Programas de Consola y JPA con NetBeans

La API de Persistencia de Java, JPA, es un marco de trabajo de programación que permite la administración de datos relacionales usando la plataforma de Java. En términos generales JPA es una capa que encapsula a la API JDBC y que nos permite, entre otras cosas, lo siguiente:

1. Establecer las **entidades**, esto es, clases cuyo estado serán persistidos en una base de datos. Esas entidades contienen información que permiten establecer el mapeo entre los objetos de la aplicación y las tablas relacionales de la base de datos.
2. Proporciona un lenguaje de consulta similar al de las consultas SQL pero que opera con objetos en lugar de tablas.
3. Maneja las transacciones.

El programa NetBeans permite, en el caso de aplicaciones de consola o de escritorio con interfaz gráfica, generar lo siguiente:

1. Una o varias entidades a partir de una tabla o tablas de una base de datos. Para ello es necesario que exista una conexión entre NetBeans y la base de datos. Para establecer esa conexión consulte el tutorial: Conexión a Bases de Datos con NetBeans.
2. Una clase controladora JPA para cada entidad. Esta clase controladora contiene los métodos para las operaciones básicas con una tabla de una base de datos: Crear una entidad y persistirla en una tabla de la base de datos, modificar una entidad o eliminar una entidad persistida en una tabla de la base de datos, consultas sobre una tabla de la base de datos.

Creación de un Proyecto

Para crear una aplicación de consola en Java con las entidades de JPA para persistir los datos en una base de datos utilizando **NetBeans** lo primero que hay que hacer es crear un proyecto. Para crear un proyecto se sigue el siguiente procedimiento:

1. Ejecute el programa **NetBeans**. Al hacerlo aparecerá la ventana principal del programa como se ilustra en la figura 1.

2. Del menú principal de NetBeans seleccione la opción **File/New Project … ,** presione las teclas **Ctrl+Mayúsculas+N** o haga clic en el icono **New Project** mostrado en la figura 2.

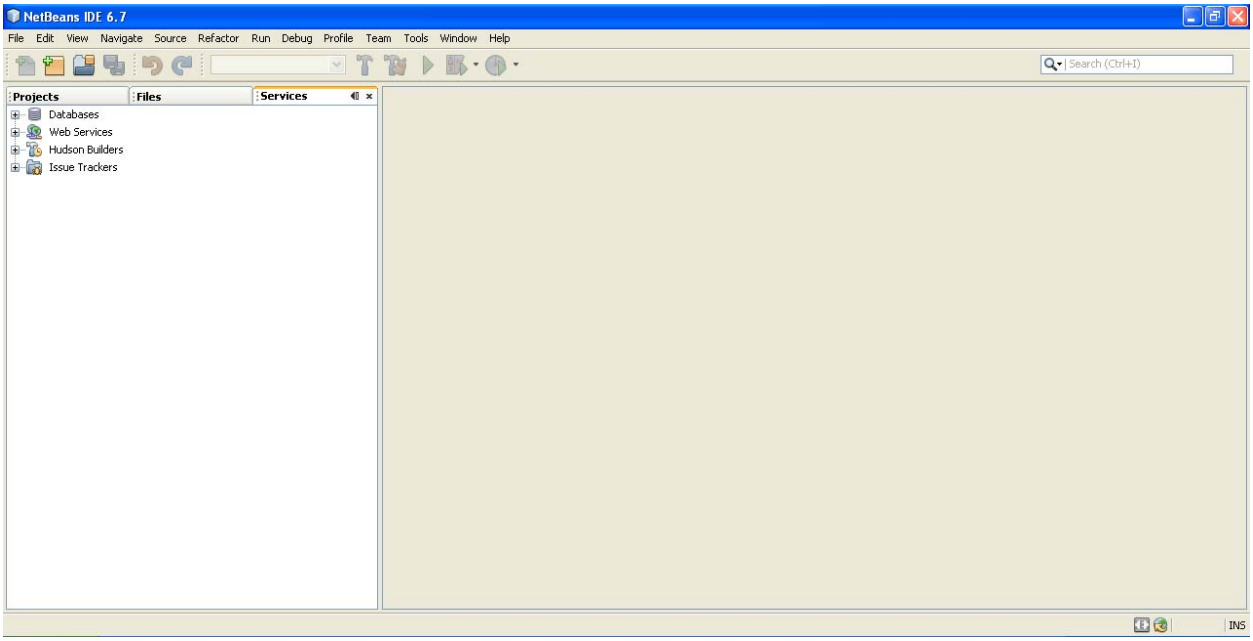

Figura 1

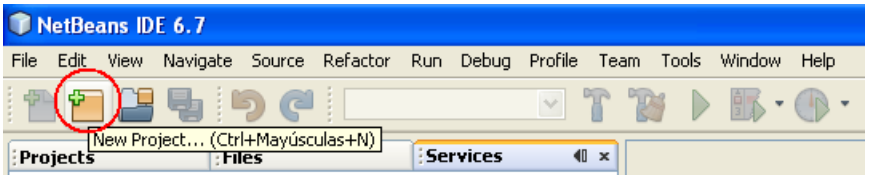

Figura 2

3. Aparece el primer cuadro de diálogo del asistente para crear un nuevo proyecto, figura 3.

4. En esta ventana del asistente seleccionaremos el tipo de proyecto que deseamos crear. Como vamos a crear una aplicación de consola, seleccionaremos la opción **Java** en el recuadro **Categories:** y la opción **Java Application** en el recuadro **Projets:**, y luego presionaremos el botón **Next>**.

5. Aparecerá la segunda ventana del asistente para crear proyectos, figura 4. En esta ventana seleccionaremos el nombre y la ubicación del proyecto.

   a) Establezca el nombre del proyecto (**Project Name**): Por ejemplo, "**amanteMusicaPersistenciaJPA**".

