<primitive-array>**.
- Uma tabela para a coleção é requerida quando ela contém valores ou referências para outras entidades mapeadas em **1:N**.

# 1) Mapeamento das Tabelas em Classes (com Arquivos)

## TURMA.HBM.XML

```
<hibernate-mapping>
  <class name="Turma">
    <id name="id">
      <generator class="increment"/>
    </id>
    <property name="nome"/>
    <many-to-one name="professor" class="Professor"
      column="Professor_Pessoa_id"/>
  </class>
</hibernate-mapping>
```

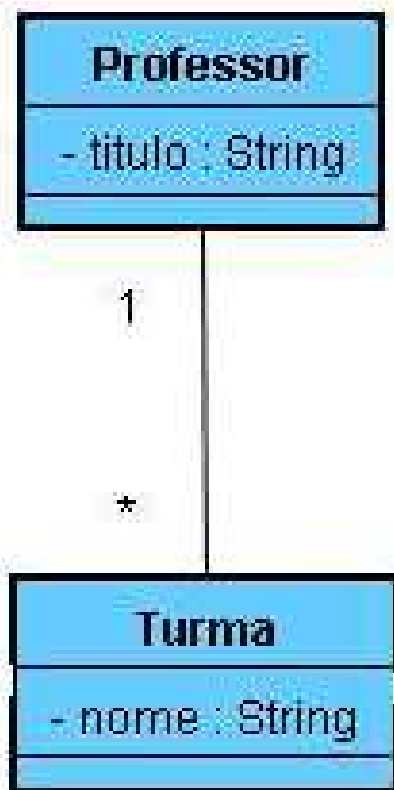
# Principais Componentes

**São 03 os Principais Componentes do Hibernate:**

- **1) Mapeamento das Tabelas em Classes**
  - 1.1) Com Arquivos .hbm.xml
  - **1.2) Com Anotações**
- 2) Arquivo de Configuração (hibernate.cfg.xml)
- 3) Classe de Gerenciamento (HibernateSessionFactory.java)

# 1) Mapeamento das Tabelas em Classes (com Anotações)

## Mapeamento 1:N



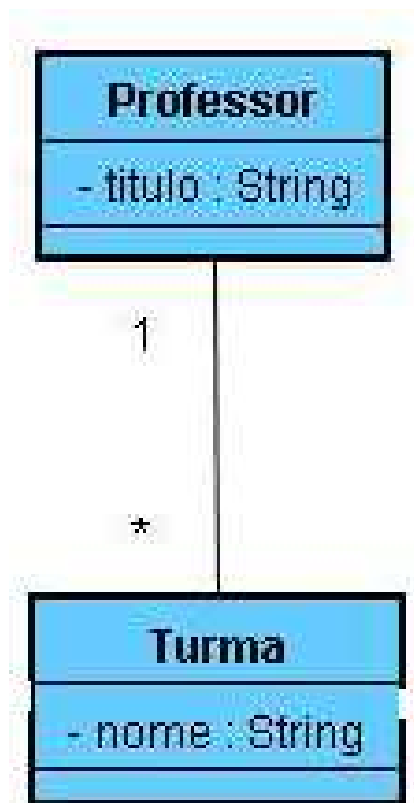
```
@Entity
@Table
public class Professor {

    @Id
    @GeneratedValue
    private Long id;
    private String name;

    @OneToMany(mappedBy="professor",
        cascade=CascadeType.PERSIST)
    private List<Turma> turmas =
        new ArrayList<Turma>();
    public Professor() {
        super();
    }
    ...
}
```

# 1) Mapeamento das Tabelas em Classes (com Anotações)

## Mapeamento 1:N



```
@Entity
@Table
public class Turma {
    @Id
    @GeneratedValue
    private Long id;

    private String name;

    @ManyToOne
    private Professor professor;

    public Turma() {}
    ...
}
```

# Exercícios

- 1) [FEPESE - 2013 - JUCESC] O hibernate pode utilizar o mapping document para mapear metadados. Que outra forma pode ser empregada para mapear metadados em hibernate?
  - a) JPA
  - b) Container
  - c) Comments
  - d) Webservice
  - e) Annotation
  
- 2) [CESPE - 2013 – INPI] No Hibernate, caso o nome da classe seja diferente do nome da tabela mapeada, é necessário informar, na anotação @Table, o nome da tabela, por meio do atributo name.
  - a) Certo
  - b) Errado



# Exercícios

- 1) [FEPESE - 2013 - JUCESC] O hibernate pode utilizar o mapping document para mapear metadados. Que outra forma pode ser empregada para mapear metadados em hibernate?
  - a) JPA
  - b) Container
  - c) Comments
  - d) Webservice
  - e) Annotation**
  
- 2) [CESPE - 2013 – INPI] No Hibernate, caso o nome da classe seja diferente do nome da tabela mapeada, é necessário informar, na anotação @Table, o nome da tabela, por meio do atributo name.
  - a) Certo**
  - b) Errado

# Principais Componentes

## São 03 os Principais Componentes do Hibernate:

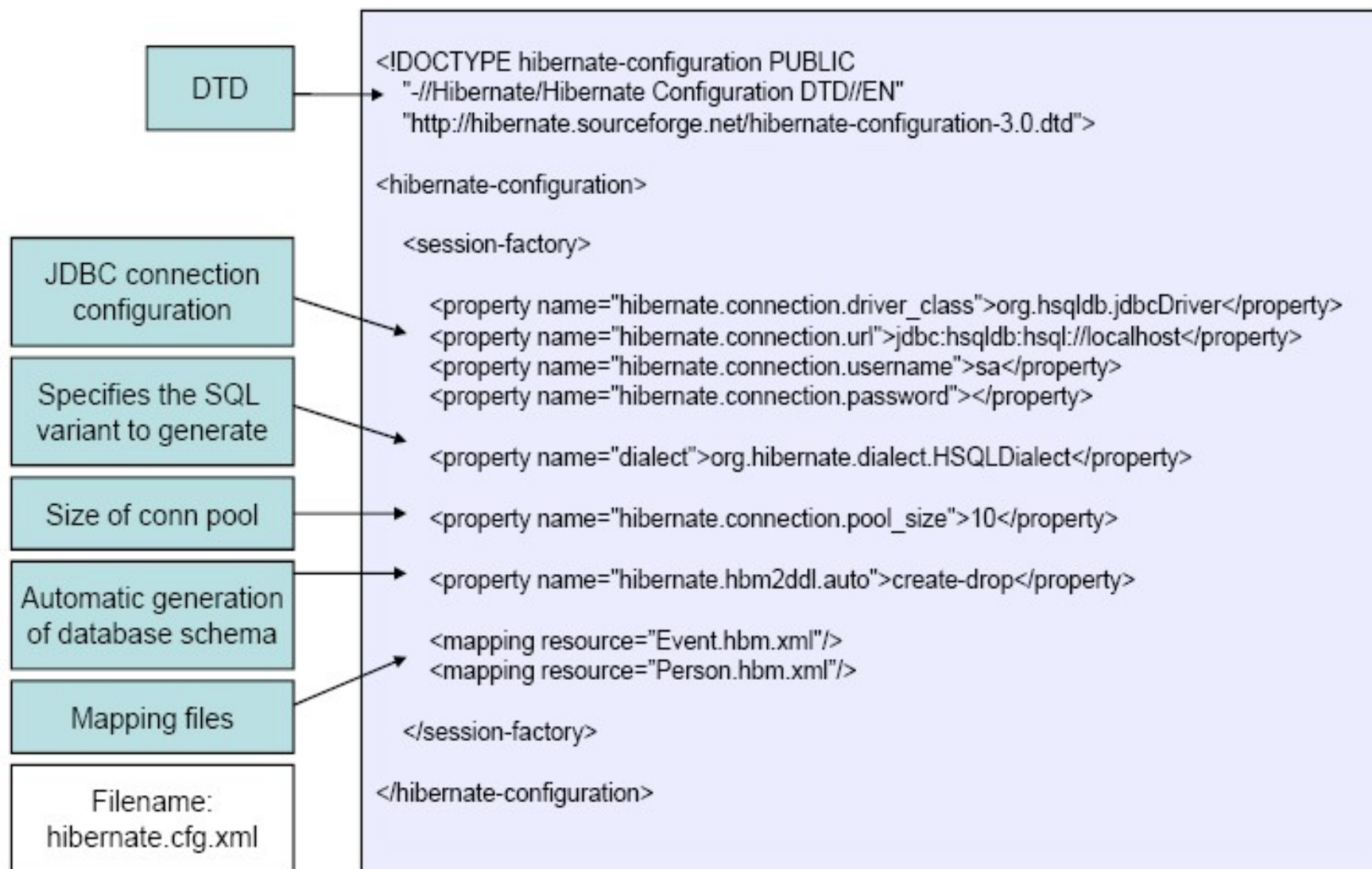
- 1) Mapeamento das Tabelas em Classes
  - 1.1) Com Arquivos .hbm.xml
  - 1.2) Com Anotações
- **2) Arquivo de Configuração (hibernate.cfg.xml)**
- 3) Classe de Gerenciamento (HibernateSessionFactory.java)

## 2) Arquivo de Configuração (hibernate.cfg.xml)

- Todas as configurações de inicialização do Hibernate, como propriedades do banco de dados, mapeamento das classes relacionadas, pool de conexão de banco de dados, etc., estão definidas no arquivo **hibernate.cfg.xml**.

