

Programas de Consola en Java con NetBeans

Creación de un Proyecto

Para crear un programa de consola en Java utilizando **NetBeans** lo primero que hay que hacer es crear un proyecto. Un proyecto nos permite administrar los archivos con el código fuente y compilado de una aplicación. Para crear un proyecto se sigue el siguiente procedimiento:

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

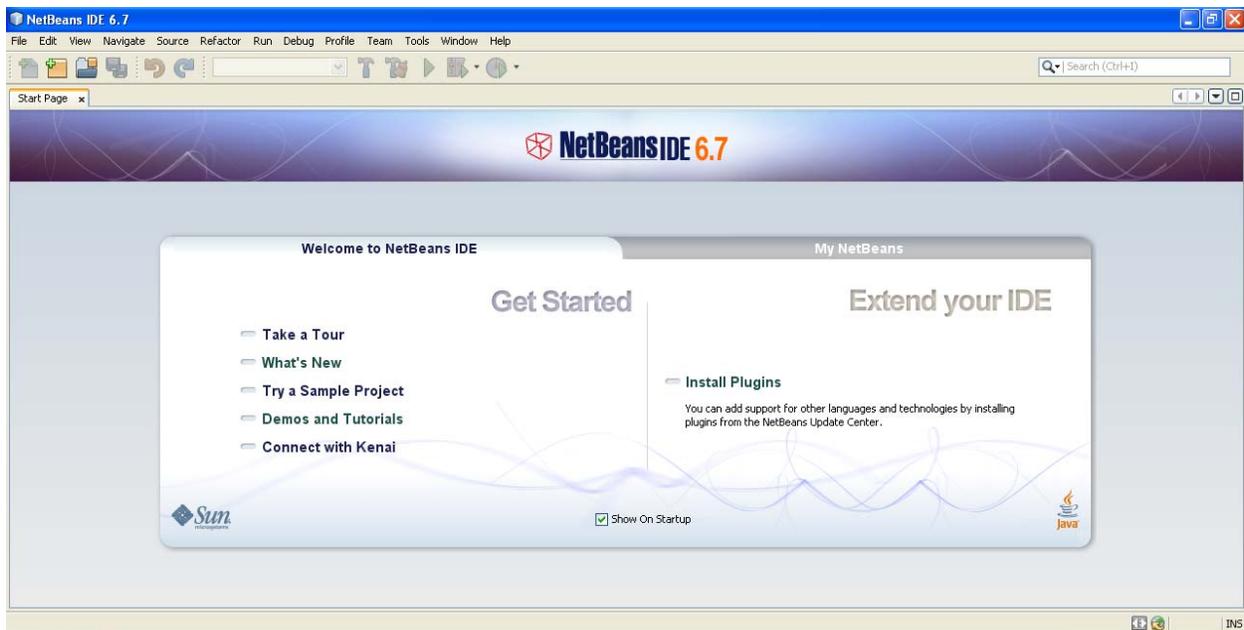


Figura 1

2. Inicialmente, en la ventana de NetBeans, aparece una página que contiene ligas a noticias y tutoriales. Podemos cerrar esa página haciendo clic en el icono **X** de la pestaña **Start Page** que se encuentra en la esquina superior izquierda de la página, figura 2. Podemos dejar que se página se muestre cada vez que ejecutemos a NetBeans o hacer que no se vuelva a mostrar. Si deseamos hacer esto último, deseleccionaremos la casilla de verificación **Show On Startup** que se encuentra en la parte inferior de la página, figura 3.

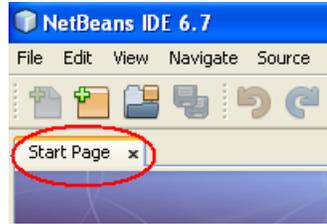


Figura 2

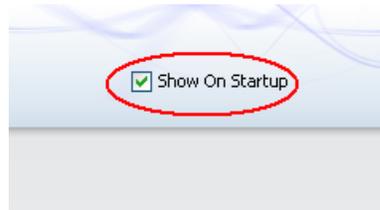


Figura 3

3. Deseleccionaremos la casilla de verificación **Show On Startup** para que la página inicial no vuelva a aparecer cada vez que ejecutemos a NetBeans y cerraremos la página inicial haciendo clic en el icono X de la pestaña **Start Page**. Aparecerá lo mostrado en la figura 4.

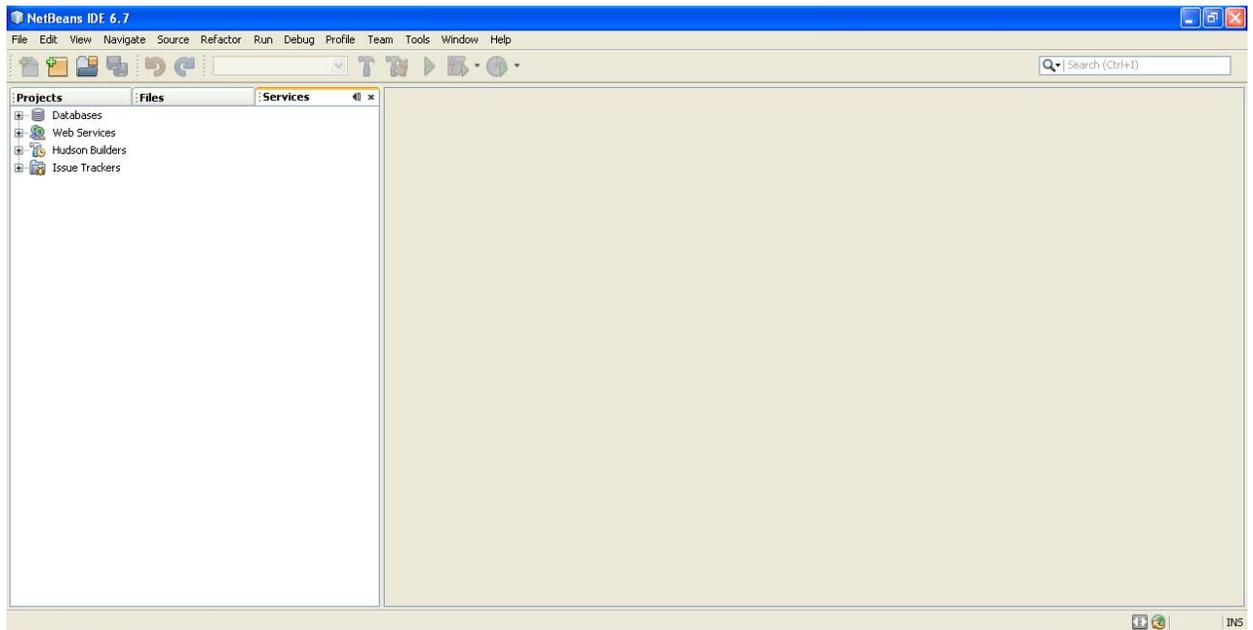


Figura 4

4. 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 5.