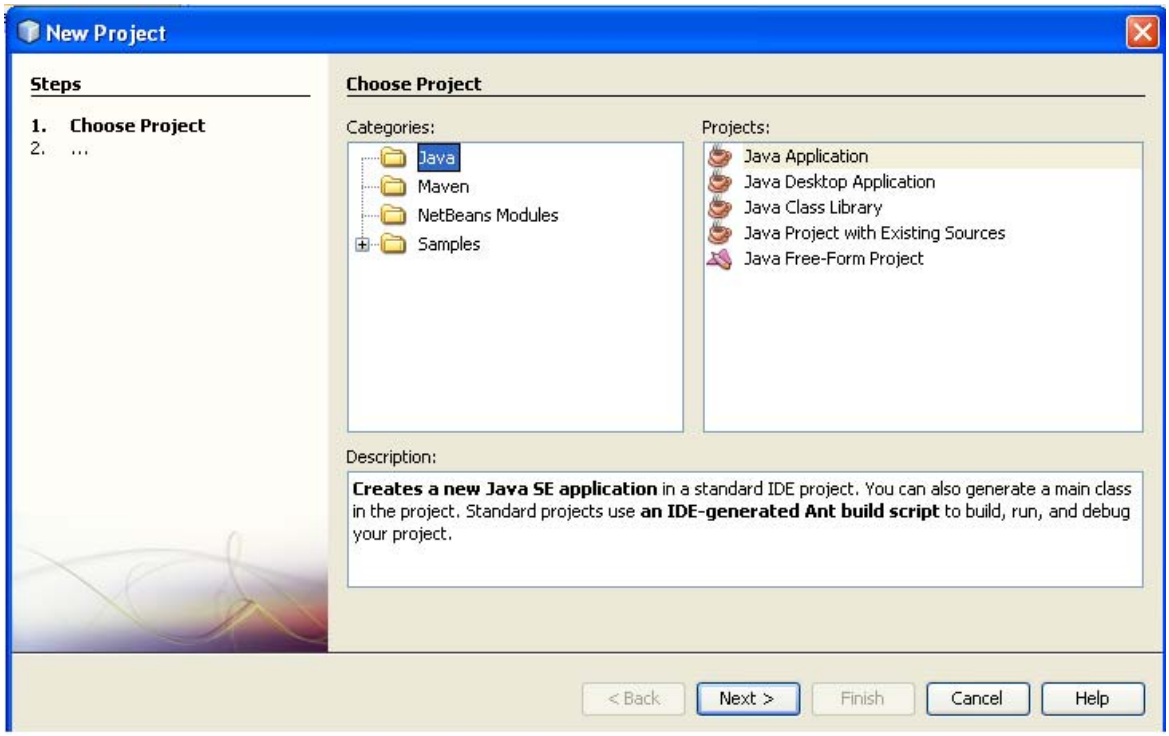

Figura 3

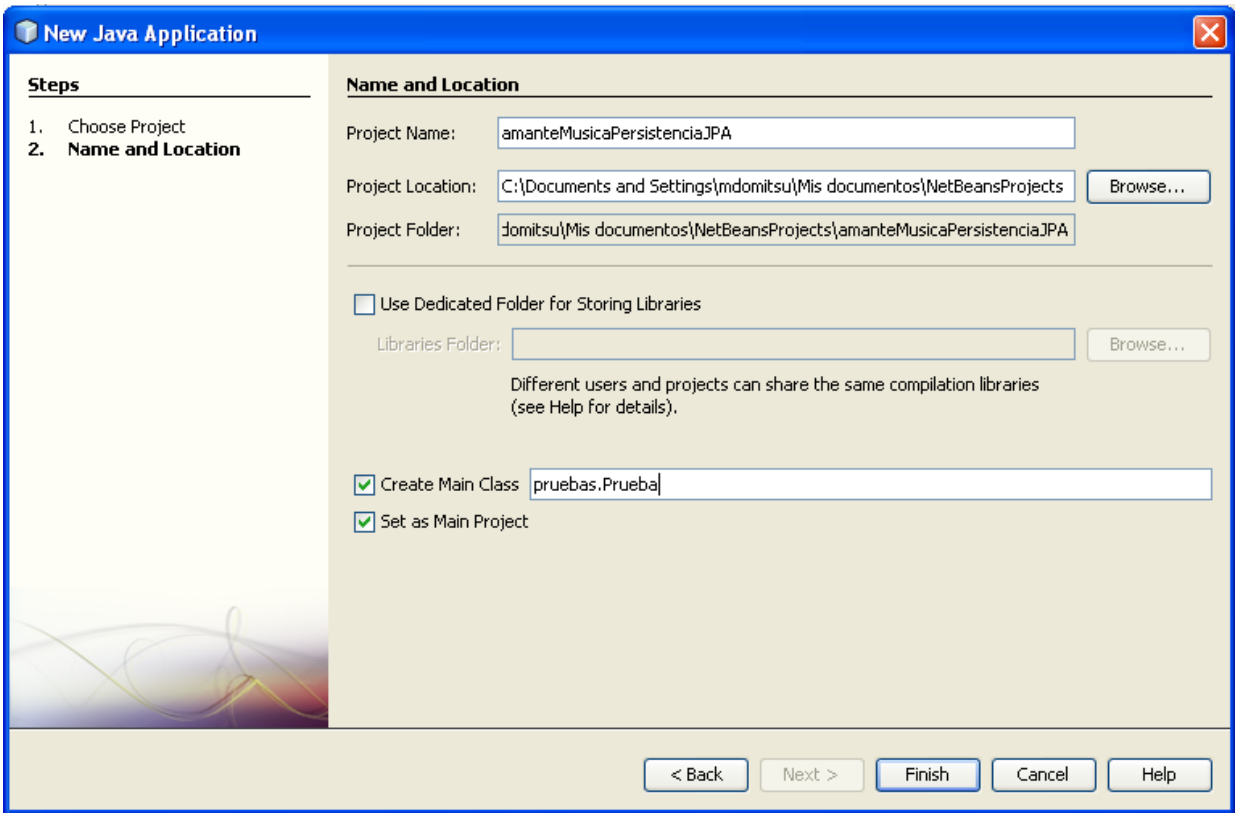

Figura 4

b) Establezca el directorio donde se almacenará el proyecto (**Projet Location**). Por ausencia en Windows 2000 y XP, el directorio es el directorio inicial del usuario: "**C:\Documents and Settings***usuario*". En este ejemplo, el proyecto se ubicó en: "**C:\Documents and Settings\mdomitsu\Mis documentos\nbproject**". En la línea siguiente puede verse la ubicación del directorio en el que se almacenarán los archivos del proyecto: **Project Location**, que es el directorio con el nombre del proyecto dentro del directorio donde se ubica el proyecto. En este ejemplo es: "**C:\Documents and Settings\mdomitsu\Mis documentos\nbproject\amanteMusicaObjNeg**".
c) Asegúrese que las casillas de selección: **Create Main Class** (Cree la clase principal, la clase con el método main()) y **Set as Main Project** (Haga que este proyecto sea el proyecto principal)estén seleccionadas.
d) En el campo de texto al lado de la casilla **Create Main Class** se establece el nombre de la clase principal. El valor por ausencia es: *nombreProyecto***.Main**, indicando que la clase principal se llamará Main y estará en el paquete *nombreProyecto*. Cambie ese valor a **pruebas.Prueba**.
e) Presione el botón **Finish**.

6. Desaparecerá el asistente para crear un nuevo proyecto y aparecerá lo mostrado en la figura 5. Del lado derecho aparece el editor de NetBeans con el esqueleto de la clase principal: Prueba.java, mientras que del lado izquierdo aparece el árbol de los proyectos, que en este momento sólo tiene el proyecto **amanteMusicaObjNeg**.

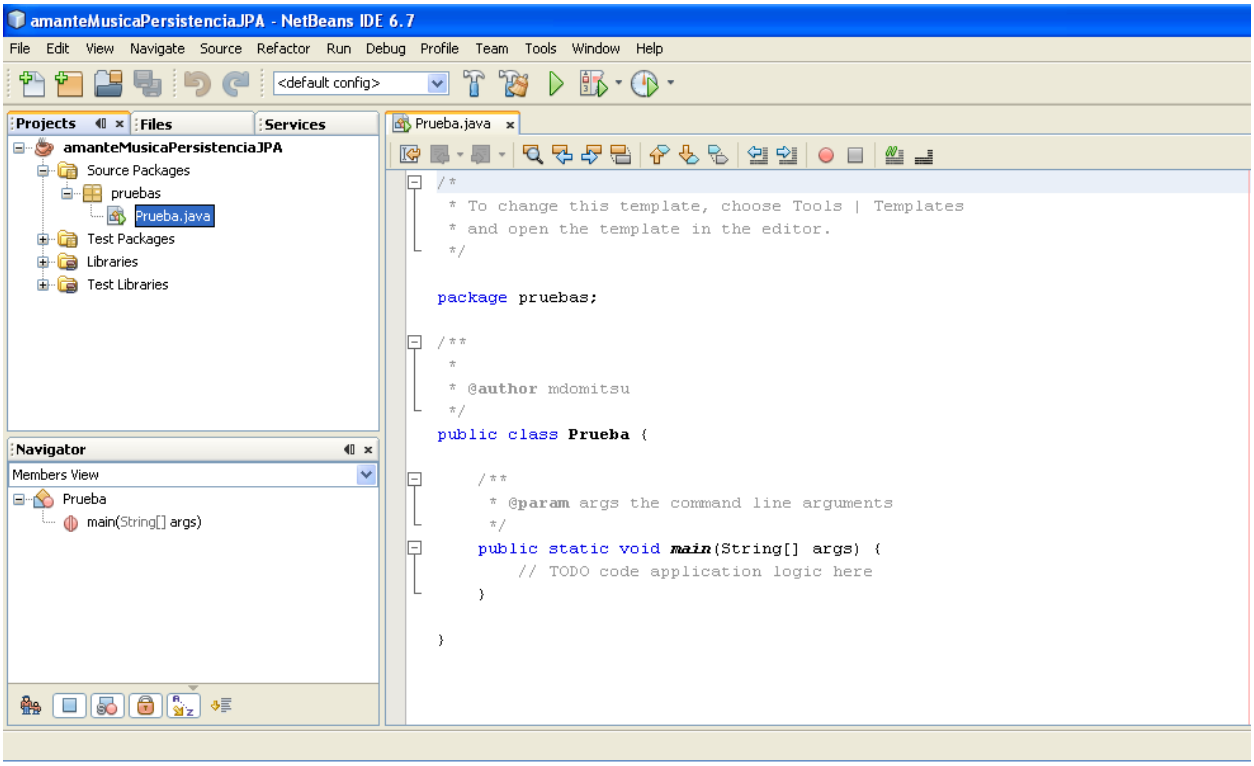

Figura 5

Creación de las Entidades a partir de una Base de Datos

En este tutorial supondremos que deseamos crear las entidades que corresponden al diagrama de clases mostrado en la figura 6:

