
Modulo Java

Guía de Ejercicios JPA

JPA – HIBERNATE	3
<i>Ejercicio I.....</i>	<i>3</i>
<i>Ejercicio II.....</i>	<i>3</i>
<i>Ejercicio III.....</i>	<i>3</i>
<i>Ejercicio IV.....</i>	<i>4</i>
<i>Ejercicio V.....</i>	<i>4</i>

JPA – Hibernate

Ejercicio I

1. Crear un proyecto MAVEN, llamarlo Guia9
2. Asociarle un groupId com.capgemini.guia9

Ejercicio II

3. Agregar las siguientes dependencias al proyecto
 - mySql
 - hibernate
 - junit
4. Dentro de la carpeta src crear la carpeta META-INF (respetar las mayusculas)
5. Crear el archivo persistence.xml dentro de la carpeta META-INF con el contenido de conexión a BD correspondiente
6. Modificar los datos de conexión **driver**, **url**, **user** y **password** según su base de datos

Ejercicio III

7. Crear la clase Employee, con las annotations que correspondan (ver slides)

Employee
id: Long
name: String
//getters y setters

8. Dentro de la clase que contiene el metodo main agregar el atributo

```
private static EntityManagerFactory managerFactory = Persistence
    .createEntityManagerFactory("ejsHibernate");
```

Notar que el nombre coincide con el nombre de la persistent unit en el XML.

Hacemos esto así porque queremos tener exactamente un EntityManagerFactory en nuestra aplicación.

9. Dentro del metodo main , crear un EntityManager

```
EntityManager em = managerFactory.createEntityManager();
```

10. Luego crear una instancia de Employee

11. Agregar el código para persistir su instancia en la base de datos

```
EntityManager tran = em.getTransaction();  
tran.begin();  
em.persist(tuInstanciaDeEmployee);  
tran.commit();  
em.close();
```

12. Ejecutar el código y verificar que se creó la tabla y se insertó un registro en la base de datos (también se va a mostrar lo que se ejecuta por consola)

Ejercicio IV

Crear y persistir las siguientes clases:

Usuario, con los siguientes atributos:

```
private Long id;  
private String firstname;  
private String lastname;  
private String username;  
private String password;  
private String email;  
private int ranking;  
private boolean admin;  
private Address domicilioParticular;  
private Address domicilioTrabajo;
```

Address, con los siguientes atributos:

```
private String calle;  
private int numero;  
private String codigoPostal;  
private String ciudad;
```

Implementar un TestCase de JUnit para

a.- Crear y Persistir un usuario (con dos direcciones)

b.- Cerrar el em, y consultar la BD en otra transacción para que nos devuelva el usuario que persistimos:

Hint: `em.createQuery("select u from User u").getSingleResult();`

c.- Supongamos que conocemos el id del objeto que buscamos. Qué método de los de em convendría usar?

Ejercicio V

Convertir en persistentes las clases 'BankingTransaction', 'CashTransaction' y 'StockTransaction'. Utilizar sucesivamente las tres estrategias de herencia, para ver cómo quedan las tablas en la BD (Cambiando unas annotations)

```
public abstract class BankingTransaction {
    private Long id = null;
    private String txType = null;
    private Date txDate = null;
    private String txDescription = null;
    private Double txFee = null;
    //getters y setters
}

public class CashTransaction extends BankingTransaction {
    private boolean isDeposit = false;
    private Double moneyAmount = 0.0;
    //getters y setters
}

public class StockTransaction extends BankingTransaction {
    private boolean isSale = false;
    private String stockSymbol = null;
    private String companyName = null;
    private Integer numShares = 0;
    private Double pricePerShare = 0.0;
    //getters y setters
}
```