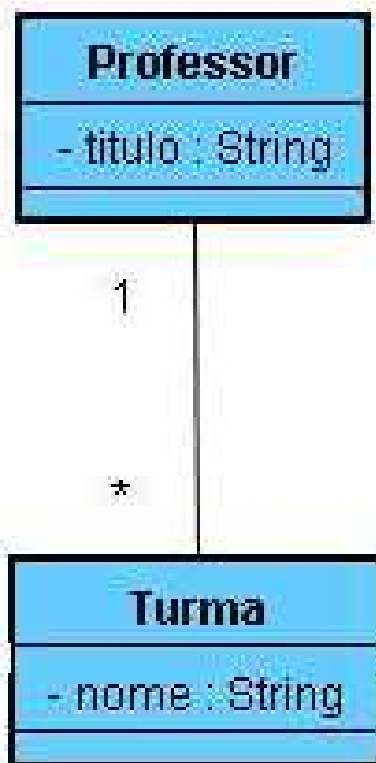
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
    </session-factory>
</hibernate-configuration>
```

## 2) Arquivo de Configuração (hibernate.cfg.xml) com Arquivos



## 2) Arquivo de Configuração (hibernate.cfg.xml) com Anotações

hibernate.cfg.xml



```
@Entity
<?xml version="1.0" encoding="utf-8"?>
<hibernate-configuration>
  <session-factory>
    ...
    <mapping class="com.abctreinamentos.Turma" />
    <mapping class="com.abctreinamentos.Professor" />
    <!--
    <mapping
      resource="com/abctreinamentos/Turma.hbm.xml">
    </mapping> -->
    <!--
    <mapping
      resource="com/abctreinamentos/Professor.hbm.xml">
    </mapping> -->
  </session-factory>
</hibernate-configuration>
```

# Exercício

- 1) [FCC - 2009 - TJ-PI] Hibernate é uma ferramenta open-source que facilita a comunicação entre aplicação Java e banco de dados, tais como manipulação de dados, gerenciamento de transações e pooling de conexões. A esse respeito, considere o trecho de código para configuração do Hibernate:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
  <class name="Pessoa" table="PESSOA">
    <id name="idPessoa" column="ID_PESSOA">
      <generator class="increment"/>
    </id>
    <property name="nome" column="NOME_PESSOA"/>
    <property name="email"/>
  </class>
</hibernate-mapping>
```

# Exercício

Tendo em vista os dados acima, analise:

- I. o código exibe a estrutura básica de um arquivo XML de mapeamento que indica ao Hibernate como carregar e gravar os objetos de uma classe de persistência;
- II. as declarações de mapeamento estão circunscritas entre as tags `</hibernate-mapping>`;
- III. o elemento `< id >` é a declaração de uma propriedade identificadora de tabela (chave-primária) e o atributo `column="ID_PESSOA"` indica qual é o campo na tabela ao qual ele está ligado.

É correto o que consta em

- a) I, apenas.
- b) I e II, apenas.
- c) I, II e III.
- d) II e III, apenas.
- e) I e III, apenas.

# Exercício

Tendo em vista os dados acima, analise:

- I. o código exibe a estrutura básica de um arquivo XML de mapeamento que indica ao Hibernate como carregar e gravar os objetos de uma classe de persistência;
- II. as declarações de mapeamento estão circunscritas entre as tags `</hibernate-mapping>`;
- III. o elemento `<id>` é a declaração de uma propriedade identificadora de tabela (chave-primária) e o atributo `column="ID_PESSOA"` indica qual é o campo na tabela ao qual ele está ligado.

É correto o que consta em

- a) I, apenas.
- b) I e II, apenas.
- c) I, II e III.**
- d) II e III, apenas.
- e) I e III, apenas.



# Principais Componentes

## São 03 os Principais Componentes do Hibernate:

- 1) Mapeamento das Tabelas em Classes
  - 1.1) Com Arquivos .hbm.xml
  - 1.2) Com Anotações
- 2) Arquivo de Configuração (hibernate.cfg.xml)
- **3) Classe de Gerenciamento (HibernateSessionFactory.java)**

# Classe de Gerenciamento (SessionFactory.java)

- É a classe responsável por abrir novas sessões Hibernate a partir de uma conexão JDBC fornecida pelo usuário.
- Cada SessionFactory deve ser configurado a partir de um único arquivo de configuração **hibernate.cfg.xml**.
- Um objeto **SessionFactory** é construído a partir do objeto **Configuration**. A partir de sua construção não é mais necessária a utilização do objeto Configuration.

# SessionFactory.java

## Características

- Tem como principal propósito fornecer instâncias de **Session**.
- É baseado no Padrão de Projeto *Factory*.
- É compartilhado pelas Threads da aplicação.
- Criação: `SessionFactory.currentSession();`

```
public class SessionFactory {  
  
    private static String CONFIG_FILE_LOCATION = "/hibernate.cfg.xml";  
    private static final ThreadLocal<Session> threadLocal =  
        new ThreadLocal<Session>();  
  
    ...  
    public static void rebuildSessionFactory() {...}  
    public static void closeSession() throws HibernateException {...}  
  
    public static Session currentSession() throws HibernateException  
    {...}  
  
}
```

# Session.java

## Características

- É obtido de uma instância de SessionFactory;
- Principal interface entre a Aplicação Java e o Hibernate. Responsável por armazenar e recuperar objetos.

- Criação: `Session session = sessionFactory.getCurrentSession();`

- Principais Métodos:

- Inclusão `session.save(transientInstance);`

- `session.persist(transientInstance);`

- Alteração

- `session.merge(transientInstance);`

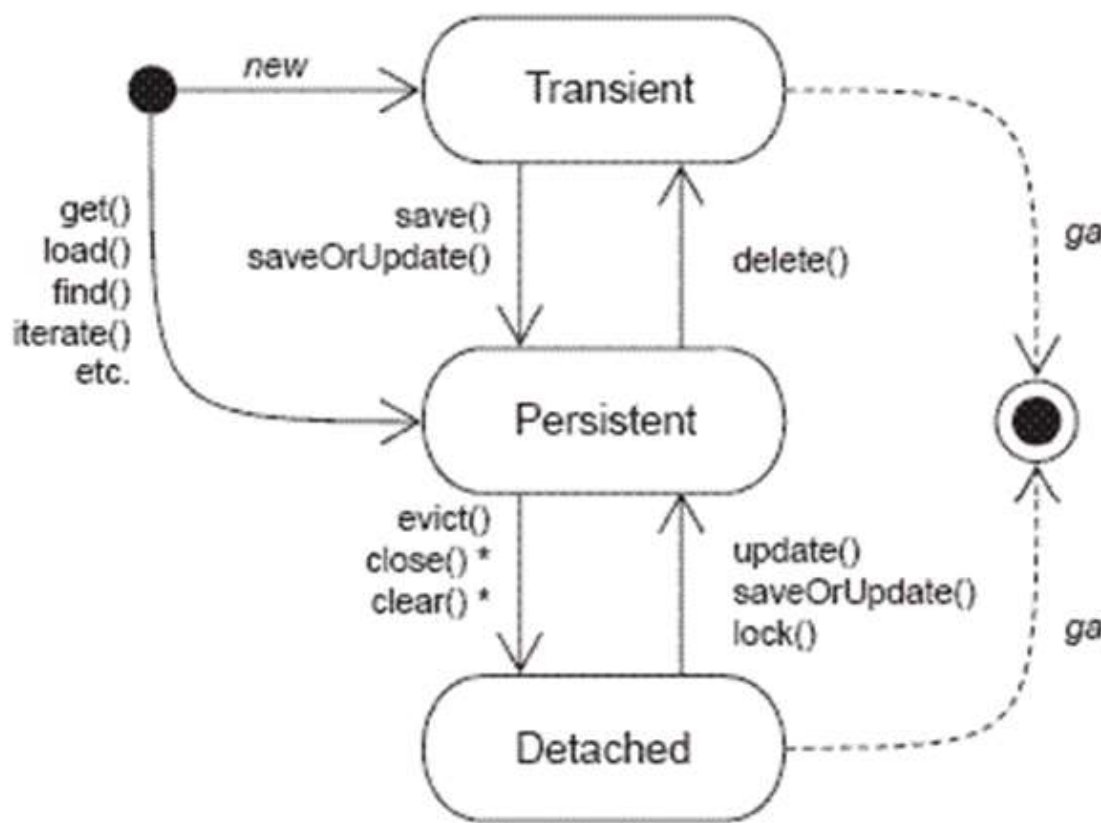
- Consulta

- `session.findAll(transientInstance);`

- Exclusão

- `session.delete(persistentInstance);`

# Máquina de Estados



\* affects all instances in a Session

## Persistente (*persistent*)

- São objetos que estão associados aos registros de um banco de dados;