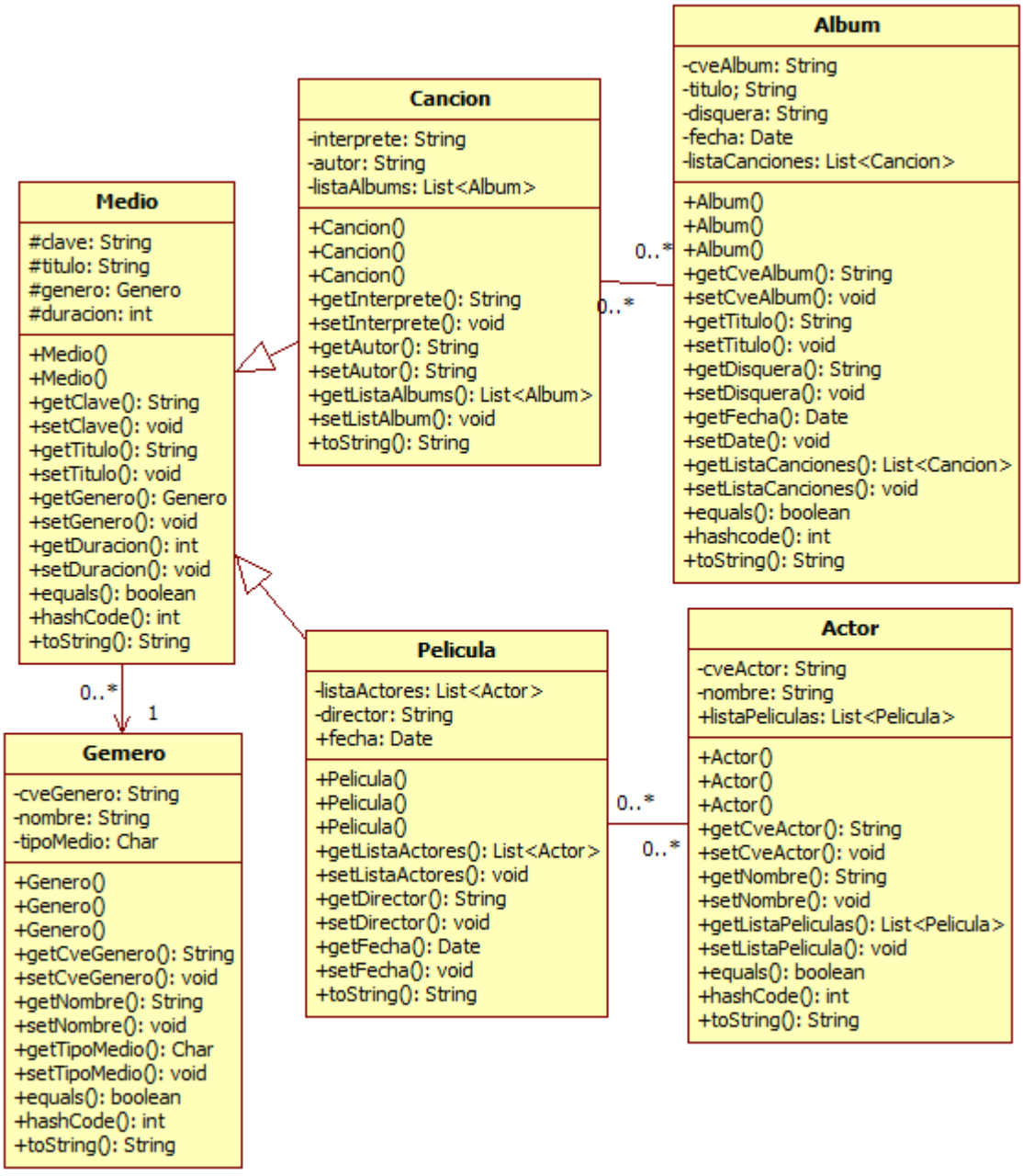

Figura 6

Esas entidades serán persistidas en la base de datos musicaJPA cuyo diagrama entidad – relación es el mostrado en la figura 7.

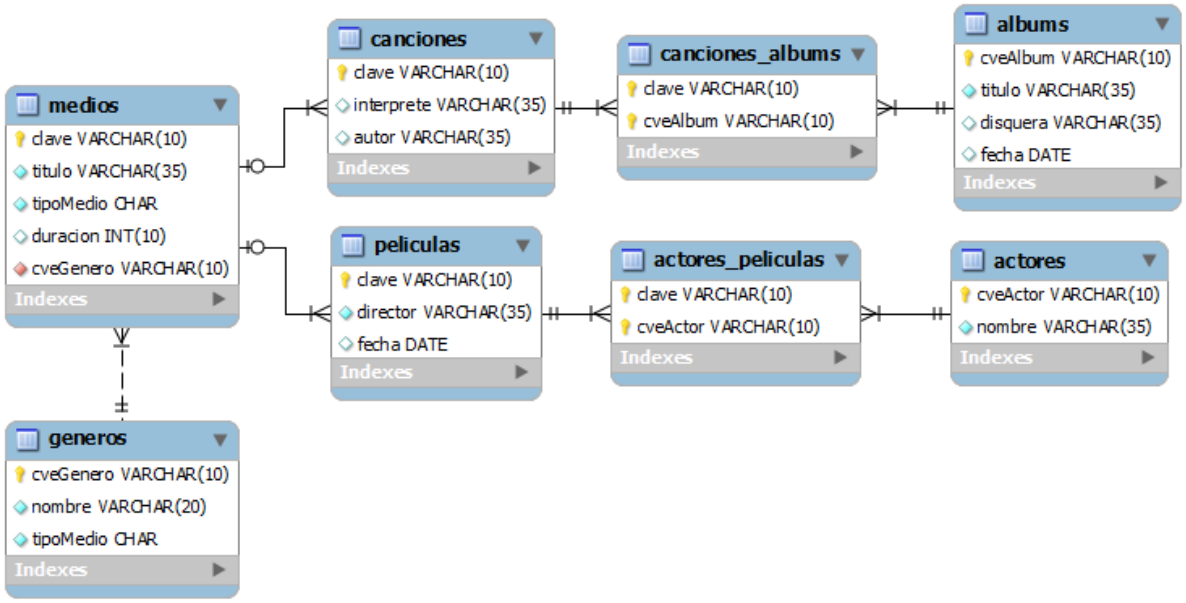

Figura 7

Recuerde que para poder generar las entidades a partir de una base de datos, es necesario que NetBeans esté conectada a esa base de datos. Si no es el caso, siga el procedimiento del tutorial: Conexión a Bases de Datos con NetBeans.

Para crear las entidades a partir de una base de datos se sigue el siguiente procedimiento:

1. Haga clic con el botón derecho en el nodo **Source Packages** del árbol del proyecto. Del menú emergente seleccione la opción **New** y del nuevo menú emergente seleccione la opción **Entity Classes from Database ...**, como se muestra en la figura 8:

2. Aparecerá el primer cuadro de diálogo del asistente para crear una o varias entidades a partir de una base de datos, figura 9. Aquí seleccionaremos la conexión con la base de datos y las tablas de las que querramos generar entidades.