Figura 5

5. Aparece el primer cuadro de diálogo del asistente para crear un nuevo proyecto, figura 6.

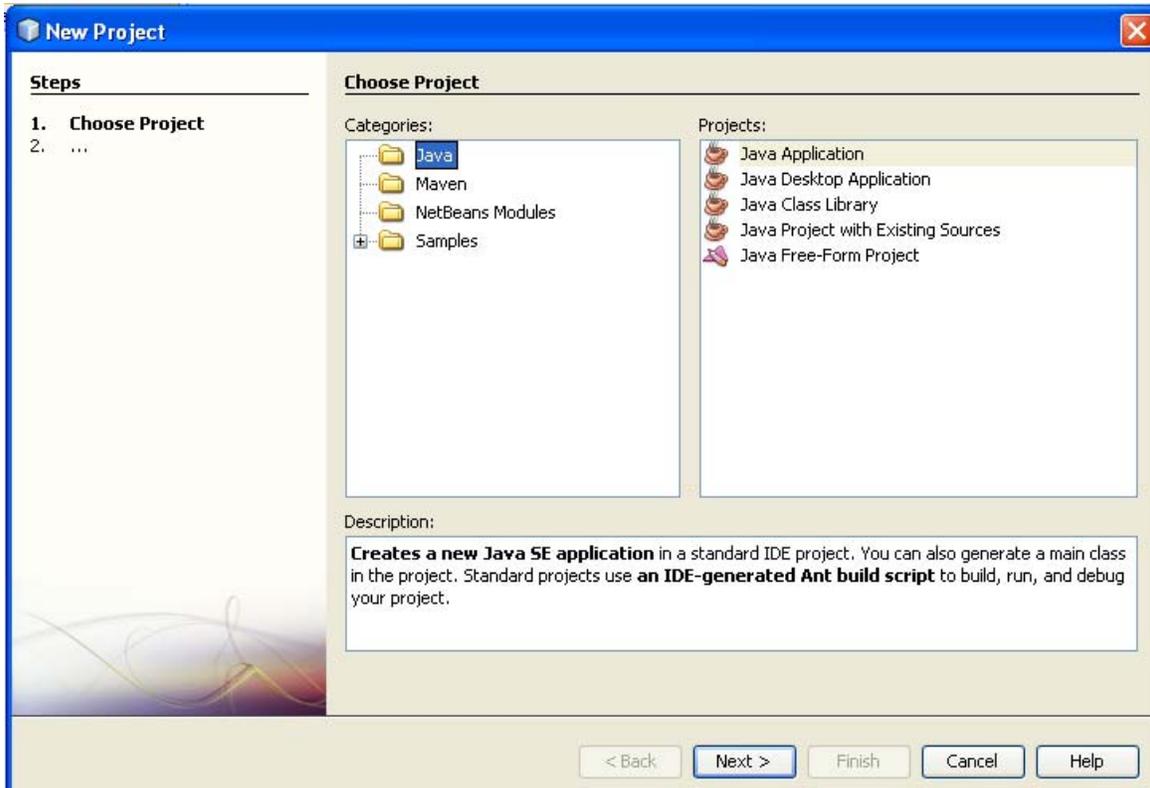


Figura 6

6. 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>**.
7. Aparecerá la segunda ventana del asistente para crear proyectos, figura 7. En esta ventana seleccionaremos el nombre y la ubicación del proyecto.
 - a) Establezca el nombre del proyecto (**Project Name**): Por ejemplo, "amanteMusicaObjNeg".

