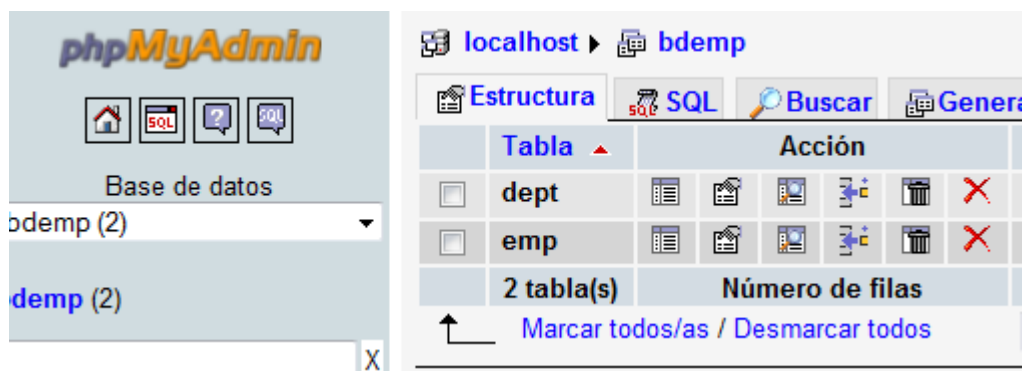


Práctica 1. Mapear tablas como objetos con JPA.

base datos bdemp de mysql

Pasos:

1. Crear unidad de persistencia
2. Crear clases de entidad desde la base de datos
3. Crear un jpa Controller
4. Crear una clase aplicación para probar, listar departamentos y dar altas



+ Opciones

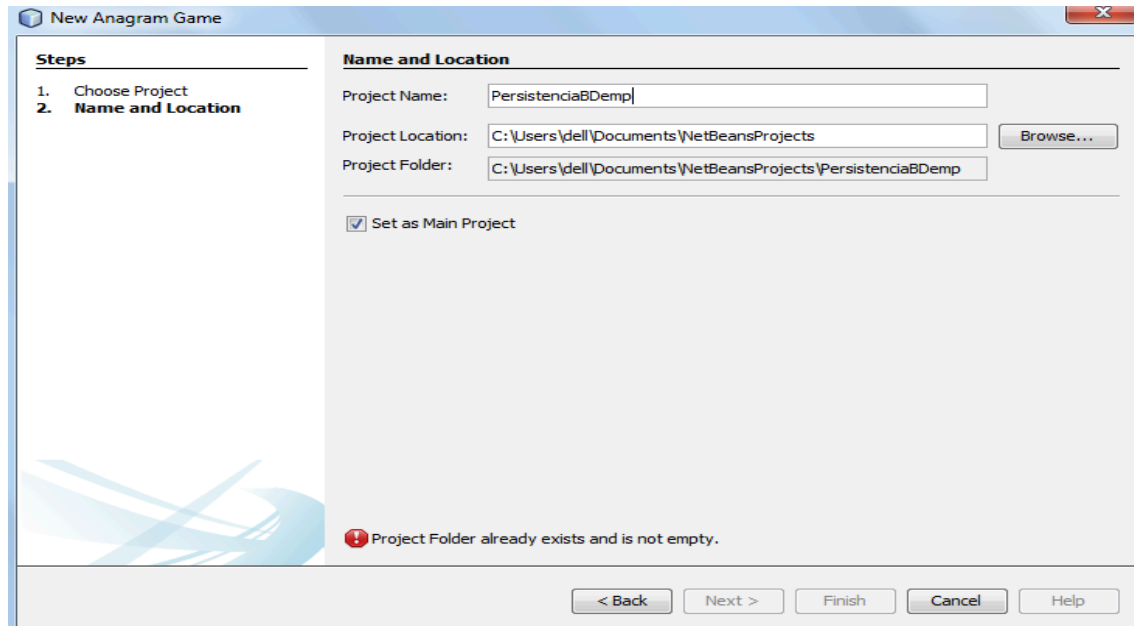
			DEPTNO	DNAME	LOC
<input type="checkbox"/>			10	ACCOUNTING	NEW YORK
<input type="checkbox"/>			20	RESEARCH	DALLAS
<input type="checkbox"/>			30	SALES	CHICAGO
<input type="checkbox"/>			40	OPERATIONS	BOSTON
<input type="checkbox"/>			50	marketing	madrid