## Transiente (*transient*)

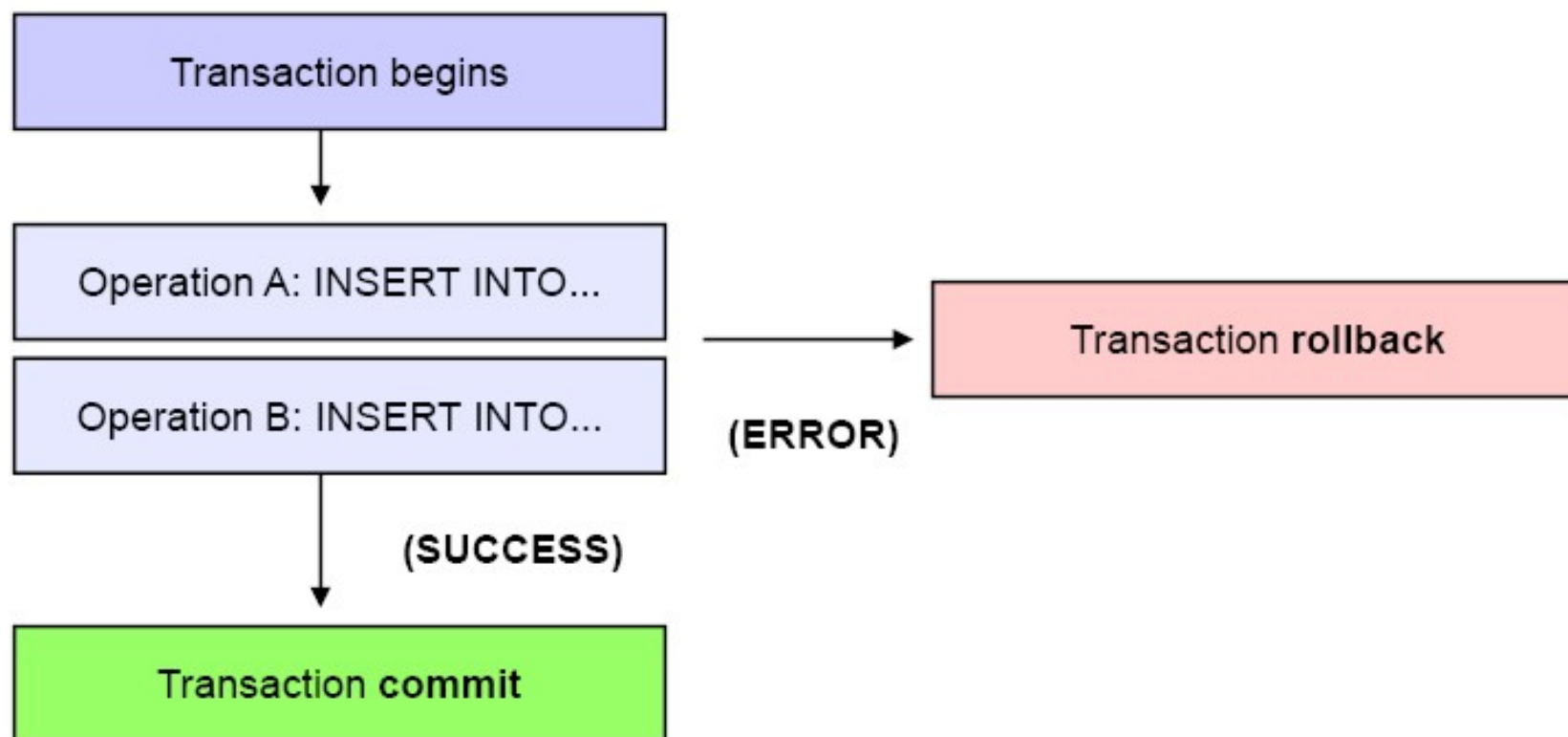
- São objetos que não estão associados aos registros de um banco de dados;

## Desanexado (*detached*)

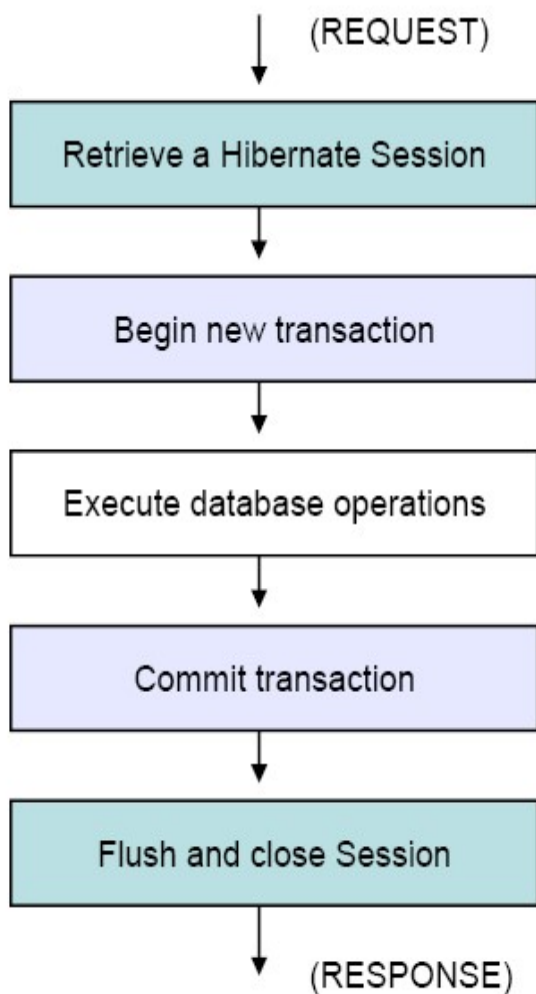
- São objetos que existem no bando de dados, porém não existem na sessão hibernate;

# Transações no Hibernate

- São um conjunto de operações de persistência no Banco de Dados que devem ser executadas por completo.
- Todas as instruções devem ser finalizadas com um comando *commit* ou canceladas com um comando *rollback*.



# Transações no Hibernate



```
Session session = sessionFactory.openSession();

Transaction transaction = null;

try
{
    transaction = session.beginTransaction();

    session.save( event );
    session.save( person );

    transaction.commit();
}
catch ( RuntimeException ex )
{
    if ( transaction != null )
    {
        transaction.rollback();
        throw ex;
    }
}
finally
{
    session.close();
}
```

# Exercícios

- 1) [CESPE - 2009 - CEHAP-PB] No framework Hibernate, é comum que uma instância de uma classe persistente tenha três estados específicos. Assinale a opção que contém esses três estados.
  - a) plugged, disconnected, timewait
  - b) connected, disconnected, detached
  - c) transient, persistent, detached
  - d) transient, connected, timewait
  
- 2) [CESGRANRIO - 2008 - TJ-RO] Sobre o framework de mapeamento objeto/relacional Hibernate, é correto afirmar que
  - a) após modificar um objeto que já está vinculado a uma session (sessão) aberta, é necessário utilizar o método update() do objeto session para que as modificações sejam gravadas no banco de dados.
  - b) um objeto se encontra no estado detached quando tiver sido criado com o construtor new, estando ainda desvinculado de qualquer session (sessão) do Hibernate.



# Exercícios

- c) um objeto gerenciado pelo Hibernate se encontra no estado transient quando tiver sido criado vinculado a uma session (sessão) do Hibernate, mas ainda não tiver sido gravado no banco de dados.
- d) um objeto gerenciado pelo Hibernate se encontra no estado persistent quando a session (sessão) a que o mesmo estava vinculado foi fechada e, em consequência, o objeto já foi gravado no banco de dados.
- e) uma session do Hibernate não é thread-safe, o que significa que pode gerar resultados imprevisíveis e instabilidade se uma instância for utilizada, simultaneamente, por mais de uma thread da aplicação sem a devida sincronização do acesso à mesma.

# Exercícios

- 1) [CESPE - 2009 - CEHAP-PB] No framework Hibernate, é comum que uma instância de uma classe persistente tenha três estados específicos. Assinale a opção que contém esses três estados.
  - a) plugged, disconnected, timewait
  - b) connected, disconnected, detached
  - c) transient, persistent, detached**
  - d) transient, connected, timewait
  
- 2) [CESGRANRIO - 2008 - TJ-RO] Sobre o framework de mapeamento objeto/relacional Hibernate, é correto afirmar que
  - e) uma session do Hibernate não é thread-safe, o que significa que pode gerar resultados imprevisíveis e instabilidade se uma instância for utilizada, simultaneamente, por mais de uma thread da aplicação sem a devida sincronização do acesso à mesma.**