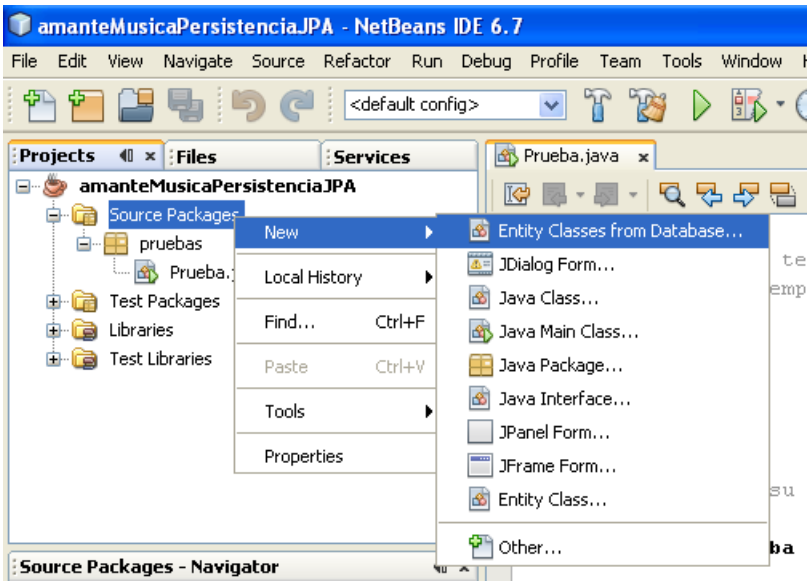

Figura 8

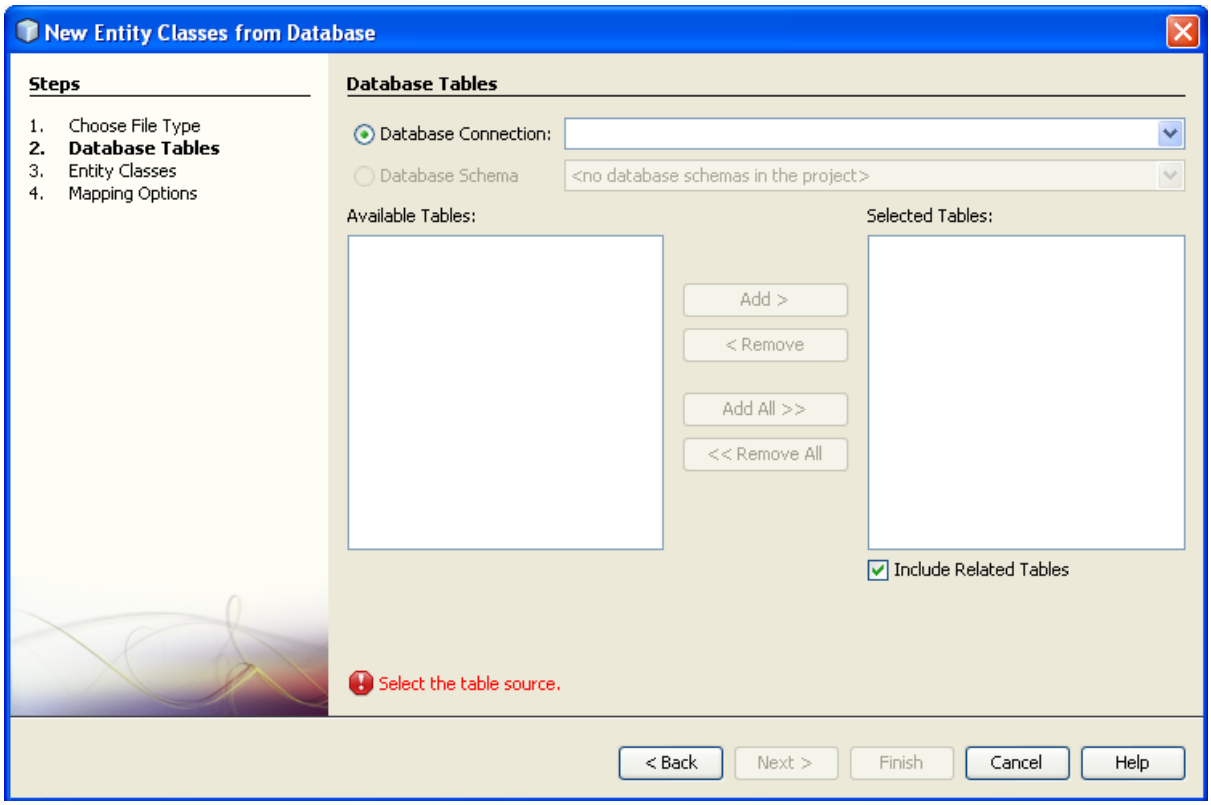

Figura 9

3. De la caja combinada Database Connection: seleccione la conexión a la base de datos deseada, **musicaJPA** en este caso, figura 10.

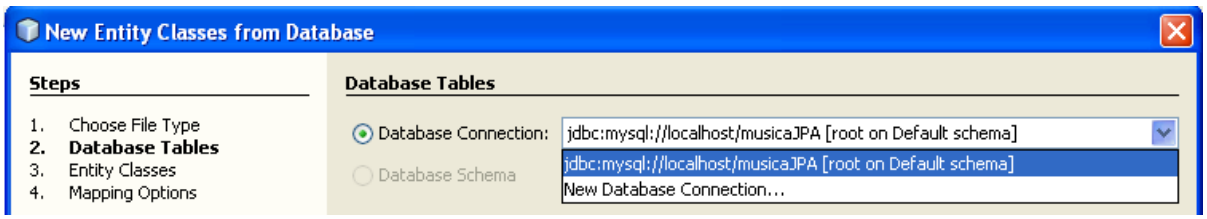

Figura 10

4. Aparecerá el área de texto: **Available Tables**: la lista de tablas de la base de datos musica, figura 11. Seleccione las tablas deseadas y presione el botón **Add >**. En este caso, como deseamos crear entidades para todas las tablas, simplemente presionaremos el botón **Add All >>**.

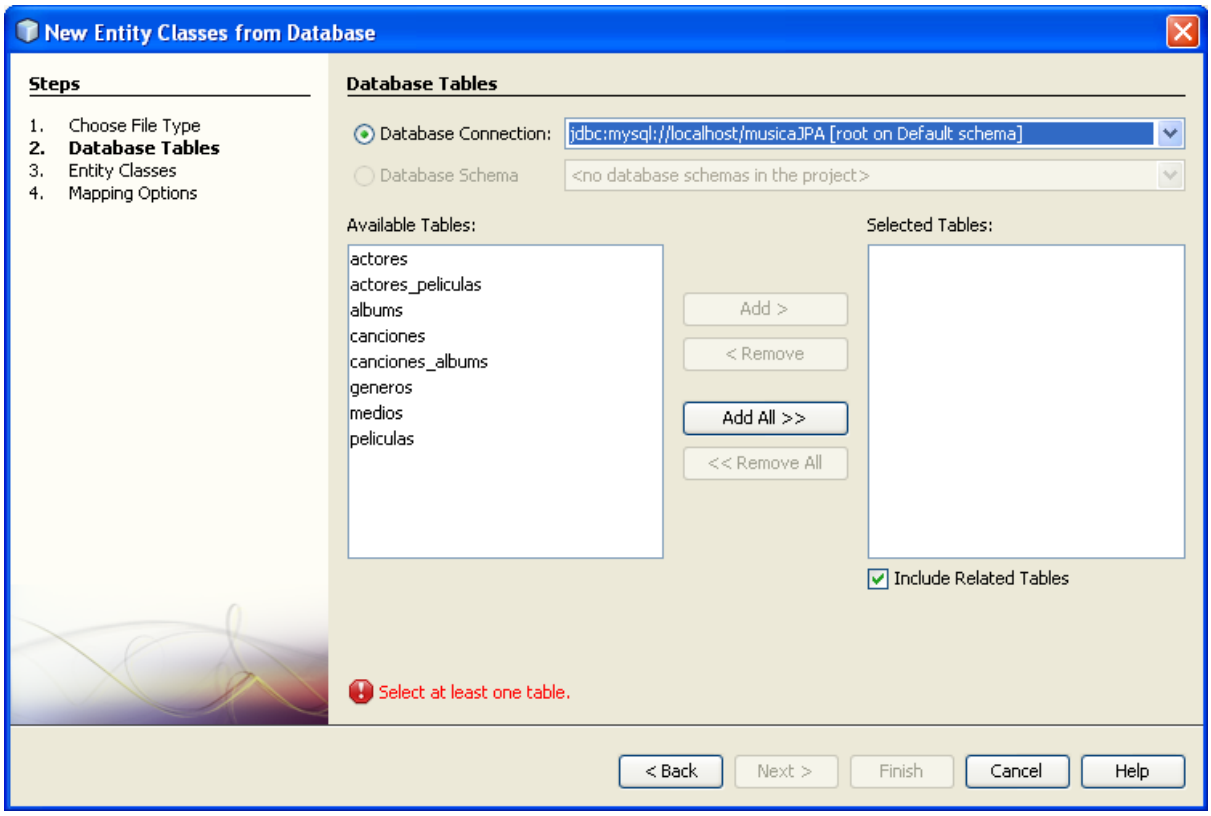

Figura 11

5. La lista de las tablas seleccionadas aparecerá en el área de texto **Selected Tables**, figura 12. Presione el botón **Next**.

6. Aparecerá el cuadro de diálogo de la figura 13. Aquí estableceremos el nombre de las entidades y su ubicación. También crearemos la unidad de persistencia.