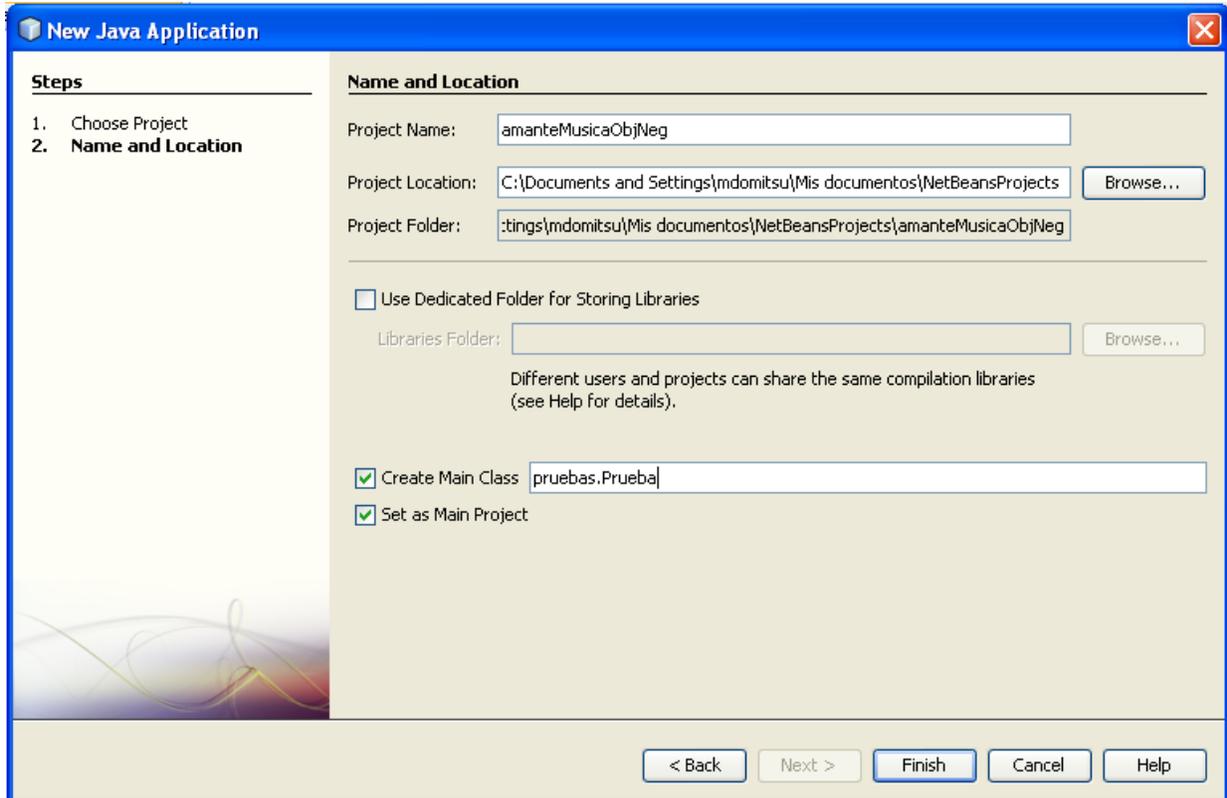


Figura 7

- b) Establezca el directorio donde se almacenará el proyecto (**Project 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**.
8. Desaparecerá el asistente para crear un nuevo proyecto y aparecerá lo mostrado en la figura 8. 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**.

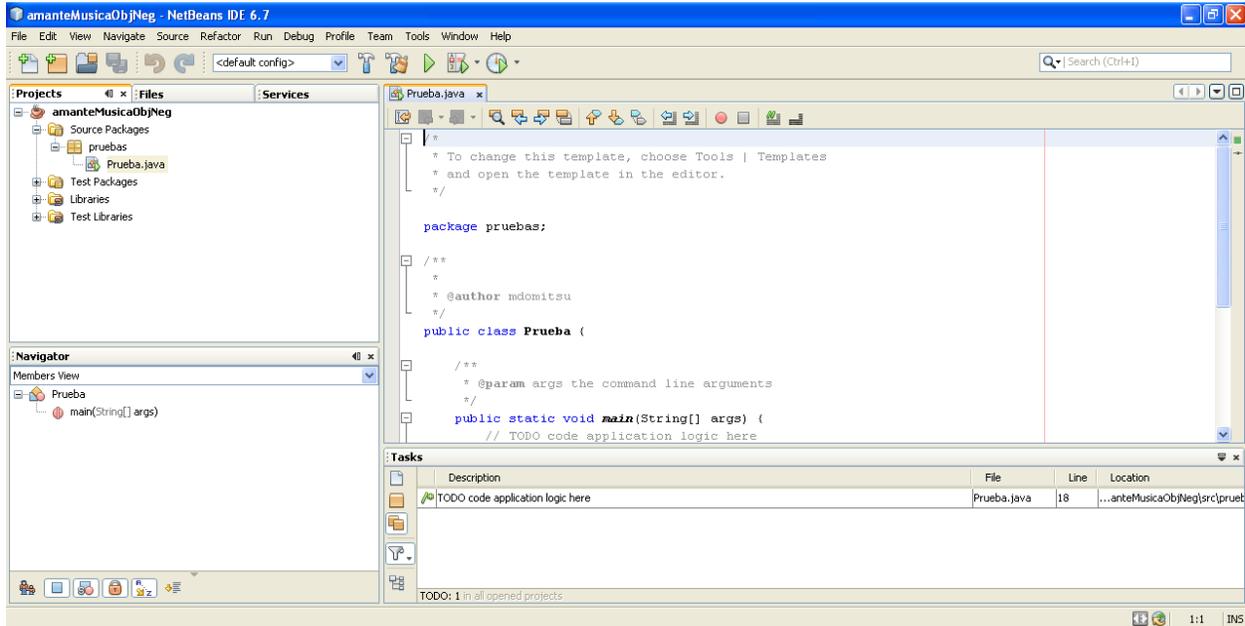


Figura 8

- Si en el recuadro del árbol de los proyectos hacemos clic en la pestaña Files, aparecerá un árbol con todos los archivos de los proyectos, figura 9. Otra vez, en este momento sólo aparecen los archivos del proyecto **amanteMusicaObjNeg**.

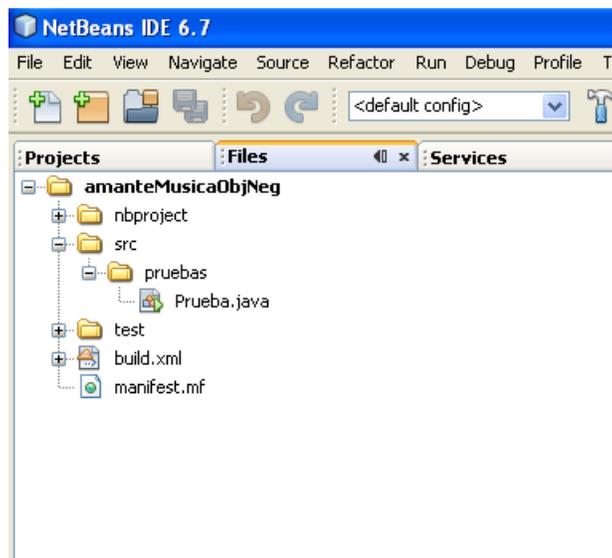


Figura 9

Ubicación de los Archivos de un Proyecto

La figura 10, muestra los directorios y archivos generados al crear el proyecto. El código fuente de una clase se guarda en un archivo que tiene el mismo nombre de la clase y con la extensión “.java”, por ejemplo, `Prueba.java` está dentro de la carpeta `pruebas` (el nombre del paquete) que a su vez está dentro de la carpeta “`src`” (donde se almacenan los archivos fuentes de este proyecto) y que a su vez está dentro de la carpeta “`AmanteMusicaObjNeg`” (que contiene todos los archivos del proyecto).