1. Crear unidad de persistencia

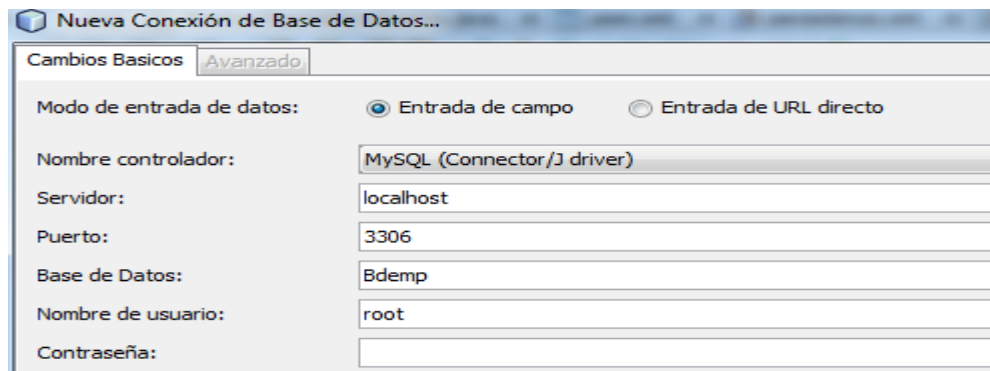
Se generará el fichero **persistence.xml**

Primero creo un proyecto llamado **PersistenciaDBemp**

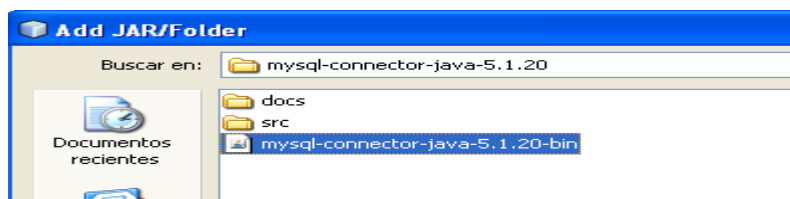
File/new Project



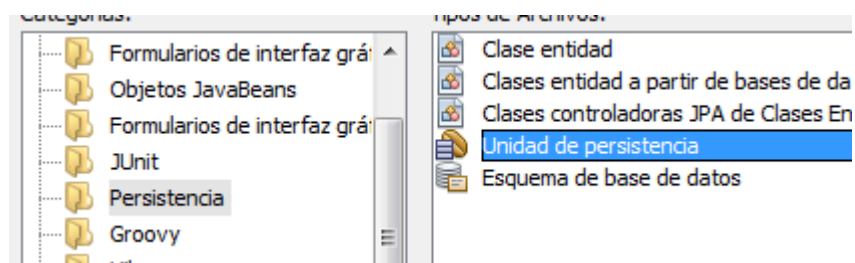
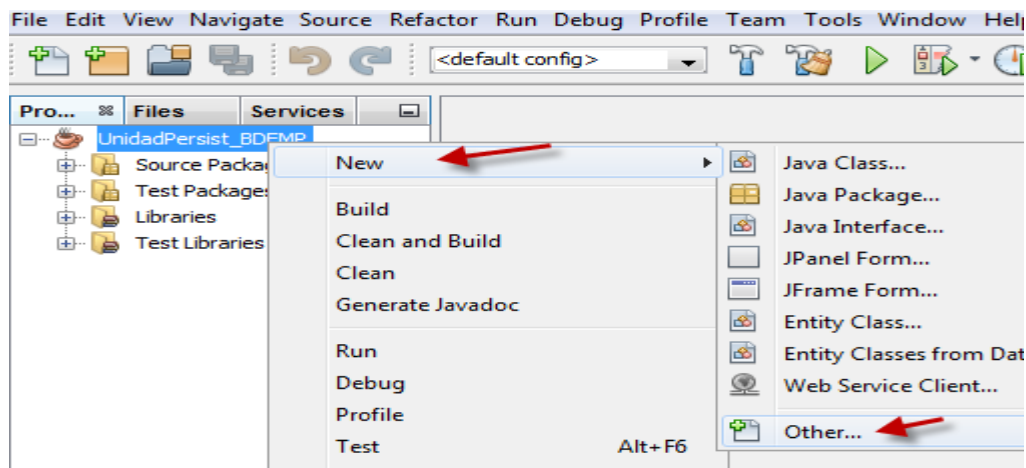
Desde neatbean creo la conexión con la bases datos bdemp