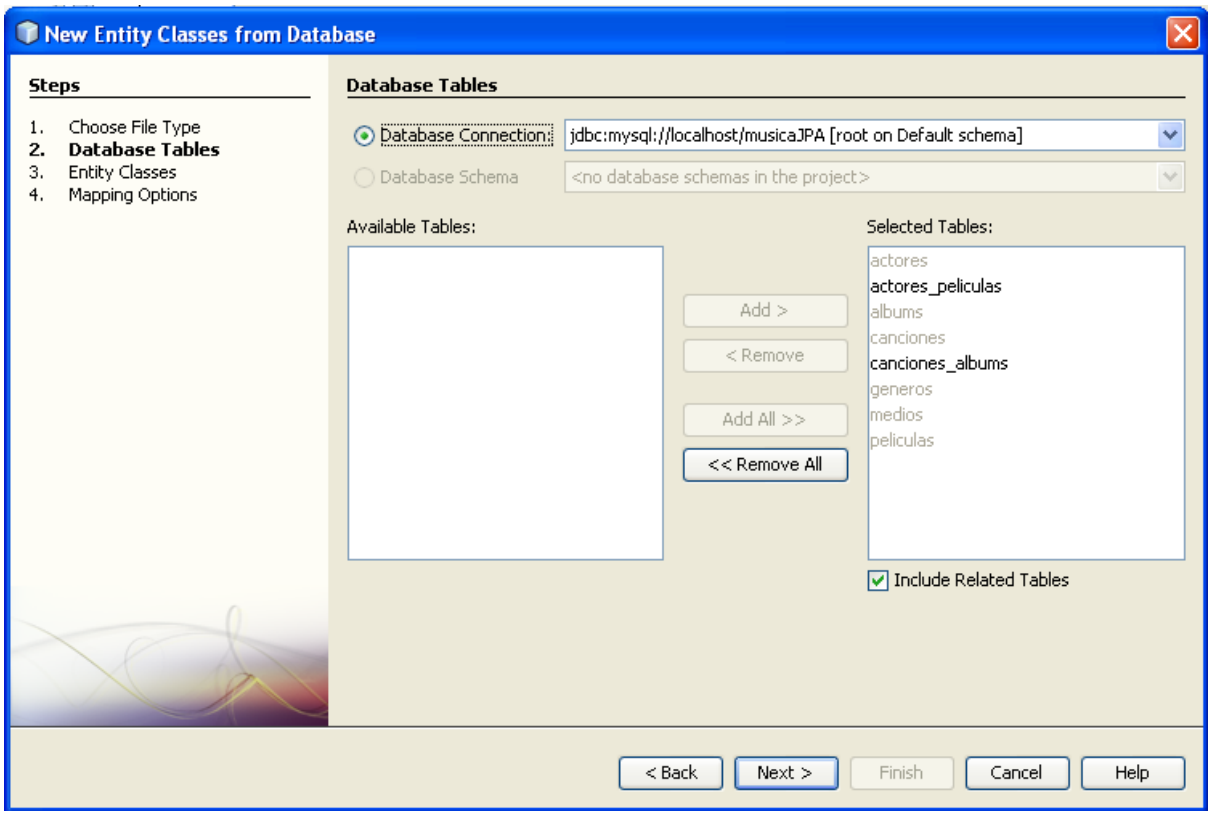

Figura 12

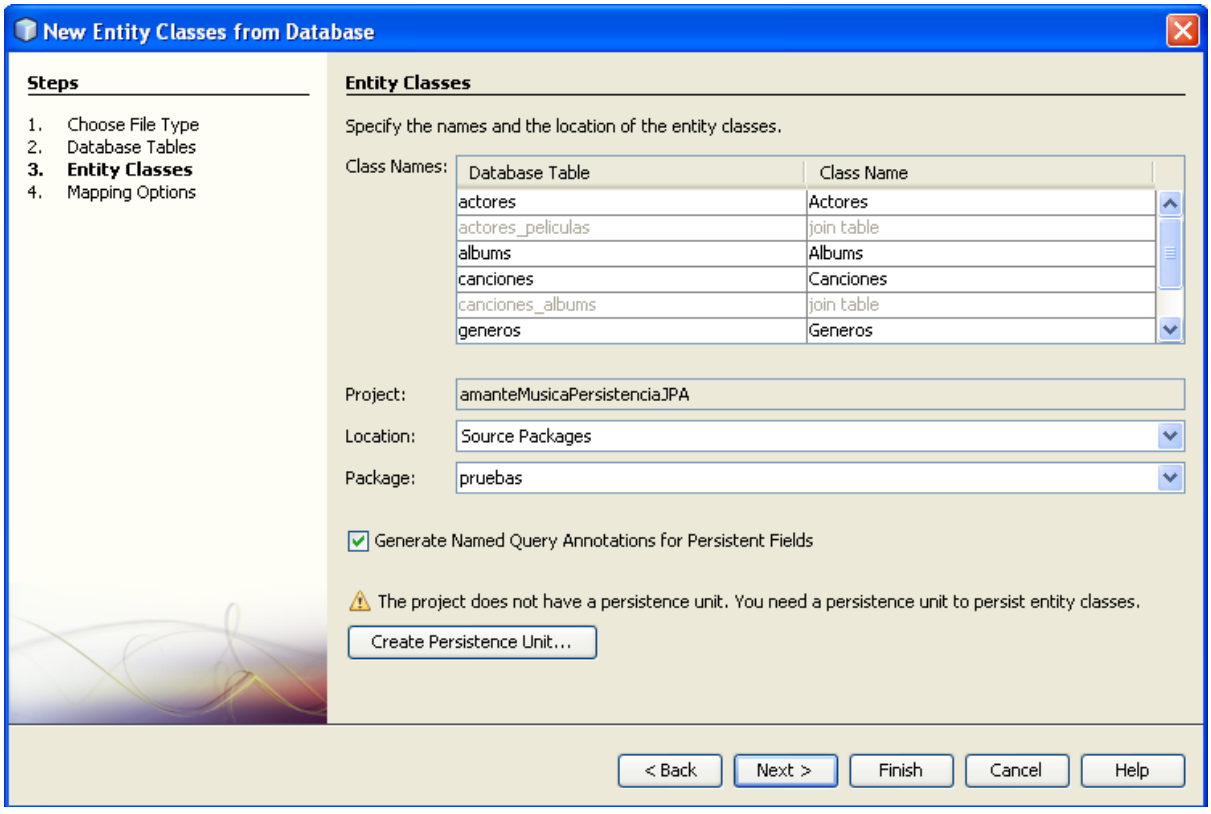

Figura 13

7. Modifique los nombres sugeridos por Netbeans para las entidades bajo la columna **Class Name** a `Actor`, `Album`, `Cancion`, `Genero`, `Medio` y `Pelicula`. Las tablas conjuntas no generan una entidad. También establezca el nombre del paquete en el que se almacenarán las entidades a `objetosNegocio`. A continuación presione el botón **Create Persistence Unit...**. Aparecerá el cuadro de diálogo de la figura 14.

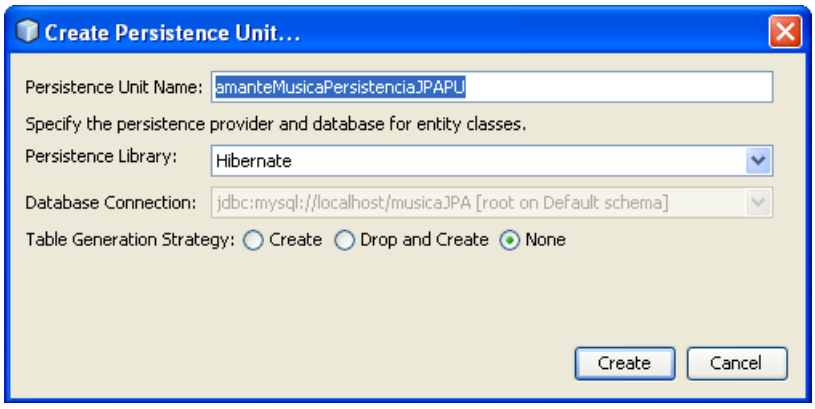

Figura 14

8. En este cuadro de diálogo estableceremos el nombre de la unidad de persistencia, **Persistence Unit Name**: dejando el nombre sugerido. De la caja combinada seleccionaremos la biblioteca de persistencia a emplear, **Persistence Library:** Seleccione la opción **TopLink**, figura 15.

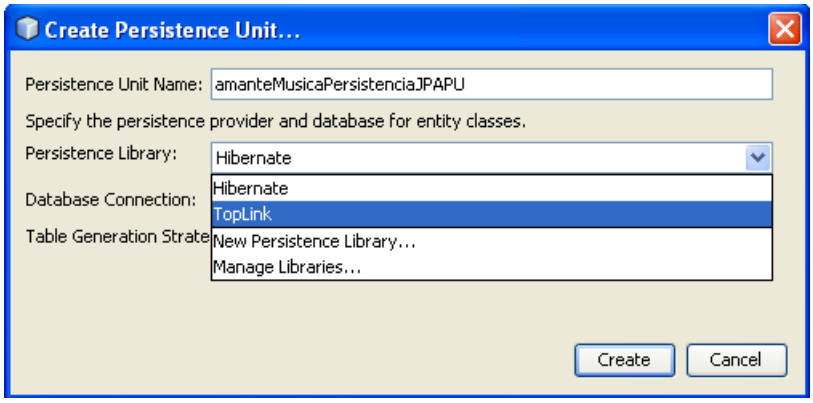

Figura 15

9. Deje los demás valores sin modificar y presione el botón **Create**. Regresaremos al cuadro de diálogo anterior, figura 16.