# Projeto Prático com o Hibernate

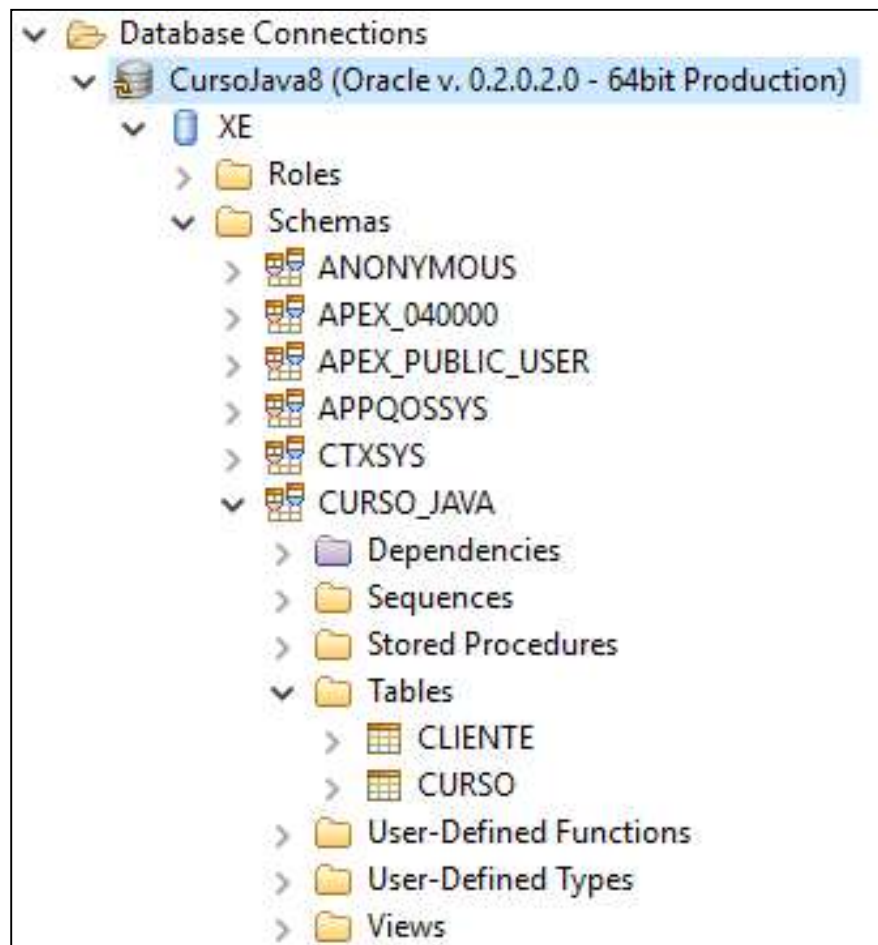
# Projeto Prático com o Hibernate

## São necessários realizar 06 Passos:

- 1. Configurar o acesso ao banco de dados no Eclipse
- 2. Fazer o download do Hibernate e copiar os seus arquivos .jar para um projeto Java
- 3. Instalar o Jboss Tools para poder utilizar o Hibernate no Eclipse
- 4. Configurar no projeto Java os arquivos de configuração do Hibernate
- 5. Realizar a Engenharia Reversa das Tabelas **Cliente, Curso e Pagamento**
- 6. Fazer os Ajustes necessários nos Códigos Gerados

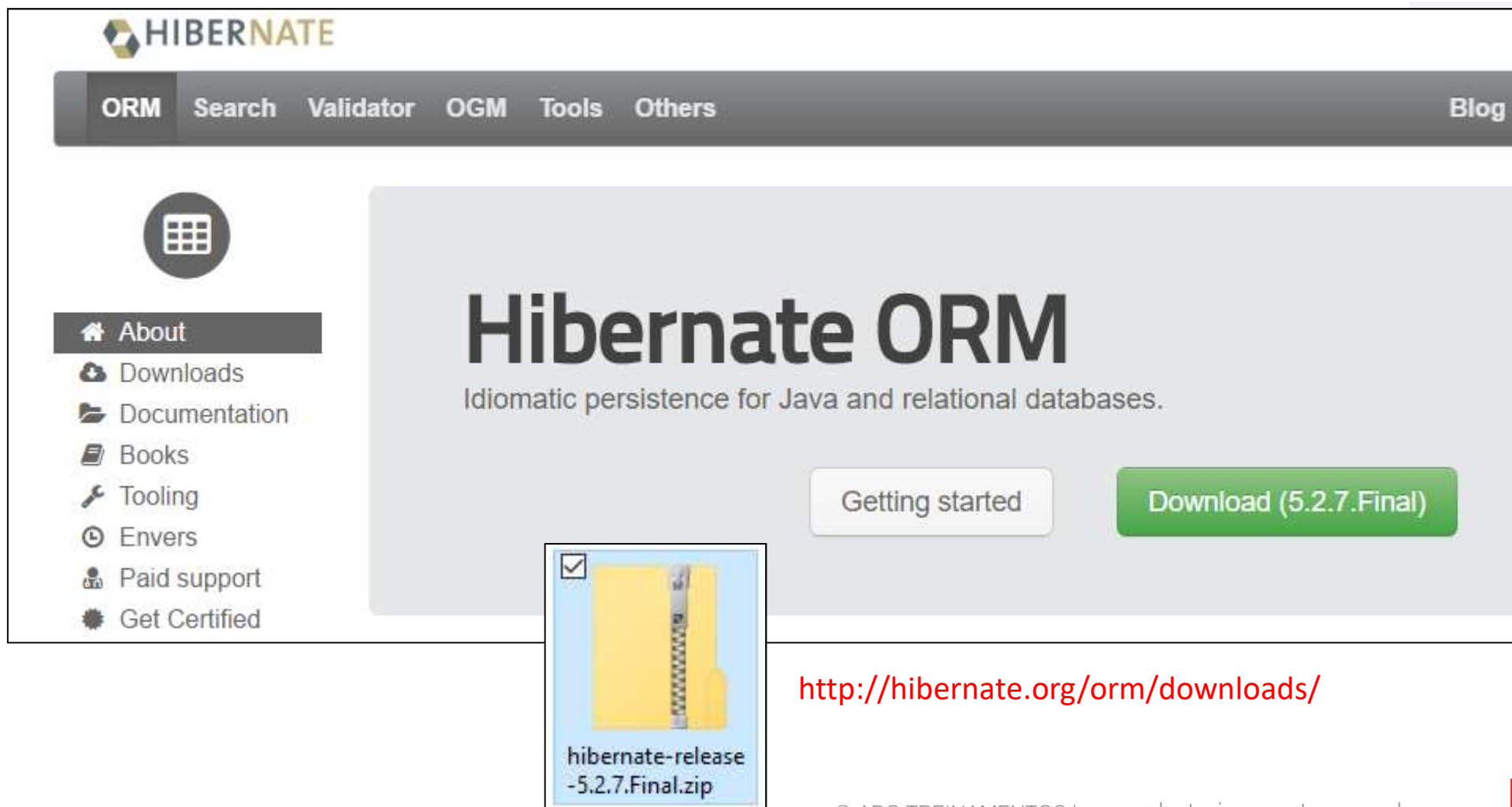
# Projeto Prático com o Hibernate

- **1. Configurar o acesso ao banco de dados no Eclipse**



# Projeto Prático com o Hibernate

- **2. Fazer o download do Hibernate e copiar os seus arquivos .jar para um projeto Java**



The screenshot shows the Hibernate ORM website. The main heading is "Hibernate ORM" with the tagline "Idiomatic persistence for Java and relational databases." Below this, there are two buttons: "Getting started" and "Download (5.2.7.Final)". A yellow folder icon with a checkmark is shown, labeled "hibernate-release-5.2.7.Final.zip".

<http://hibernate.org/orm/downloads/>

# Projeto Prático com o Hibernate

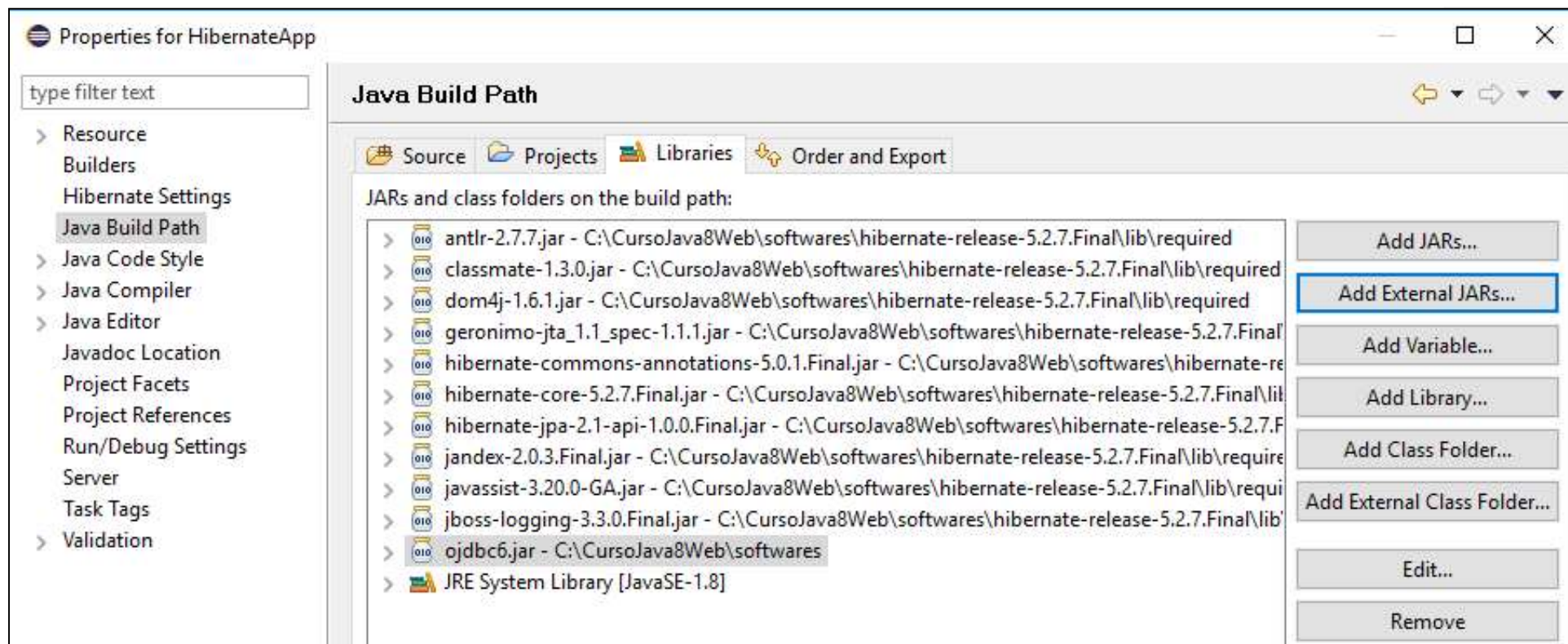
- **2. Fazer o download do Hibernate e copiar os seus arquivos .jar para um projeto Java**
- Criar um novo projeto Java (Hibernate) e adicionar as principais bibliotecas do Hibernate nesse projeto:
  - ...\\hibernate-release-5.2.7.Final\\lib\\required
- Adicionar também o driver Oracle (**ojdbc6.jar**).
- **Fazer o download do Apache Commons Logging**
- Copiar o arquivo (**commons-logging-1.2.jar**) para o projeto Java (HibernateApp).