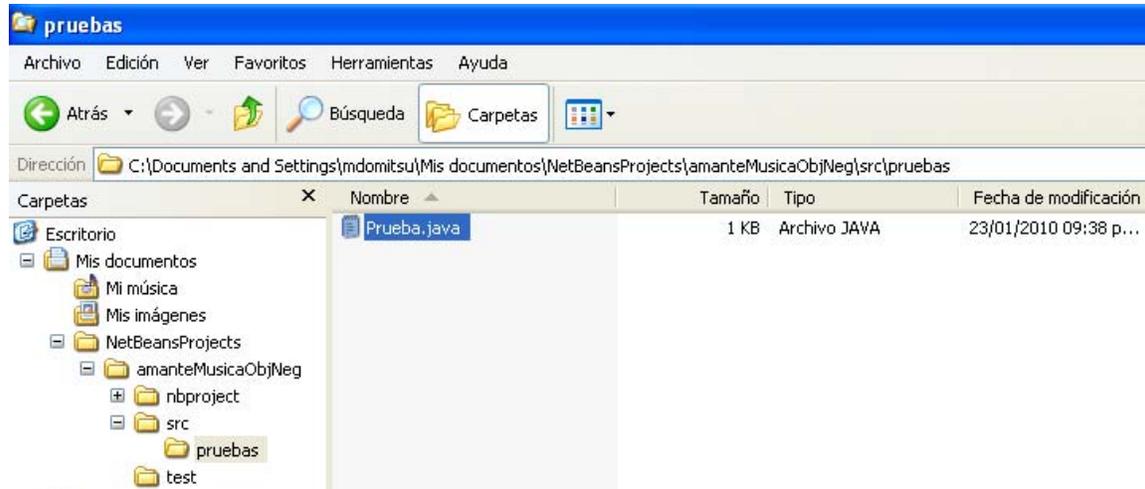


Figura 10

Creación de una Clase

Para crear una clase se sigue el siguiente procedimiento:

1. De la barra de menú de NetBeans, seleccione la opción **Files/New File**, presione las teclas **Ctrl+ N** o haga clic en el icono **New File**, como se muestra en la figura 11:

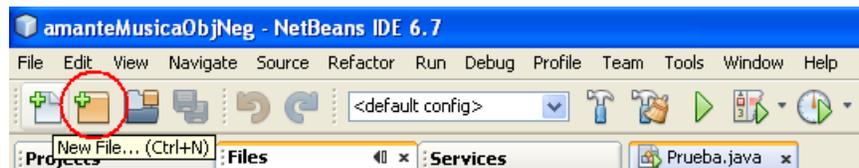


Figura 11

2. Aparecerá la primera ventana del asistente para crear una nueva clase, figura 12. Aquí seleccionaremos el tipo de clase que deseamos crear. Seleccionaremos la opción **Java Classes** en el recuadro **Categories:** y la opción **Java Class** en el recuadro **File Types:**, y luego presionaremos el botón **Next>**.

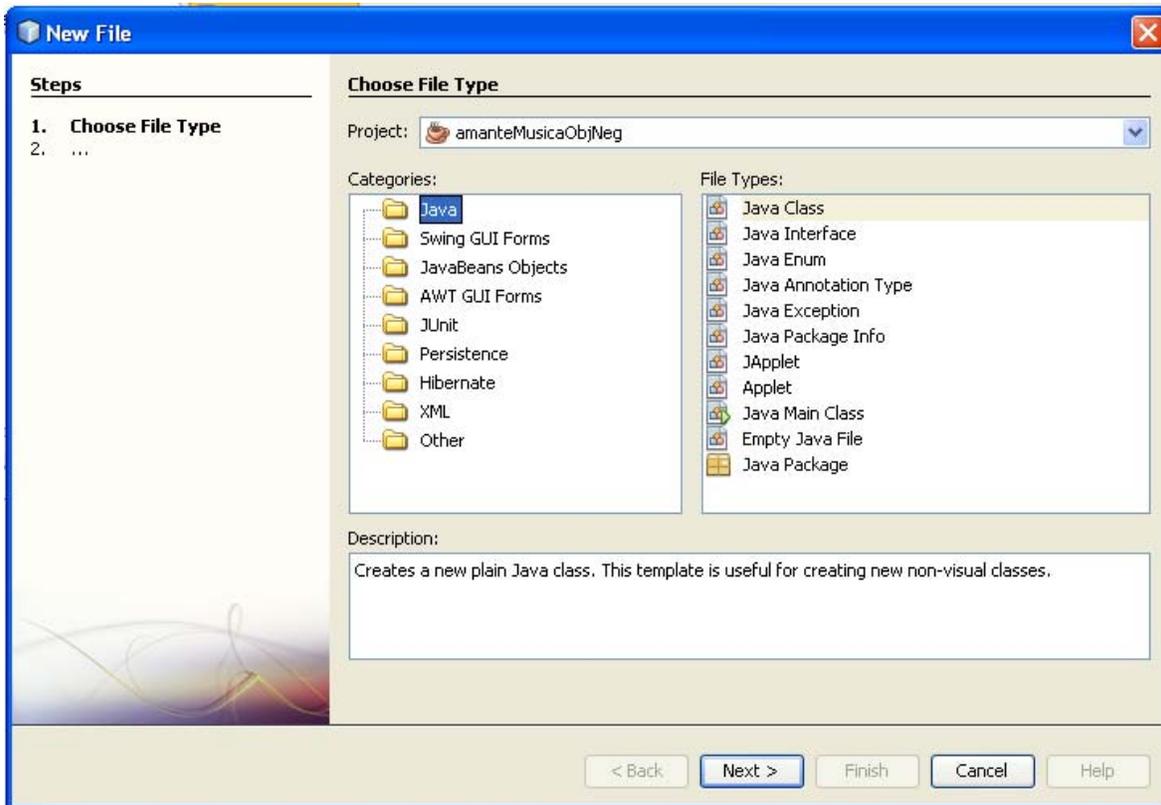


Figura 12

3. Aparecerá la segunda ventana del asistente para crear clases, mostrada en la figura 13. En esta ventana seleccionaremos el nombre y la ubicación de la clase.
 - a) Establezca el nombre de la clase (**Class Name**): Por ejemplo, “**Genero**”.
 - b) Establezca el paquete donde estará la clase (**Package**). Por ejemplo, “**objetosNegocio**”
 - c) Presione el botón **Finish**.
4. Desaparecerá el asistente para crear una nueva clase y aparecerá el esqueleto de la clase creada, 14.