10. Presione el botón **Next**. Aparecerá el cuadro de diálogo de la figura 17.

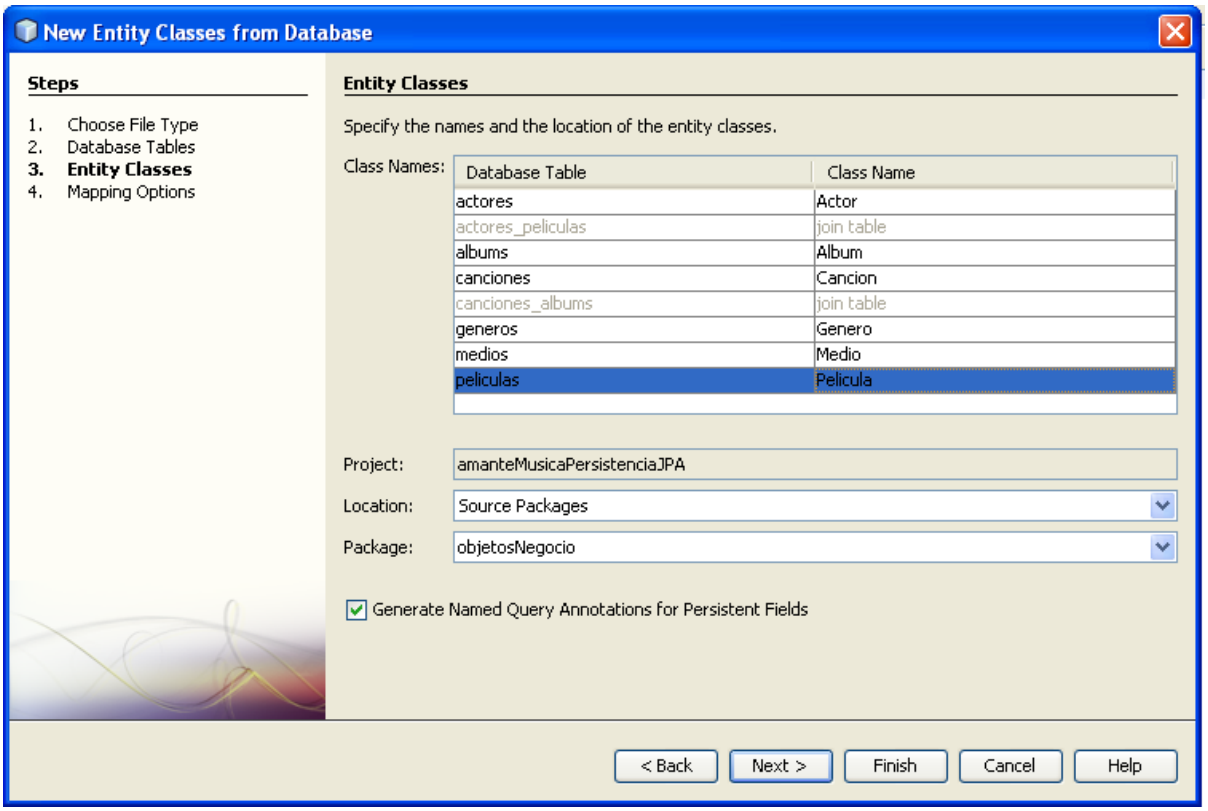

Figura 16

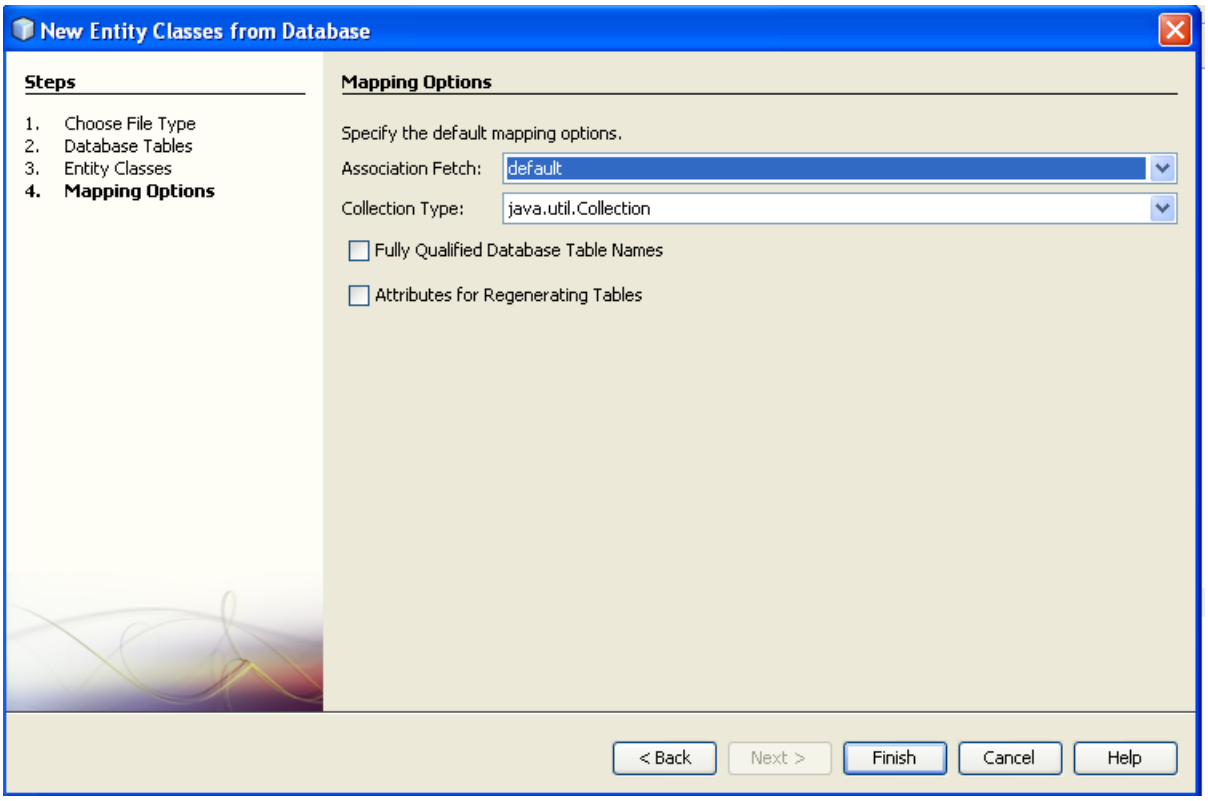

Figura 17

11. En este cuadro de diálogo seleccionaremos las opciones de mapeo. El único cambio lo haremos en la caja combinada **Collection Type:** Seleccionaremos la opción `java.util.List`, figura 18.

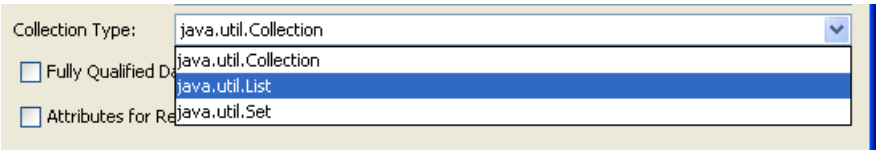

Figura 18

12. Por último, presione el botón **Finish**. NetBeans generará las entidades como se pueden ver en el árbol del proyecto, figura 19.

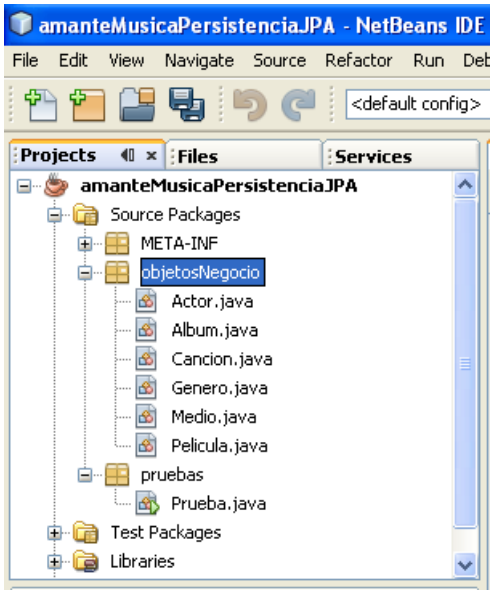

Figura 19

Edición del Código de las Entidades

A continuación se modificarán las entidades generadas por NetBeans:

1. En la entidad `Genero` que se encuentra en el archivo **Genero.java**:

   a. Convierte la relación muchos a uno bidireccional entre las entidades `Medio` y `Genero` a una relación unidireccional , eliminando las líneas:

```
@OneToMany(cascade = CascadeType.ALL, mappedBy = "cveGenero")
private List<Medio> medioList;
```

   y los métodos de acceso: `getMedioList()` y `setMedioList()`.

b. Si no existe, agrega un constructor que inicialice todos los atributos de la entidad.

c. Modifique el método `toString()` para que regrese una cadena con los valores de los atributos de la entidad separados por comas.

2. En la entidad `Medio` que se encuentra en el archivo **Medio.java**:

a. Despues de la anotación `@Table`, agrega las siguientes líneas:

```
@Inheritance(strategy = Inheritancetype.JOINED)
@DiscriminatorColumn(name = “tipoMedio”,
     discriminatorType = DiscriminatorType.CHAR)
```

para establecer que esta entidad va a ser la entidad padre de una jerarquía que se va a persistir usando la estrategia de tablas conjuntas y que la columna discriminadora se llama “tipoMedio” y será del tipo char.

b. Cambia el modificador de acceso de todos los atributos de la entidad de `prívate` a `protected`.

c. Elimine las líneas:

```
@OneToOne(cascade = CascadeType.ALL, mappedBy = "medio")
private Pelicula pelicula;

@OneToOne(cascade = CascadeType.ALL, mappedBy = "medio")
private Cancion cancion;
```

y los métodos: `getCancion()`, `setCancion()`, `getPelicula()` y `setPelicula()` ya que las relaciones entre las entidades Medio, Canción y Pelicula no son uno a uno sino de herencia.

d. Renombre el atributo `cveGenero` a `genero`.

e. Renombre los métodos `getCveGenero()` a `getGenero()` y `setCveGenero()` a `setGenero()`.

f. Renombra el parámetro `cveGenero` del método `setGenero()` a `genero`.

g. Si no existe, agrega un constructor que inicialice todos los atributos de la entidad.

h. Modifique el método `toString()` para que regrese una cadena con los valores de la clave, el título y el nombre del género separados por comas.