Añado la librería del conector para mysql al proyecto



Desde el proyecto creado botón derecho voy a others



New Persistence Unit	
Steps <ol style="list-style-type: none"> 1. Choose File Type 2. Provider and Database 	Provider and Database <p>Persistence Unit Name: <input type="text" value="PersistenciaBDempPU2"/></p> <p>Specify the persistence provider and database for entity classes.</p> <p>Persistence Library: <input type="text" value="EclipseLink (JPA 2.0)"/></p> <p>Database Connection: <input type="text" value="EclipseLink (JPA 2.0)"/></p> <p>Table Generation Strategy: <input type="text" value="TopLink Essentials (JPA 1.0)"/></p> <p>New Persistence Library...</p> <p>Manage Libraries...</p>



General:

Persistence Unit Name: **PersistenciaBDempPU** ← nombre de la unidad de persistencia

Persistence Library: EclipseLink (JPA 2.0)

JDBC Connection: jdbc:mysql://localhost:3306/bdemp [root on Default schema]

☐ Use Java Transaction APIs

Table Generation Strategy: ☒ Create ☐ Drop and Create ☐ None

Validation Strategy: ☒ Auto ☐ Callback ☐ None

Shared Cache Mode: ☐ All ☐ None ☐ Enable Selective ☐ Disable Selective ☒ Unspecified

En este punto además de dar un nombre a la unidad de persistencia, indicamos con que ORM vamos a trabajar.(EclipseLink, Hibernate...)

Veo el fichero persistence.xml si elijo EclipseLink como ORM

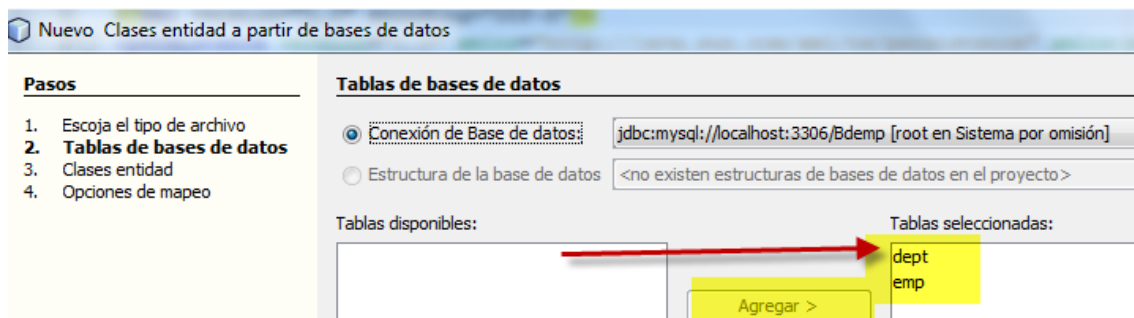
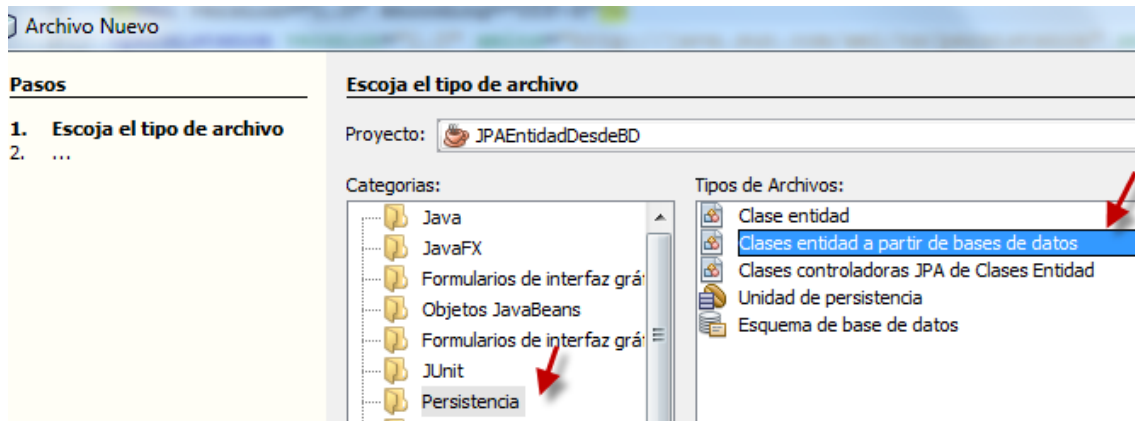
```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/
  <persistence-unit name="PersistenciaBDempPU" transaction-type="RESOURCE_LOCAL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/bdemp"/>
      <property name="javax.persistence.jdbc.password" value=""/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.user" value="root"/>
      <property name="eclipselink.ddl-generation" value="create-tables"/>
    </properties>
  </persistence-unit>
</persistence>
```