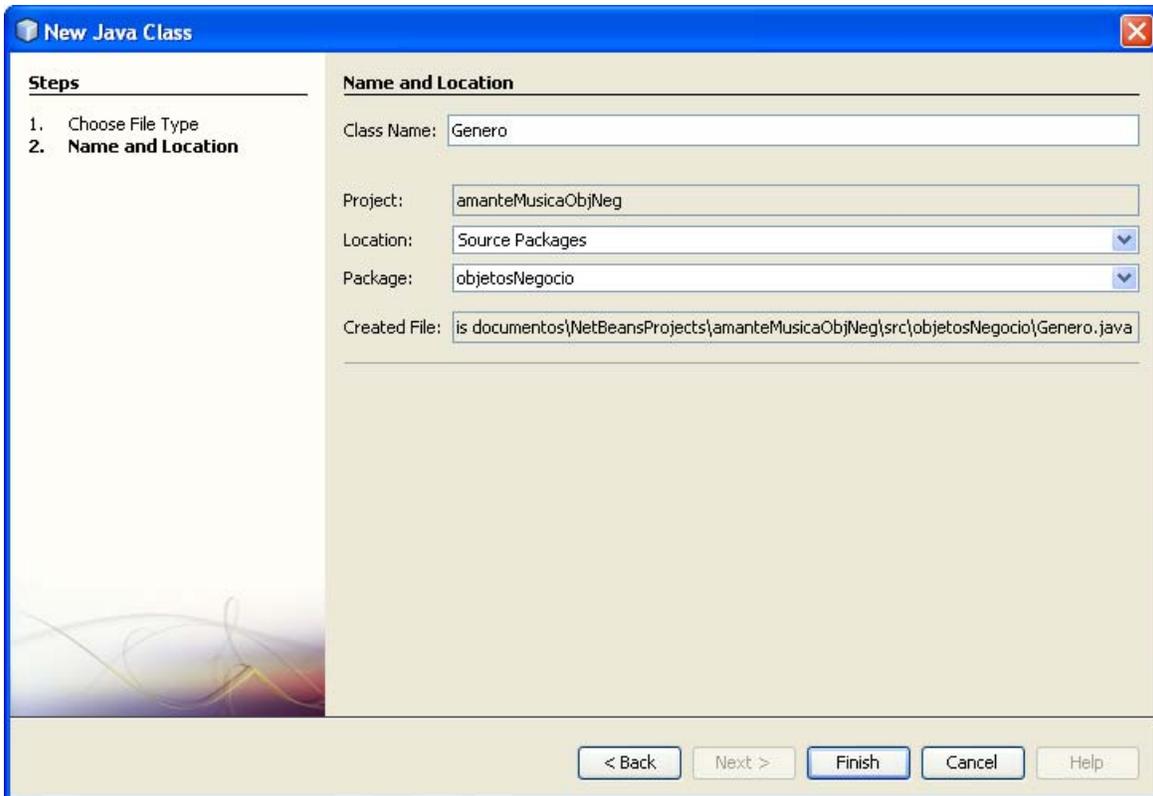


Figura 13

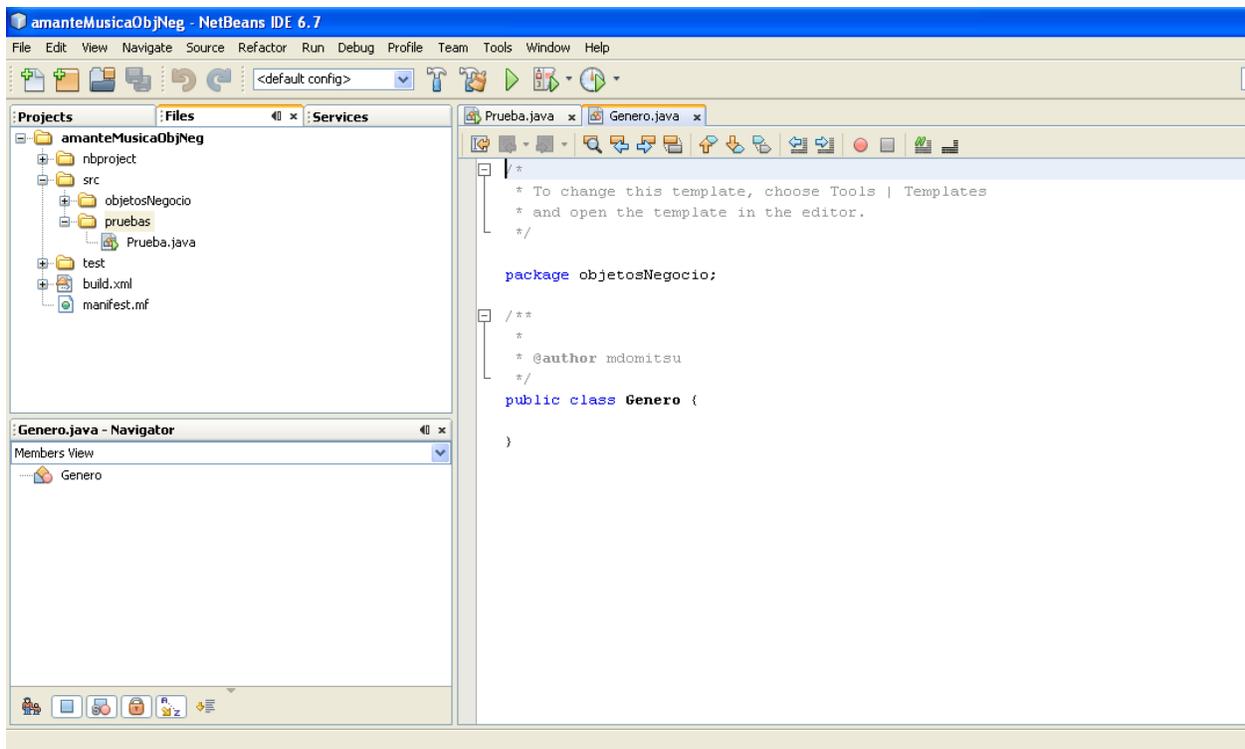


Figura 14

5. Agregue el código de los atributos, constructores y método `toString()` siguientes:

Genero.java

```
/*
 * Genero.java
 *
 * Creada el 28 de julio de 2008, 07:18 PM
 */

package objetosNegocio;

/**
 * Esta clase representa el género de una canción o película
 *
 * @author mdomitsu
 */
public class Genero {
    private String cveGenero;
    private String nombre;
    private char tipoMedio;

    public Genero() {
    }

    public Genero(String cveGenero, String nombre, char tipoMedio) {
        this.cveGenero = cveGenero;
        this.nombre = nombre;
        this.tipoMedio = tipoMedio;
    }

    public Genero(String cveGenero) {
        this(cveGenero, null, ' ');
    }

    public String toString() {
        return cveGenero + ", " + nombre + ", " + tipoMedio;
    }
}
```

6. Guarde la clase seleccionando del menú principal la opción **File/Save**, presione las teclas **Ctrl+S** o haga clic en el icono **Save All**, mostrado en la figura 15.