3. En la entidad `Cancion` que se encuentra en el archivo **Cancion.java**:

a. Establece que la entidad Cancion hereda de la entidad Medio.

b. Después de la anotación `@Table`, agrega las siguientes líneas:

```
@DiscriminatorValue(value = "C")
@PrimaryKeyJoinColumn(name = "clave")
```

para establecer que esta entidad (que hereda de la clase Medio) va a descriminarse por el valor “C” en la columna descriminadora de la tabla asociada a la entidad padre. También se establece que las entidades van a relacionarse por el valor de la llave primaria clave.

c. Elimina las líneas:

```
@JoinColumn(name = "clave",
            referencedColumnName = "clave",
            insertable = false, updatable = false)
@OneToOne(optional = false)
private Medio medio;
```

y los métodos `getMedio()` y `setMedio()` ya que las relaciones entre las entidades Medio, Canción y Pelicula no son uno a uno sino de herencia.

d. Elimina las líneas:

```
@Id
@Basic(optional = false)
@Column(name = "clave")
private String clave;
```

y los métodos: `public Clave getClave()`, `public void setClave()`, `public int hashCode()` y `public boolean equals()` ya que este atributo ya se encuentra en la entidad padre.

e. Renombre el atributo `albumList` a `listaAlbums`.

f. Renombre los métodos `getAlbumList()` a `getListaAlbums()` y `setAlbumList()` a `setListaAlbums()`.

g. Renombra el parámetro `albumList` del método `setListaAlbums()` a `listaAlbums`.

h. Modifica el constructor:

```
public Cancion(String clave) {
   this.clave = clave;
}
```

a

```
public Cancion(String clave) {
    super(clave);
}
```

i. Si no existe, agrega un constructor que inicialice todos los atributos de la entidad (incluyendo la entidad padre) menos `listaAlbums`.

j. Modifica el método `toString()` para que regrese una cadena con lo que regresa el método `toString()` de su clase padre.

4. En la entidad `Pelicula` que se encuentra en el archivo **Pelicula.java**:

   a. Repite los pasos 3.a a 3.d y 3.i.

   b. Renombre el atributo `actorList` a `listaActores`.

   c. Renombre los métodos `getActorList()` a `getListaActores()` y `setActorList()` a `setListaActores()`.

   d. Renombra el parámetro `actorList` del método `setListaActores()` a `listaActores`.

   e. Modifica el constructor:

```
public Pelicula(String clave) {
   this.clave = clave;
}
```

   a

```
public Pelicula(String clave) {
    super(clave);
}
```

   f. Si no existe, agrega un constructor que inicialice todos los atributos de la entidad (incluyendo la entidad padre) menos `listaActores`.

   g. Repite el paso 3.ji.

5. En la entidad `Album` que se encuentra en el archivo **Album.java**:

   a. Modifica la línea:

```
@ManyToMany(mappedBy = "albumCollection")
```

   a

```
@ManyToMany(mappedBy = "ListaAlbums")
```

b. Renombre el atributo `cancionList` a `listaCanciones`.

c. Renombre los métodos `getCancionList()` a `getListaCanciones()` y `setCancionList()` a `setListaCanciones()`.

d. Renombra el parámetro `cancionList` del método `setListaCanciones()` a `listaCanciones`.

e. Si no existe, agrega un constructor que inicialice todos los atributos de la entidad (incluyendo la entidad padre) menos `listaCanciones`.

f. Modifica el método `toString()` para que regrese una cadena con la clave y título del album.

6. En la entidad `Actor` que se encuentra en el archivo **Actor.java**:

a. Modifica la línea:

```
@ManyToMany(mappedBy = " actorCollection")
```

A

```
@ManyToMany(mappedBy = " listaActores")
```

b. Renombre el atributo `cancionList` a `listaCanciones`.

c. Renombre los métodos `getCancionList()` a `getListaCanciones()` y `setCancionList()` a `setListaCanciones()`.

d. Renombra el parámetro `cancionList` del método `setListaCanciones()` a `listaCanciones`.

e. Si no existe, agrega un constructor que inicialice todos los atributos de la entidad (incluyendo la entidad padre) menos `listaPeliculas`.

f. Modifica el método `toString()` para que regrese una cadena con la clave y nombre del actor.

Creación de las Clases Controladoras JPA a partir de las Entidades

Para crear las clases controladoras JPA a partir las entidades se sigue el siguiente procedimiento:

1. Haga clic con el botón derecho en el nodo **Source Packages** del árbol del proyecto. Del menú emergente seleccione la opción **New** y del nuevo menú emergente seleccione la opción **JPA Controller Classes from Entity Classes ...**, como se muestra en la figura 20:

2. Aparecerá el primer cuadro de diálogo del asistente para crear una o varias clases controladoras de entidades a partir de las entidades figura 21.

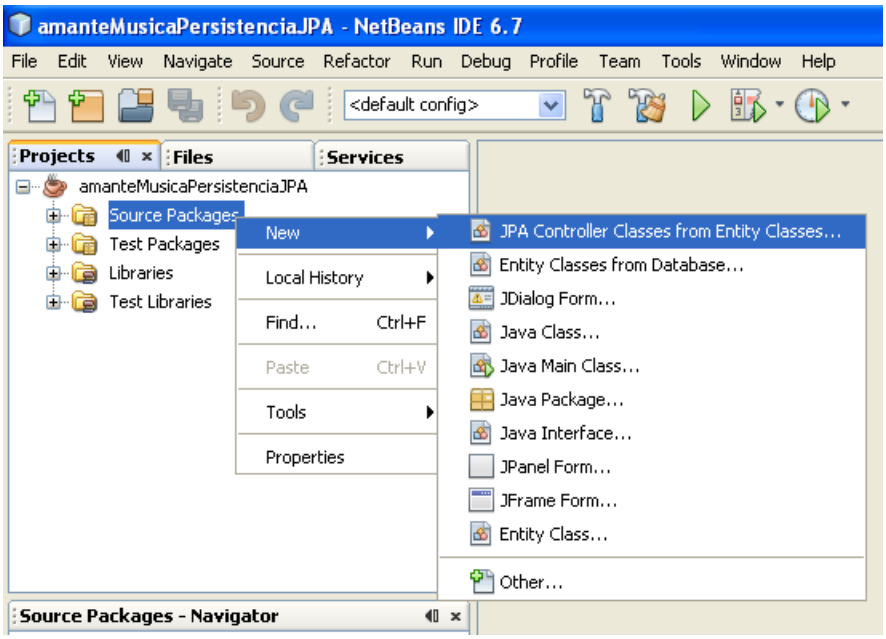

Figura 20

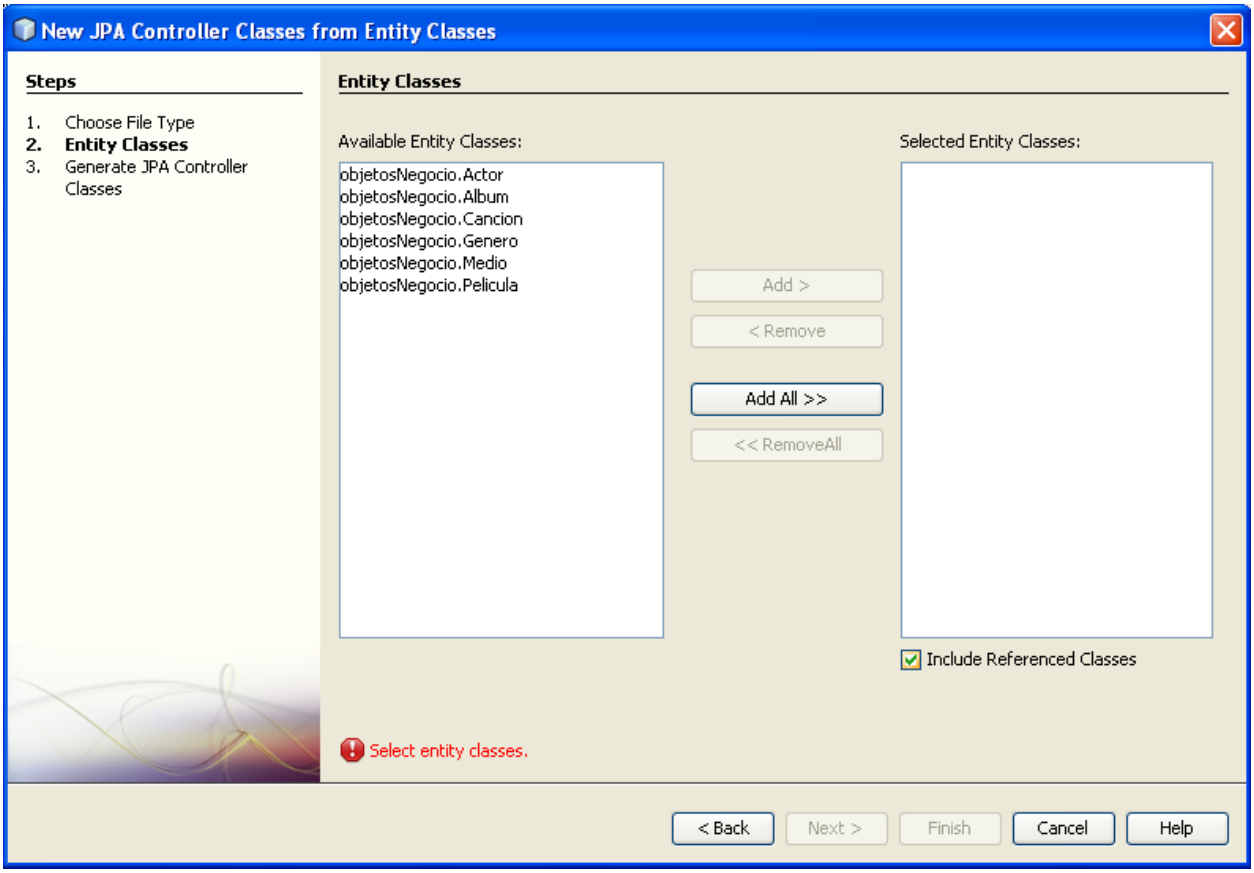

Figura 21

3. Aquí seleccionaremos las entidades de las que querramos generar clases controladoras JPA. En el área de texto: **Available Entity Classes**: está la lista de entidades. Seleccione las entidades deseadas y presione el botón **Add >**. En este caso, como deseamos crear clases controladoras JPA para todas las entidades, simplemente presionaremos el botón **Add All >>**.