Veo el fichero persistence.xml si elijo Hibernate como ORM

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org
  <persistence-unit name="PersistenciaBDempPU" transaction-type="RESOURCE_LOCAL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/bdemp"/>
      <property name="javax.persistence.jdbc.password" value=""/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.user" value="root"/>
      <property name="eclipselink.ddl-generation" value="create-tables"/>
    </properties>
  </persistence-unit>
  <persistence-unit name="PersistenciaBDempPU2" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
      <property name="hibernate.connection.username" value="root"/>
      <property name="hibernate.connection.driver_class" value="com.mysql.jdbc.Driver"/>
      <property name="hibernate.connection.password" value=""/>
      <property name="hibernate.connection.url" value="jdbc:mysql://localhost:3306/bdemp"/>
      <property name="hibernate.cache.provider_class" value="org.hibernate.cache.NoCacheProvider"/>
      <property name="hibernate.hbm2ddl.auto" value="update"/>
    </properties>
  </persistence-unit>
</persistence>
```

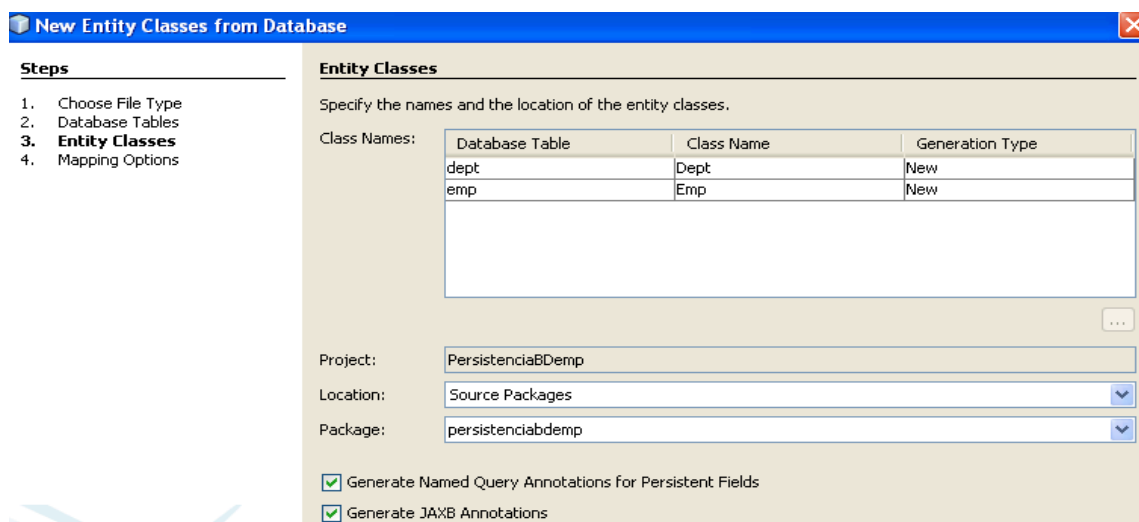
2. Crear clases entidad desde base de datos.

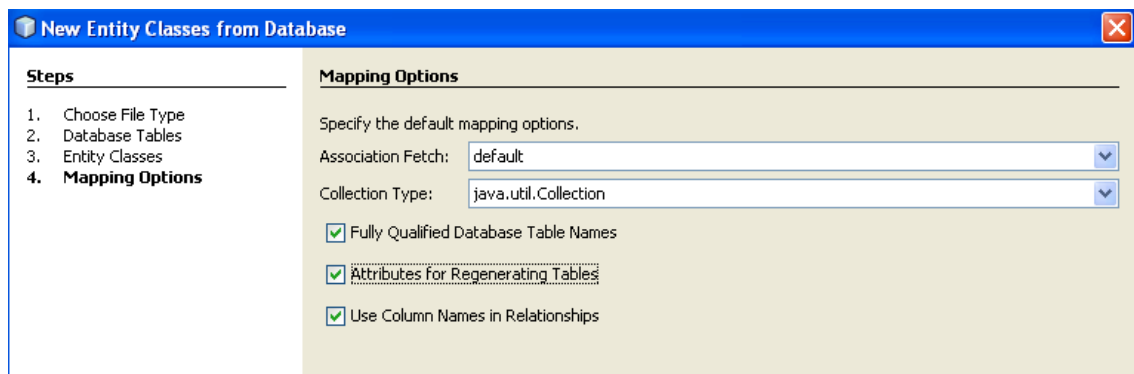
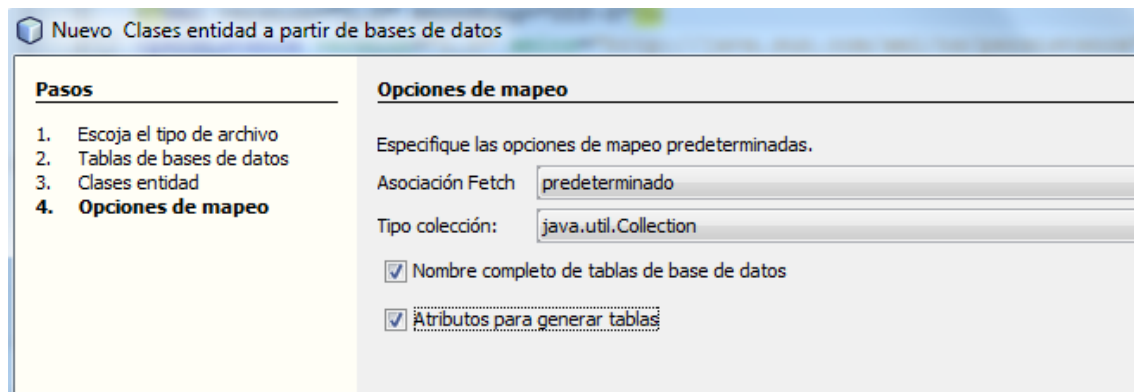
Para crear una clase de entidad persistente lo vamos a hacer de una tabla ya creada, en este caso, las tablas dept y emp. Para ello hacemos click botón derecho new/others/persistence/EntityClasses from Database



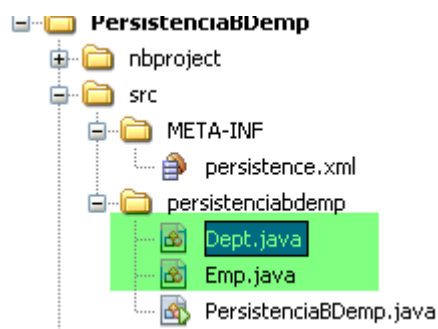
En este caso vamos a convertir las tablas dept y emp en clases persistentes llamadas igual con sus anotaciones.

Configuramos las diferentes opciones de mapeo, como por ejemplo el tipo de asociaciones o con que interfaz se manejan las colecciones etc..





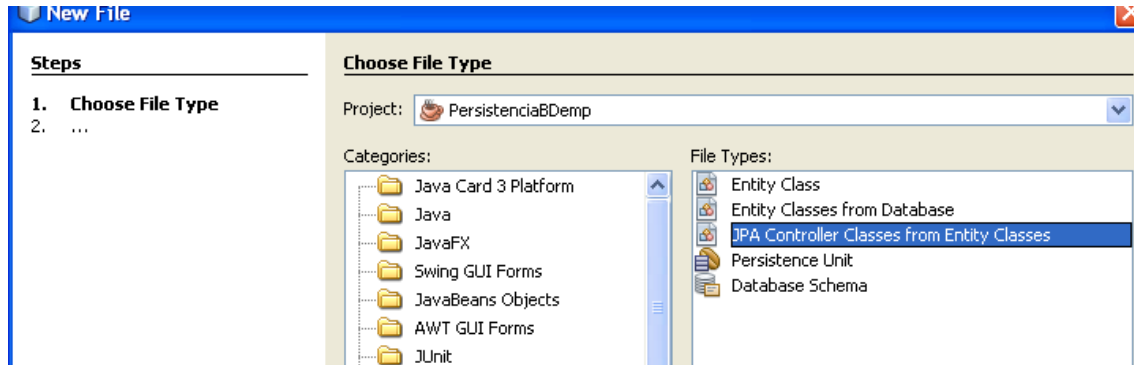
Se generan clases con una serie de atributos anotados y propiedades