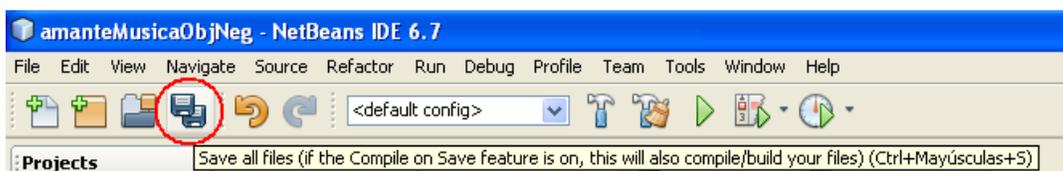


Figura 15

Generación de los Métodos de Acceso

NetBeans permite la generación automática de los métodos de acceso (**getXxx()** y **setXxx()**) que nos permiten inspeccionar y modificar el valor de los atributos de una clase. Para generar los métodos de acceso de la clase que se encuentra en la ventana de edición se sigue el siguiente procedimiento:

1. Coloque el cursor en el lugar en que desee insertar los métodos de acceso y haga clic con el botón derecho del ratón. Aparecerá el menú emergente de la figura 16. Seleccione la opción **Insert Code...** Otra forma de hacer lo mismo es presionando las teclas **Alt + Insertar**. Aparecerá el menú emergente de la figura 17.

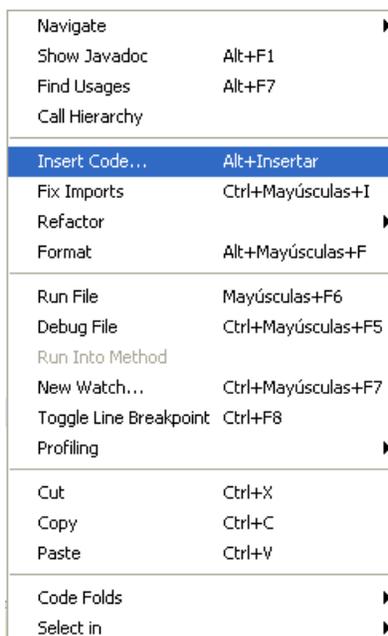


Figura 16

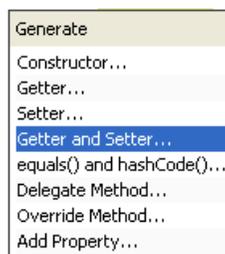


Figura 17

2. Seleccione la opción **Getter and Setter...** Aparecerá el cuadro de diálogo mostrado en la figura 18. Seleccione la casilla con el nodo **Genero** para seleccionar generar los métodos de acceso de todos los atributos (o seleccione

sólo las casillas de los atributos de los que desea generar los métodos de acceso y haga clic en el botón **Generate**.

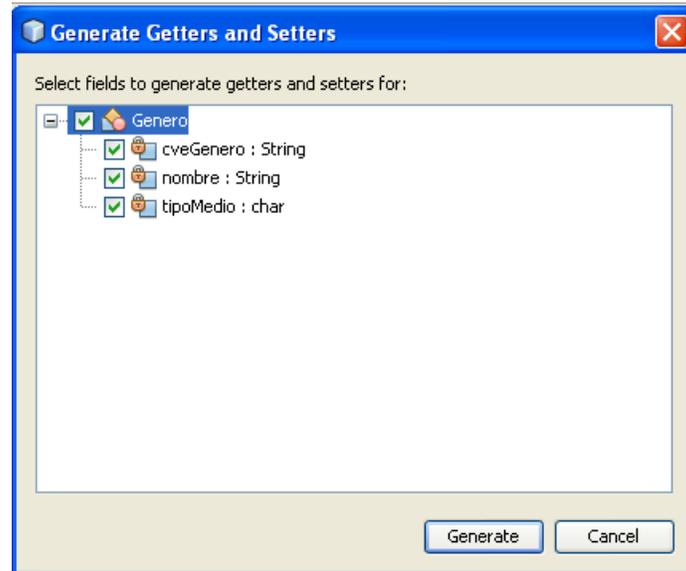


Figura 18

3. NetBeans genera los métodos de acceso seleccionados, como se muestra en la figura 19.

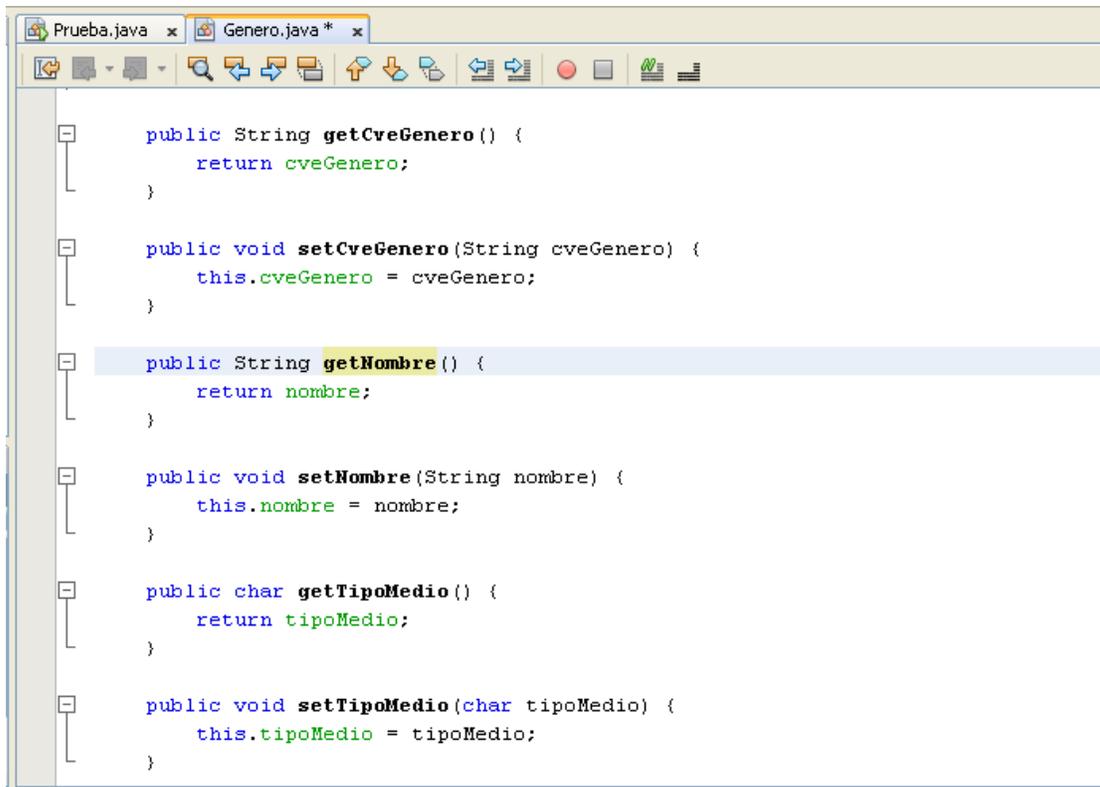


Figura 19

Generación de los equals() y hashCode()