# Projeto Prático com o Hibernate

- **2. Fazer o download do Hibernate e copiar os seus arquivos .jar para um projeto Java**

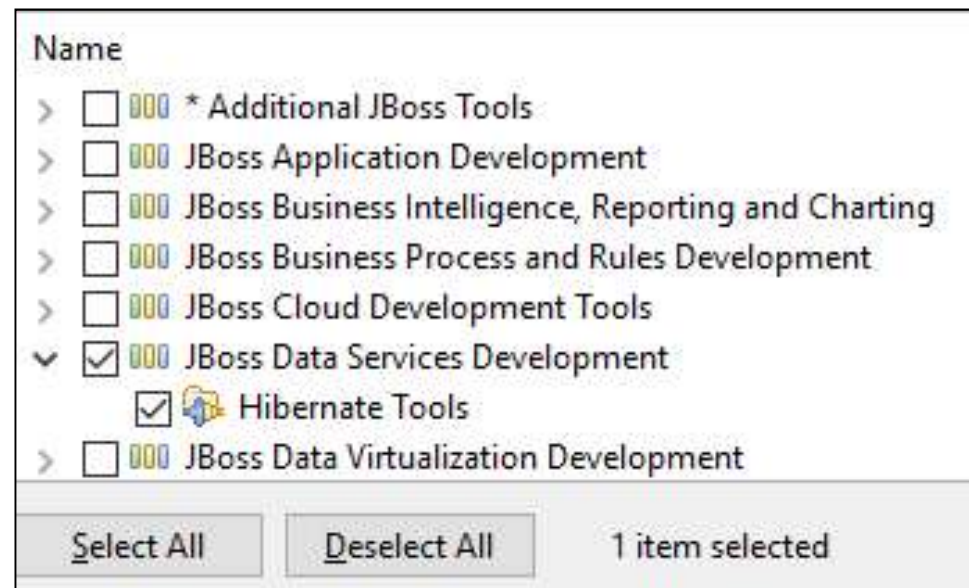




# Projeto Prático com o Hibernate

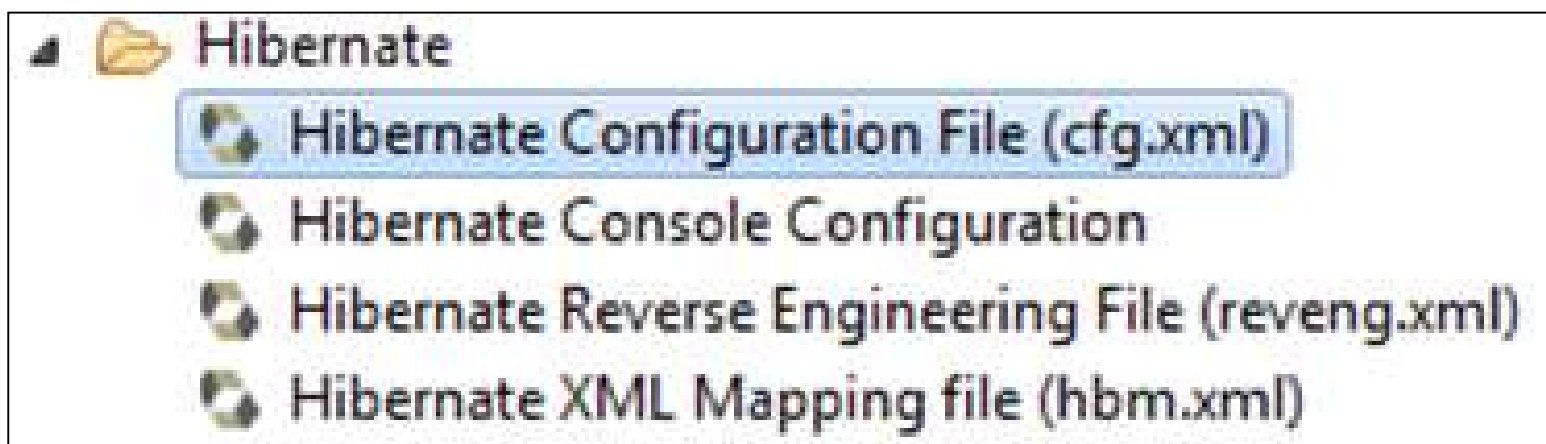
- **3. Instalar o Jboss Tools para poder utilizar o Hibernate no Eclipse**
- Necessário instalar o Jboss Tools para poder utilizar o Hibernate no Eclipse.
- No Eclipse, selecionar o Menu “Help” e depois a opção “Install New Software”. Informar o endereço abaixo:

<http://download.jboss.org/jbosstools/updates/stable/luna/>



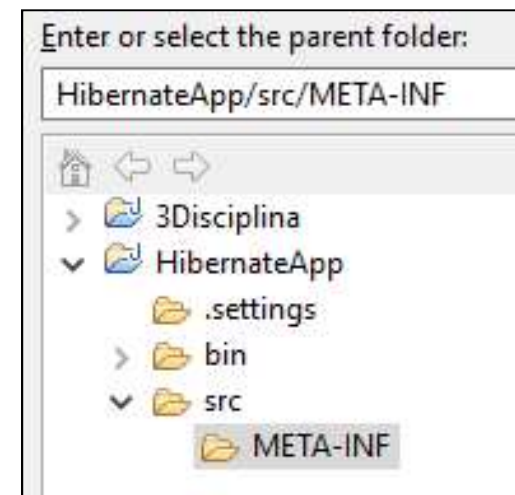
# Projeto Prático com o Hibernate

- **4. Configurar no projeto Java os arquivos de configuração do Hibernate**
- São 04 os arquivos de configuração necessários para criar um projeto Java no Eclipse com suporte ao Hibernate.



# Projeto Prático com o Hibernate

- 4. Configurar no projeto Java os arquivos de configuração do Hibernate
  - 4.1. Hibernate Configuration File (cfg.xml)
  - Criar a pasta 'META-INF' em 'src'. Em seguida, criar o arquivo de configuração (hibernate.cfg.xml) nesta pasta.
  - Clicar com o botão direito na pasta 'META-INF' e escolha “New” ⇒ “Other...” ⇒ “Hibernate” ⇒ “Hibernate Configuration File (cfg.xml)”
  - Pressionar o botão “Next >”.
  - Pressionar novamente o botão “Next >”.



# Projeto Prático com o Hibernate

- 4. Configurar no projeto Java os arquivos de configuração do Hibernate
- 4.1. Hibernate Configuration File (cfg.xml)
- Selecionar a opção “**Get values from Connection**” e preencher os campos restantes.