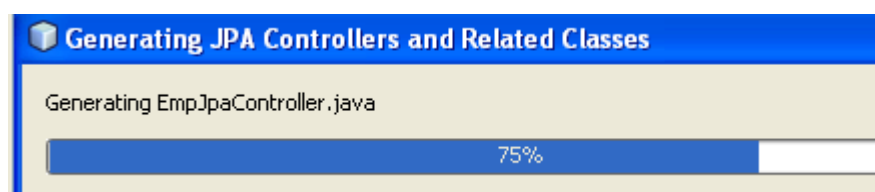
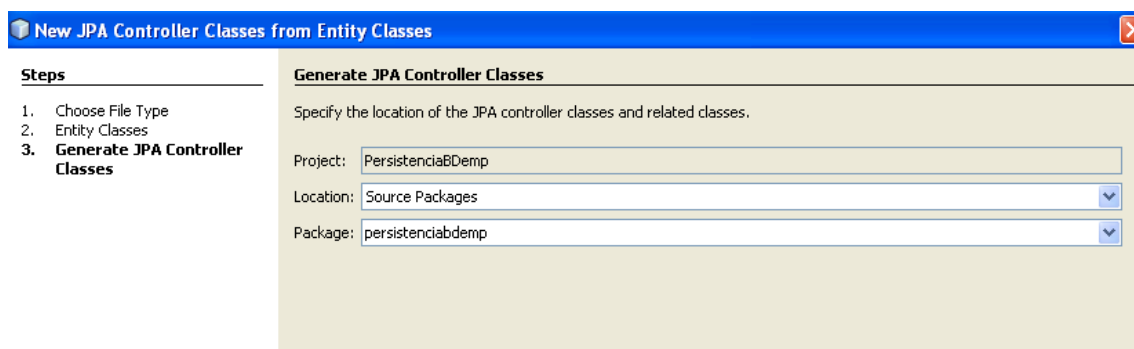
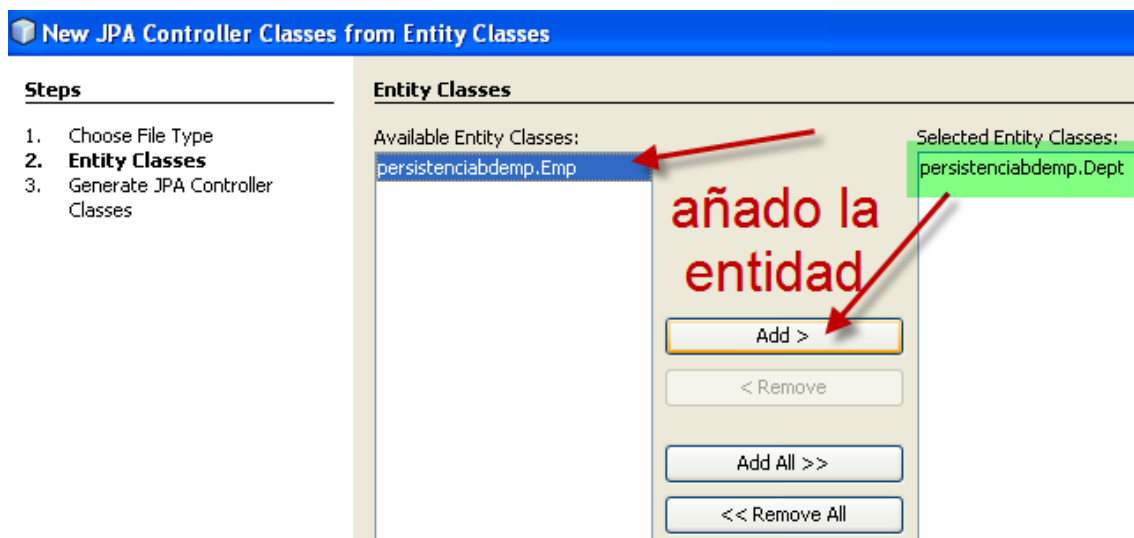
El resultado de este proceso es una clase una serie de atributos anotados y propiedades. Observa detenidamente el Código de la clase dept.java y el de emp.java

3. Crear un JPA Controller.

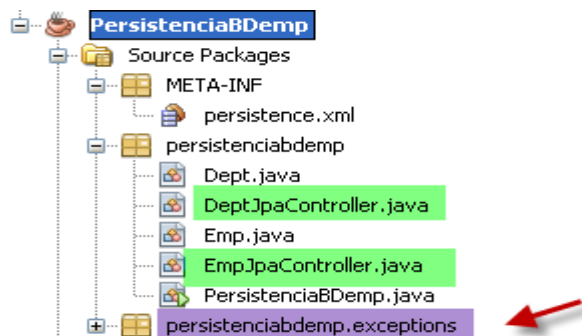
Botón derecho sobre el paquete new/others/persistence/JPA Controller Classes from Entity Classes



Seleccionamos la entidad persistente a la que vamos a crear su controlador de acceso a datos



Veremos que se crea nuestro controlador `DeptJpaController.java` y `EmpJpaController.java`, además netbeans nos genera un paquete nuevo llamado `persistenciabdemp.exceptions` que serán las excepciones que se lanzará nuestro controlador en caso de que ocurra un error.



4. Crear clase aplicación para:

a) Ejecución 1: listar los nombres de los departamentos

Previamente hay que modificar el código, **esto no siempre es necesario.**