NetBeans permite la generación automática de los métodos **equals()** y **hashCode()** que nos permiten comparar el objeto de la clase con otro objeto para determinar si son iguales bajo un criterio de comparación y obtener el código hash del objeto de la clase. Para generar los métodos **equals()** y **hashCode()** de la clase que se encuentra en la ventana de edición se sigue el siguiente procedimiento:

1. Coloque el cursor en el lugar en que desee insertar los métodos **equals()** y **hashCode()** y haga clic con el botón derecho del ratón. Aparecerá el menú emergente de la figura 16. Seleccione la opción **Insert Code...** Otra forma de hacer lo mismo es presionando las teclas **Alt + Insertar**. Aparecerá el menú emergente de la figura 20.

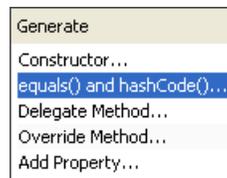


Figura 20

2. Seleccione la opción **equals() and hashCode()...** Aparecerá el cuadro de diálogo mostrado en la figura 21. Seleccione las casillas con los atributos que van a usarse para comparar los objetos y generar el código hash. En este caso seleccione sólo la casilla **cveGenero** tanto para el método **equals()** como **hashCode()**.

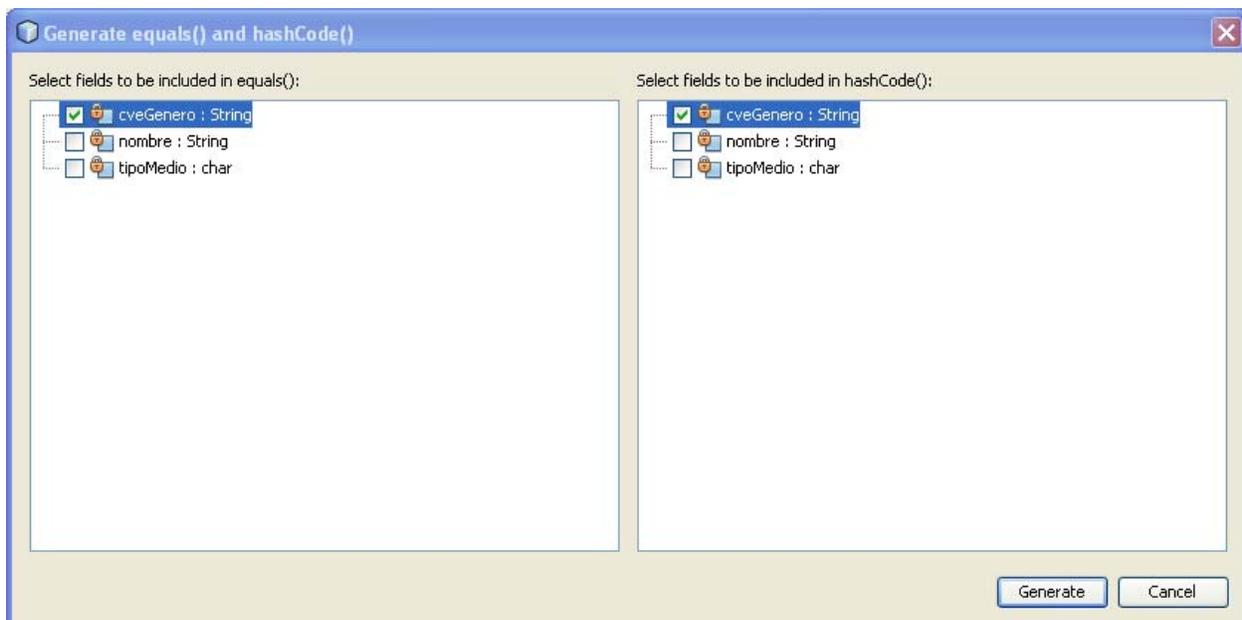


Figura 21

3. NetBeans genera los métodos equals() y hashCode(), como se muestra en la figura 22.

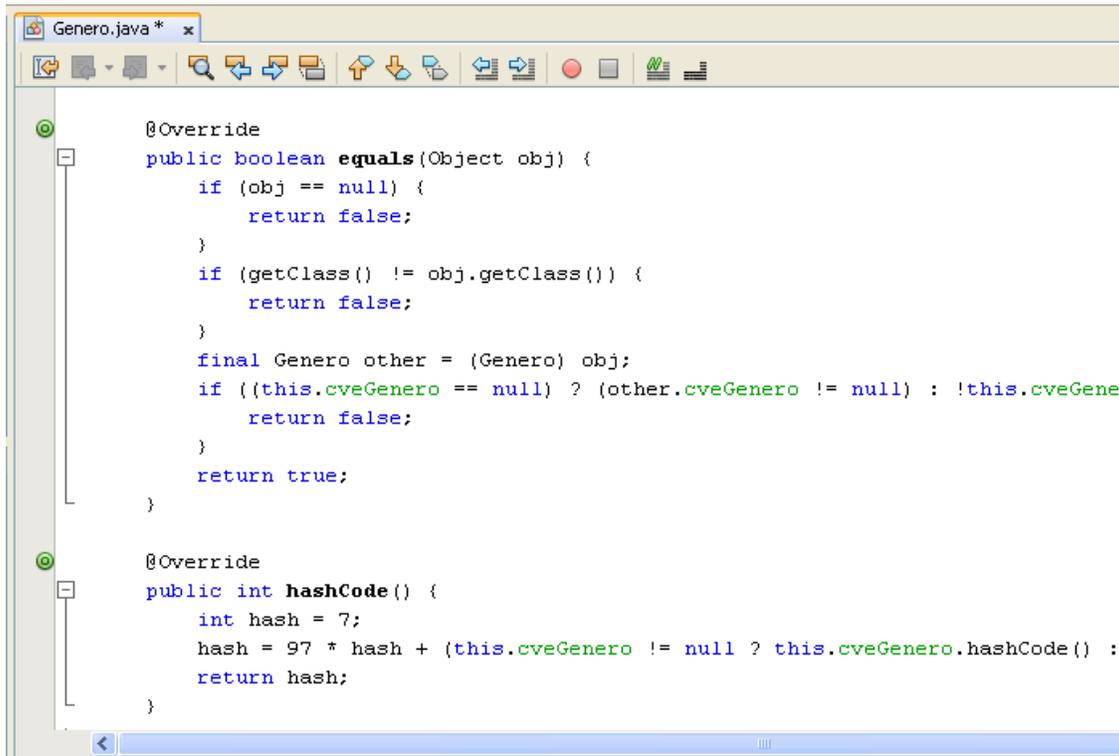


Figura 22

Edición de Múltiples Clases

Podemos editar más de una clase en la ventana de edición, cada una en su propio panel, como se muestra en la figura 23 en la que tenemos en la ventana de edición de NetBeans dos clases: Prueba.java y Medio.java. Las pestañas en la parte superior nos permiten seleccionar la clase que se desea en la ventana de edición.

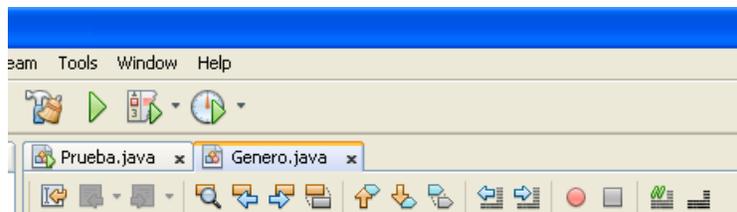


Figura 23

Podemos eliminar una clase de la ventana de edición, haciendo clic en el icono con la X que se encuentra en la pestaña de cada clase, figura 24.

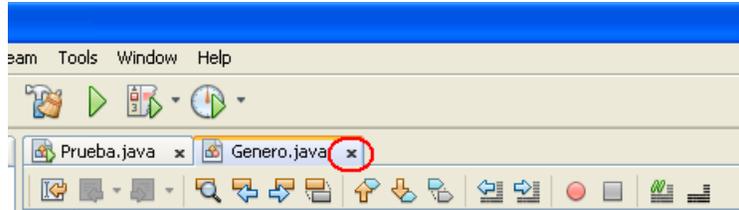


Figura 24

Si deseamos agregar una clase ya creada a la ventana de edición podemos hacer clic en su nombre en el árbol de archivos o en el árbol de proyectos que se encuentran a la izquierda, como se muestra en la figura 25.

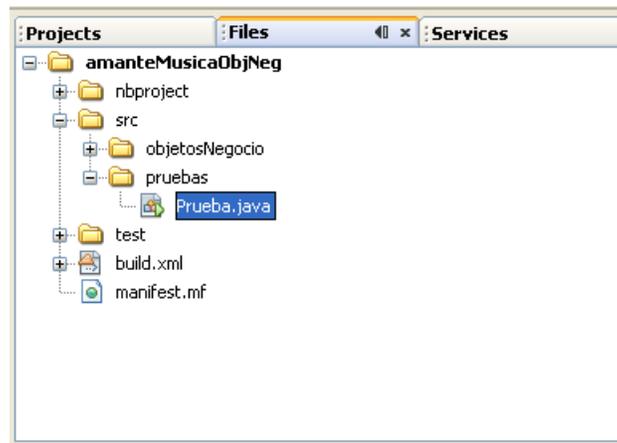


Figura 25

Edite la clase Prueba (la clase principal), agregándole el código siguiente:

Prueba.java

```

/*
 * Prueba.java
 *
 * Creada el 8 de septiembre de 2006, 12:21 PM
 */
package pruebas;