4. La lista de las entidades seleccionadas aparecerá en el área de texto **Selected Entity Classes**, figura 22. Presione el botón **Next**.

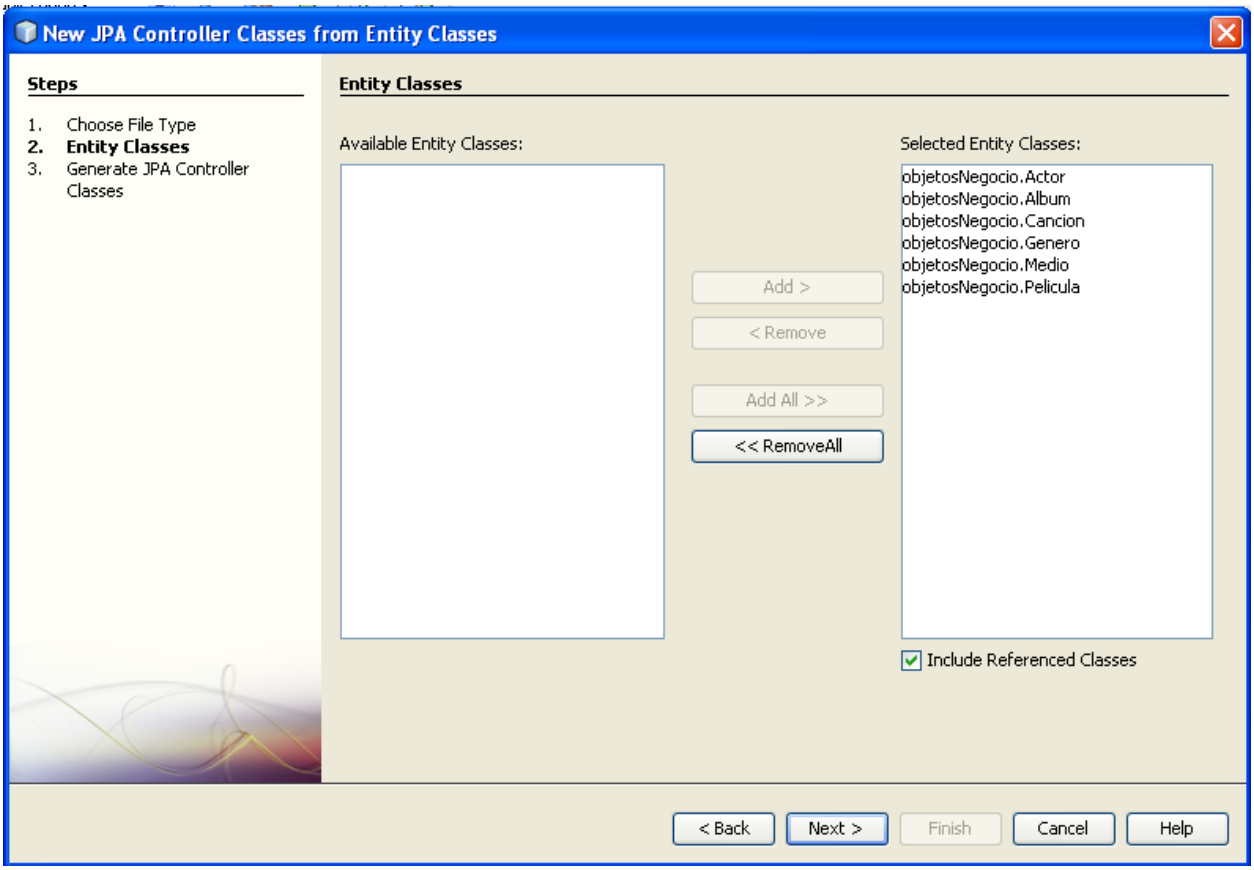

Figura 22

5. Aparecerá el cuadro de diálogo de la figura 23. Aquí estableceremos la ubicación de las clases controladoras JPA y otras clases relacionadas con éstas.

6. Establezca el nombre del paquete en el que se almacenarán las clases controladoras JPA a `persistencia`. A continuación presione el botón **Finish** NetBeans generará las clases controladoras JPA como se pueden ver en el árbol del proyecto, figura 24 y también varias clases de excepciones.

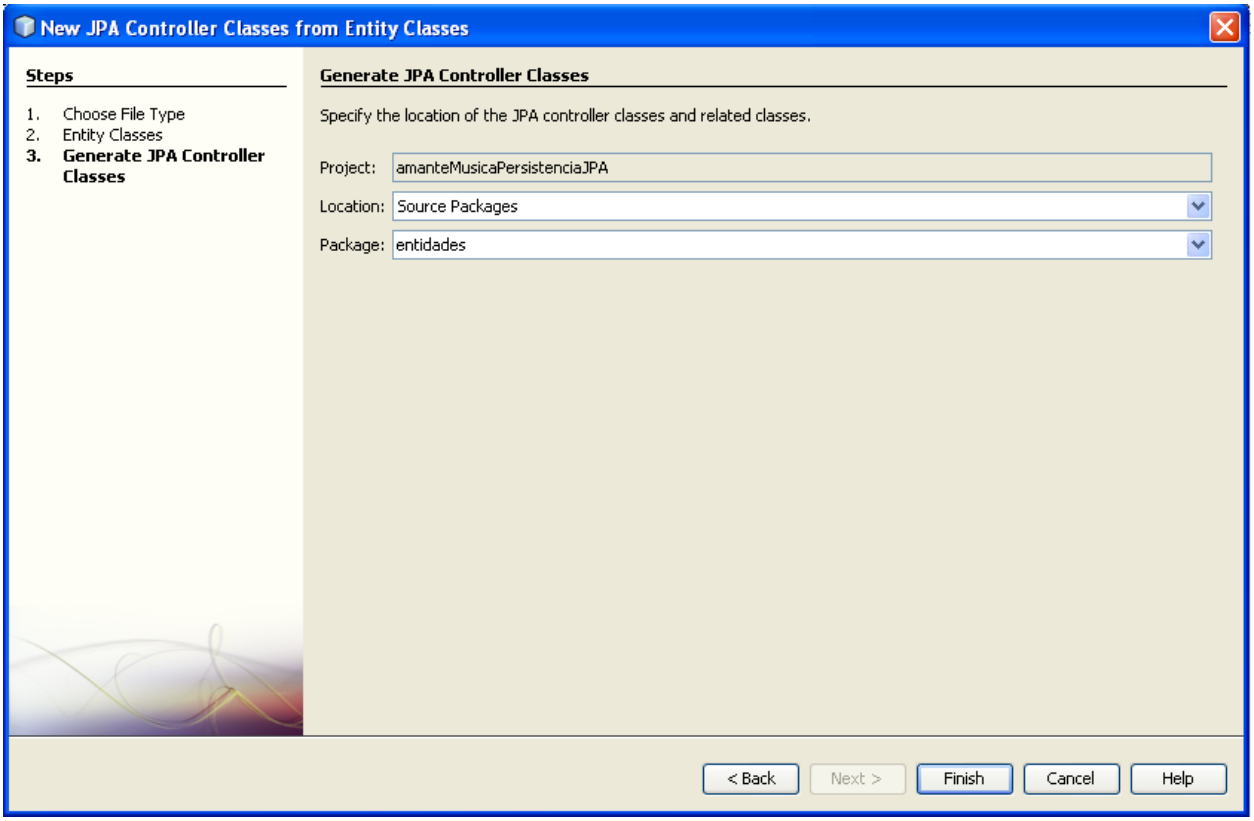

Figura 23

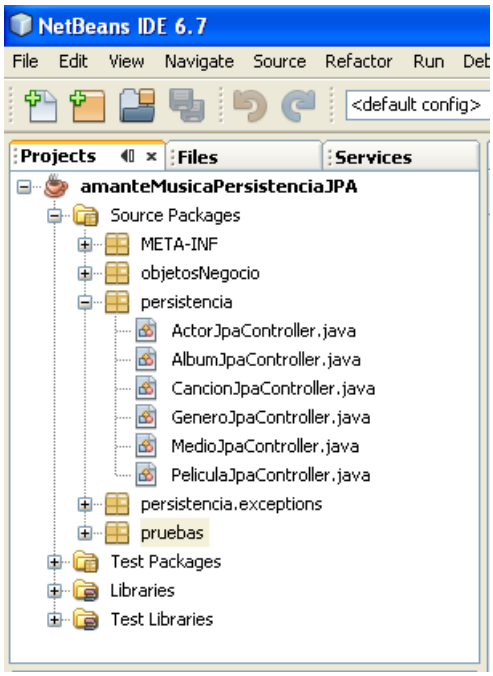

Figura 24