```
public class DeptJpaController implements Serializable {

    // public DeptJpaController(UserTransaction utx, EntityManagerFactory emf) {
    public DeptJpaController( EntityManagerFactory emf) {
        //this.utx = utx;
        this.emf = emf;
    }
    // private UserTransaction utx = null;
    private EntityManagerFactory emf = null;
```

cambiar el constructor y eliminar el parametro

```
package persistenciabdemp;
import java.util.List;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class PersistenciaBDemp {
    public static void main(String[] args) {

        EntityManagerFactory emf =
            Persistence.createEntityManagerFactory("PersistenciaBDempPU");
        DeptJpaController dao = new DeptJpaController(emf);
        List<Dept> lista = dao.findDeptEntities();
        for ( Dept departamento : lista ){
            System.out.println("nombre departamento: " + departamento.getDname());
        }
    }
}
```

La salida será esta:

```
nombre departamento: ACCOUNTING
nombre departamento: RESEARCH
nombre departamento: SALES
nombre departamento: OPERATIONS
nombre departamento: marketing
BUILD SUCCESSFUL (total time: 7 seconds)
```

Si falla comprobar que en persistence.xml este añadido este código

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="PersistenciaBDempPU" transaction-type="RESOURCE_LOCAL">
    <class>persistenciabdemp.Dept</class>
    <class>persistenciabdemp.Emp</class>
  </persistence-unit>
  <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
```

b) Ejecución 2: dar de alta un departamento nuevo, con estos datos(60,RRHH,Burgos)

```
//ejecución 2 dar alta departamento
em = emf.createEntityManager();
Dept undept = new Dept();
undept.setDeptno(60);
undept.setDname("RRHH");
undept.setLoc("Burgos");
em.getTransaction().begin();
em.persist(undept);
em.getTransaction().commit();
```

Comprobar que se ha dado de alta desde una consulta a la base de datos

```
1 select * from dept
```

select * from dept ×

select * from dept ×			
Page Size: 20			
#	DEPTNO	DNAME	LOC
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON
5	50	marketing	madrid
6	60	RRHH	Burgos