**Hibernate Configuration File (cfg.xml)**  
This wizard creates a new configuration file to use with Hibernate.

Container: /HibernateApp/src/META-INF  
File name: hibernate.cfg.xml  
Session factory name: sessionFactory

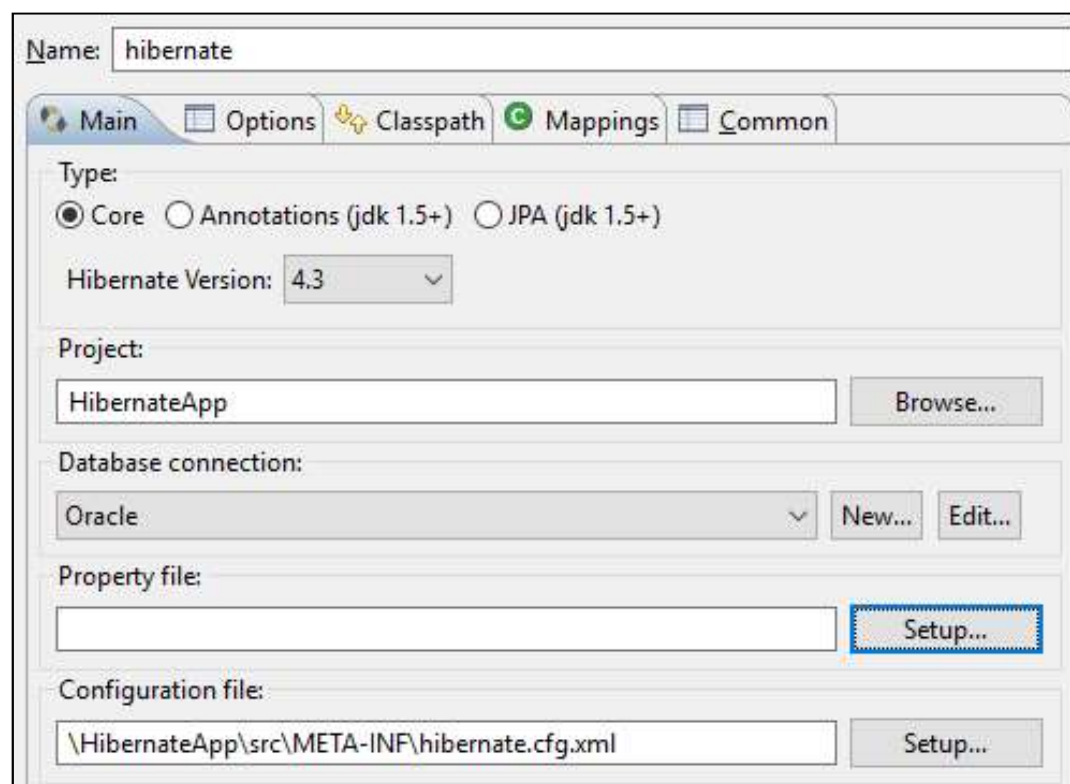
[Get values from Connection](#)

Database dialect: Oracle 10g  
Driver class: oracle.jdbc.OracleDriver  
Connection URL: jdbc:oracle:thin:@localhost:1521:XE  
Default Schema: CURSO\_JAVA  
Default Catalog:  
Username: curso\_java  
Password: schema

☐ Create a console configuration

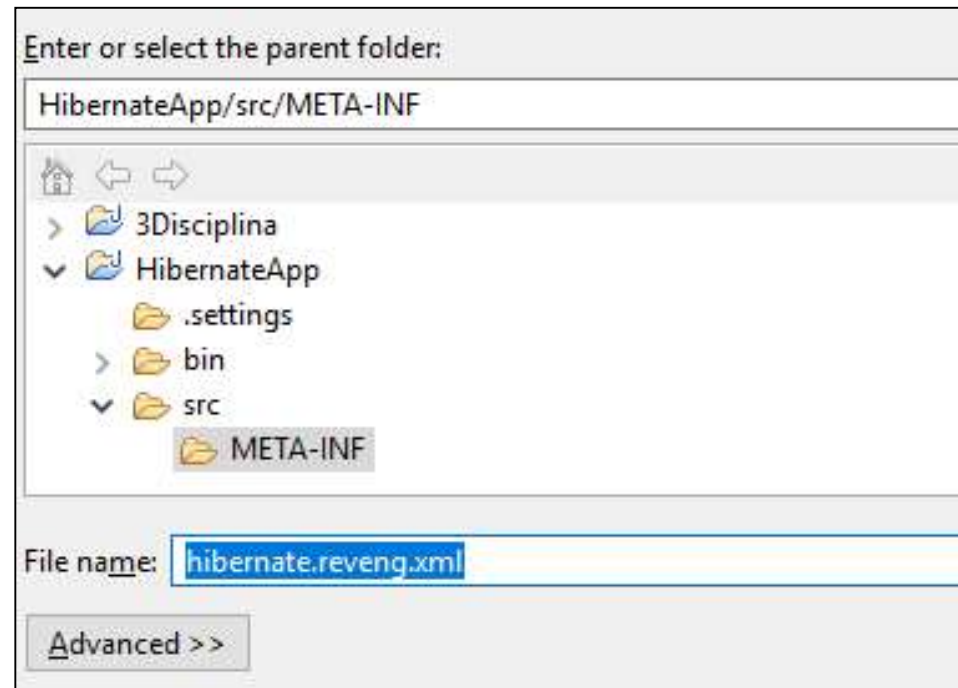
# Projeto Prático com o Hibernate

- 4. Configurar no projeto Java os arquivos de configuração do Hibernate
- 4.2. Hibernate Console Configuration
- Clicar com o botão direito na pasta 'META-INF' e escolha **"New"** ⇒ **"Other..."** ⇒ **"Hibernate"** ⇒ **"Hibernate Console Configuration"**
- Pressionar o botão **"Next >"**.
- Pressionar o botão **"Finish"**.



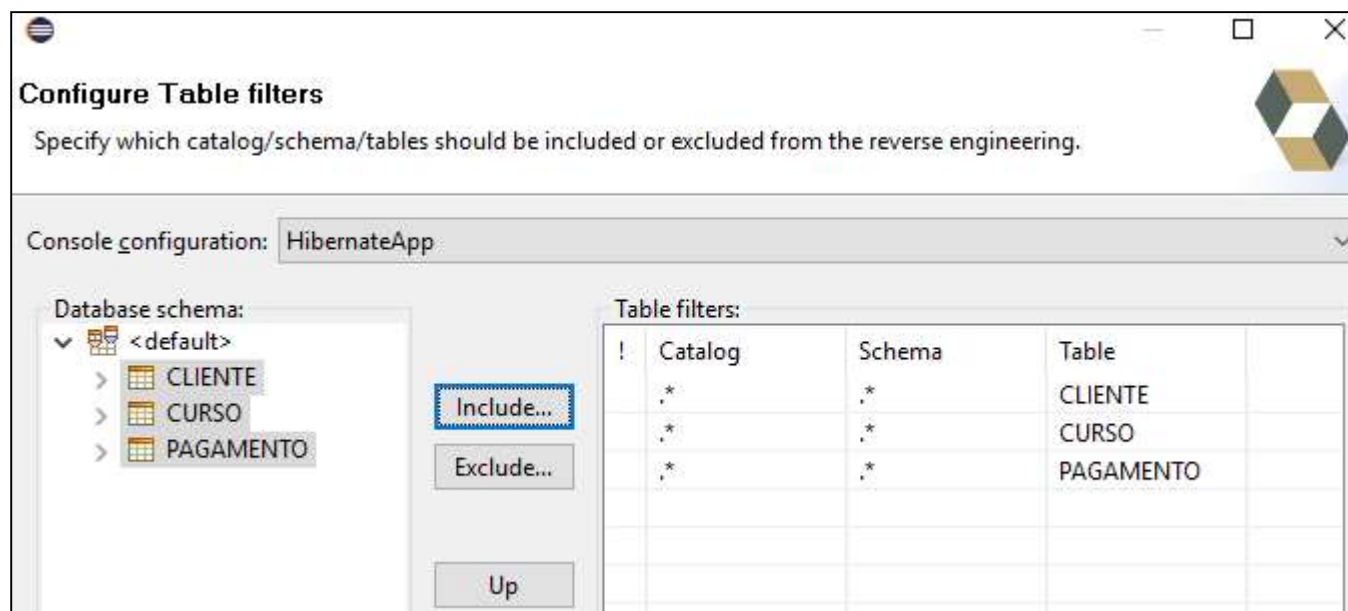
# Projeto Prático com o Hibernate

- 4. Configurar no projeto Java os arquivos de configuração do Hibernate
- 4.3. Hibernate Reverse Engineering File (revenge.xml)
- Clicar com o botão direito na pasta 'META-INF' e escolha “New” ⇒ “Other...” ⇒ “Hibernate” ⇒ “Hibernate Reverse Engineering File (revenge.xml)”
- Pressionar o botão “Next >”.
- Pressionar novamente o botão “Next >”.



# Projeto Prático com o Hibernate

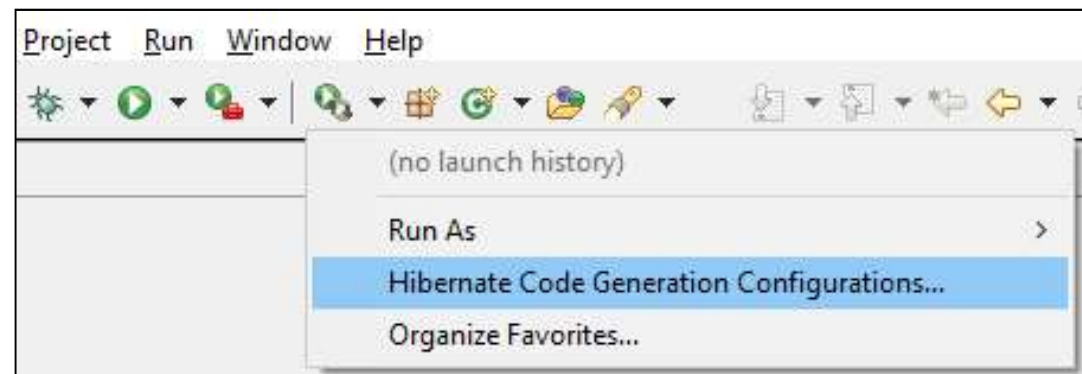
- 4. Configurar no projeto Java os arquivos de configuração do Hibernate
- **4.3. Hibernate Reverse Engineering File (revenge.xml)**
- Selecionar as Tabelas **Cliente**, **Curso** e **Pagamento**.
- Pressionar o botão **"Include >"**.



- Pressionar o botão **"Finish"**.

# Projeto Prático com o Hibernate

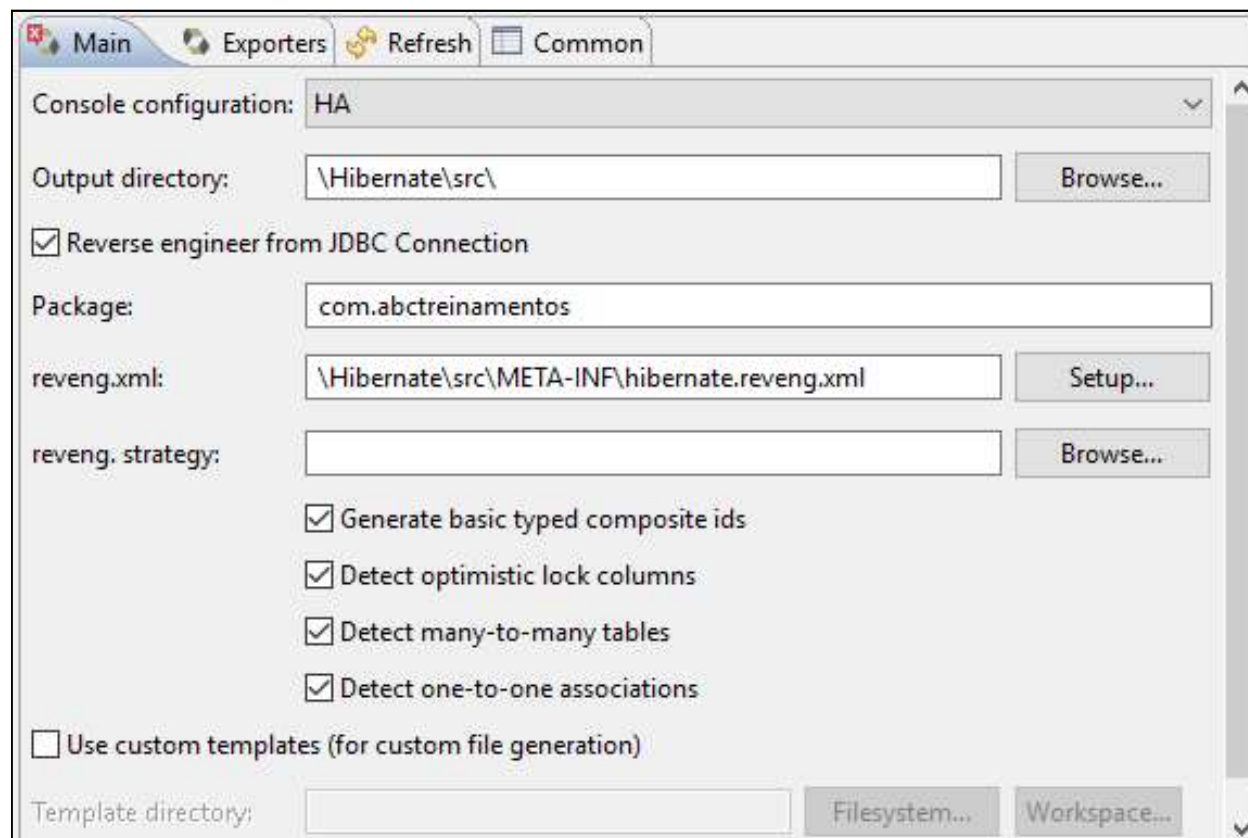
- **5. Realizar a Engenharia Reversa das Tabelas Cliente, Curso e Pagamento**
- Criar o pacote **com.abctreinamentos**.
- Selecionar no menu a opção “**Hibernate Code Generation Configurations** ... >”.





# Projeto Prático com o Hibernate

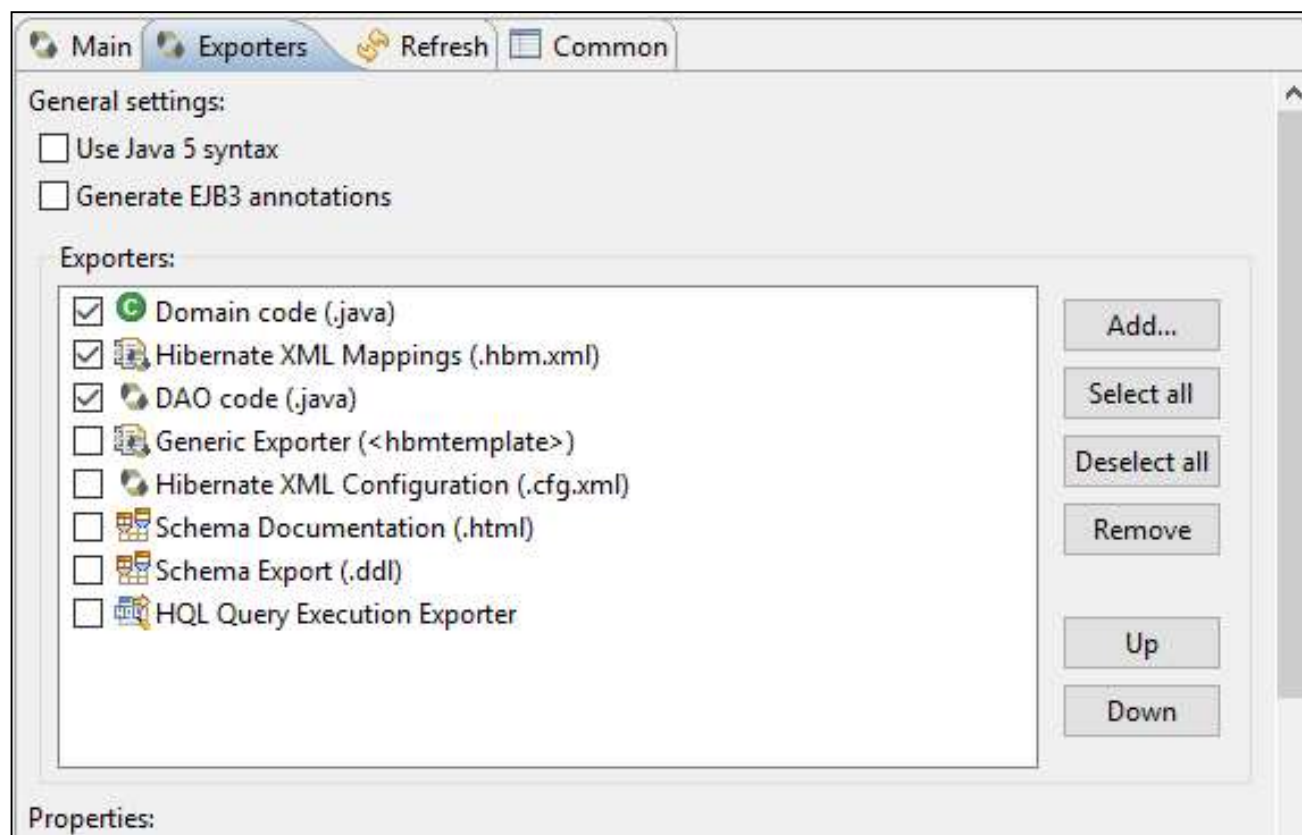
- 5. Realizar a Engenharia Reversa das Tabelas Cliente e Curso



- Pressionar o botão **Finish**.

# Projeto Prático com o Hibernate

- 5. Realizar a Engenharia Reversa das Tabelas Cliente e Curso

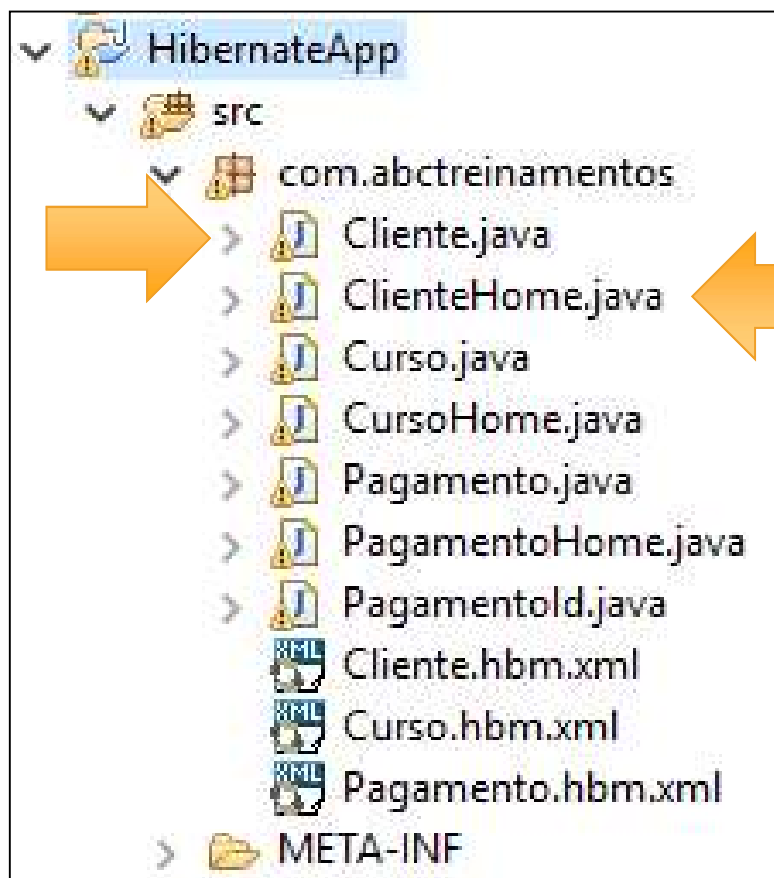


- Pressionar o botão “**Finish**”.

# Projeto Prático com o Hibernate

- 5. Realizar a Engenharia Reversa das Tabelas Cliente e Curso

VO



DAO

# Projeto Prático com o Hibernate

- **6. Fazer os Ajustes necessários nos Códigos Gerados**

- A) Incluir os códigos abaixo no arquivo **hibernate.cfg.xml**

```
<property name="hibernate.current_session_context_class">thread</property>
<mapping resource="com/abctreinamentos/Cliente.hbm.xml"/>
<mapping resource="com/abctreinamentos/Curso.hbm.xml"/>
<mapping resource="com/abctreinamentos/Pagamento.hbm.xml"/>
```

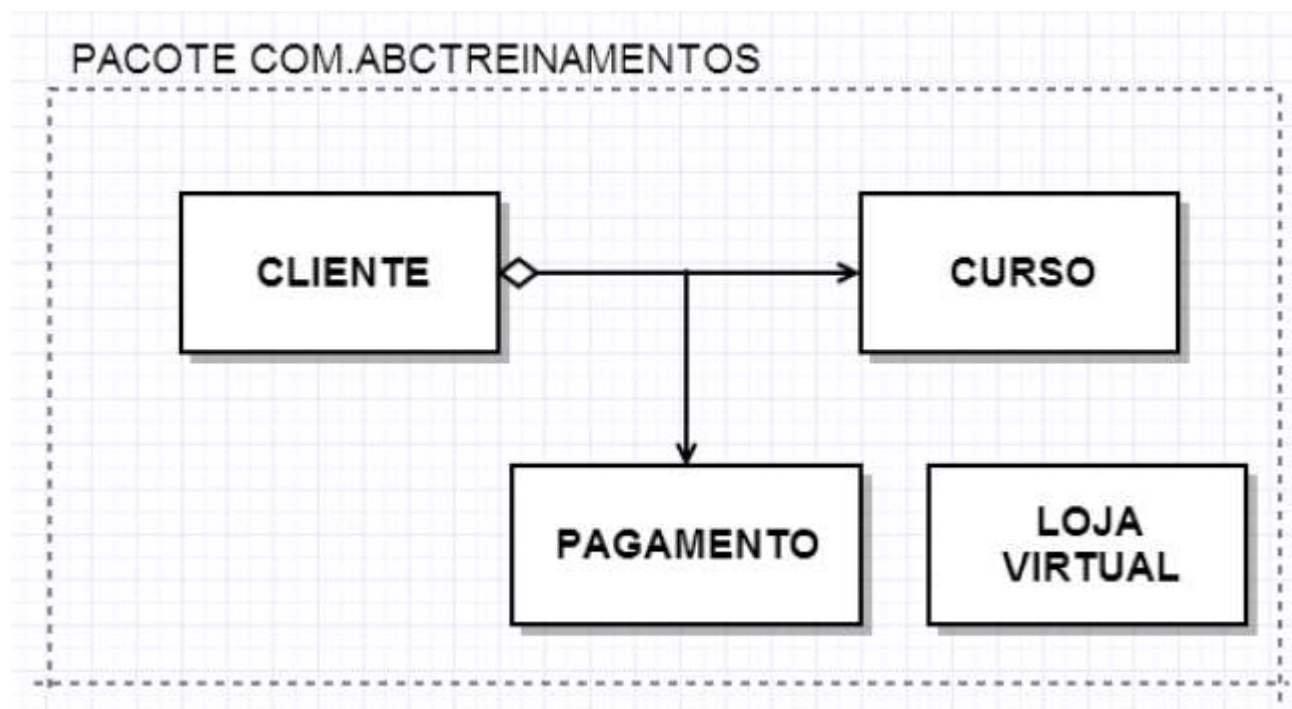
- B) Trocar **BigDecimal** por **long** nas **Classes e Arquivos XML**

- C) Alterar o método **getSessionFactory()** nas classes **DAO (Home)**

```
protected SessionFactory getSessionFactory() {
    SessionFactory sessionFactory = new Configuration().
        configure(new File("src/META-INF/hibernate.cfg.xml"))
        .buildSessionFactory();
    return sessionFactory;
}
```

# Exercícios

- 1) Criar as classes **ClienteApp**, **CursoApp** e **PagamentoApp** para realizar as operações de **CRUD** nas suas respectivas Tabelas.
- 2) **ATIVIDADE EXTRA:** Copiar o arquivo **LojaVirtual** escrito na **Unidade 3** e adaptá-lo às operações de CRUD realizadas no exercício (1).



# Tipos de Consultas no Hibernate

# Tipos de Consultas no Hibernate

## **São três os tipos de consultas no Hibernate:**

- Hibernate Query Language (HQL);
  - Criteria Query API;
  - SQL “puro”.
- 
- A maioria das consultas são resolvidas com HQL e Criteria. As específicas são resolvidas com o SQL “puro”.

# Hibernate Query Language

- O HQL é uma linguagem parecida com o SQL, porém, "orientada a objetos".
- Possibilita descrever consultas polimórficas e consultas sobre coleções.

```
from Pessoa pessoa where upper(pessoa.nome) like 'MARIA%'
```

- A consulta acima retorna todos os objetos da classe pessoa e de suas subclasses que tenham o nome iniciado por 'MARIA'.

```
public List find(String nome)
{
    String sql = "from Contatos contatos where contatos.nome like :nome";
    Query q = getSession().createQuery(sql);
    q.setParameter("nome", nome);
    List contatos = q.list();
    return contatos;
}
```

- A consulta acima retorna todos os objetos da classe Contatos e de suas subclasses que tenham o nome informado via parâmetro.



# Criteria

- É uma API (bastante intuitiva) utilizada para executar consultas.
- Para criar uma nova consulta, basta criar um objeto Criteria e definir os critérios da consulta desejada.

```
Criteria consulta = session.createCriteria(Contactos.class);  
consulta.add( Expression.like("nome", "Maria%") );  
List resultado = consulta.list();
```

- A consulta acima vai retornar todas as 'Marias'.



- Outro exemplo de consulta com a definição de 'filtros':

```
Criteria criteria = session.createCriteria( Event.class );  
  
criteria.add( Restrictions.eq( "title", "Rolling Stones" ) );  
criteria.add( Restrictions.gt( "date", new Date() ) );  
  
criteria.setMaxResults( 10 );  
  
List events = criteria.list();
```

# Exercícios

- 1) [CESPE - 2010 - TRE-BA] No Hibernate, apenas a linguagem de consulta HQL (hibernate query language) pode ser utilizada. A HQL executa os pedidos SQL sobre as classes de persistência do Java em vez de tabelas no banco de dados, o que diminui a distância entre o desenvolvimento das regras de negócio e o banco de dados.  
a) Certo    b) Errado
- 2) [CESPE - 2010 - TCU] A tecnologia Hibernate 3.5 é apropriada para o sistema a ser desenvolvido: entre as características que a credenciam, está o fato de ela possibilitar a recuperação de objetos por meio da formulação de queries em linguagens HQL (hibernate query language) e SQL (structured query language), bem como pelo uso de APIs (application programming interfaces) de busca por critério, entre outras.  
a) Certo    b) Errado
- 3) Incluir o método **consultarTodos** nas classes **ClienteApp**, **CursoApp** e **PagamentoApp**.

# Exercícios

- 1) [CESPE - 2010 - TRE-BA] No Hibernate, apenas a linguagem de consulta HQL (hibernate query language) pode ser utilizada. A HQL executa os pedidos SQL sobre as classes de persistência do Java em vez de tabelas no banco de dados, o que diminui a distância entre o desenvolvimento das regras de negócio e o banco de dados.  
a) Certo    **b) Errado**
- 2) [CESPE - 2010 - TCU] A tecnologia Hibernate 3.5 é apropriada para o sistema a ser desenvolvido: entre as características que a credenciam, está o fato de ela possibilitar a recuperação de objetos por meio da formulação de queries em linguagens HQL (hibernate query language) e SQL (structured query language), bem como pelo uso de APIs (application programming interfaces) de busca por critério, entre outras.  
a) **Certo**    b) Errado
- 3) Incluir o método **consultarTodos** nas classes **ClienteApp**, **CursoApp** e **PagamentoApp**.

# RESUMO

# TÓPICOS APRESENTADOS

- Neste conjunto de videoaulas nós vimos:
  - **Introdução ao Hibernate**
  - **Principais Componentes do Hibernate**
  - **Projeto Prático com o Hibernate**
  - **Tipos de Consultas no Hibernate**

# ATIVIDADES PARA SE APROFUNDAR

- 1) Implementar as operações de **Novo, Consultar, Alterar e Excluir** na aplicação **LojaVirtual**.