import objetosNegocio.Genero;

/**
 * Esta clase se utiliza para probar las clases del proyecto
 * amanteMusicaObjNeg
 *
 * @author mdomitsu
 */
public class Prueba {

    /**
     * Método main() en el que se invocan a los métodos de las clases
     * Genero, Medio, Cancion y Pelicula para probarlos
     * @param args Argumentos Los argumentos en la línea de comando
     */
}

```

```
public static void main(String[] args) {
    // Se crean tres géneros de canciones
    Genero genero1 = new Genero("GC001", "Balada", 'C');
    Genero genero2 = new Genero("GC002", "Bossanova", 'C');
    Genero genero3 = new Genero("GC003", "Rock", 'C');

    // Se crean tres géneros de películas
    Genero genero4 = new Genero("GP001", "Drama", 'P');
    Genero genero5 = new Genero("GP002", "Ciencia Ficción", 'P');
    Genero genero6 = new Genero("GP003", "Comedia", 'P');

    // Se despliegan los datos del género 1
    System.out.println("Género 1: " + genero1);
    // Se despliegan los datos del género 2
    System.out.println("Género 2: " + genero2);
    // Se despliegan los datos del género 3
    System.out.println("Género 3: " + genero3);
    // Se despliegan los datos del género 4
    System.out.println("Género 4: " + genero4);
    // Se despliegan los datos del género 5
    System.out.println("Género 5: " + genero5);
    // Se despliegan los datos del género 6
    System.out.println("Género 6: " + genero6);

    // Se modifica el nombre del genero genero2 a "Jazz"
    genero2.setNombre("Jazz");

    // Se despliegan, de nuevo, los datos del género 2
    System.out.println("Género 2: " + genero2);
}
}
```

Compilación de una Clase

Para compilar la clase que se encuentra en la ventana de edición seleccione del menú principal la opción **Run/Compile** “**NombreClase.java**” o presione la tecla **F9**, figura 26, “**NombreClase**” es el nombre de la clase a compilar, por ejemplo “**Genero**”.

Durante la compilación, NetBeans muestra los mensajes resultantes del proceso, como se muestra en la figura 27.

Compilación del Proyecto

Para compilar todas las clases de un proyecto seleccione del menú principal la opción **Run/Build Main Project**, figura 28, presione la tecla **F11** o presione el icono **Build Main Project**, mostrado en la figura 29. Durante la compilación, NetBeans muestra los mensajes resultantes del proceso, como se muestra en la figura 27.

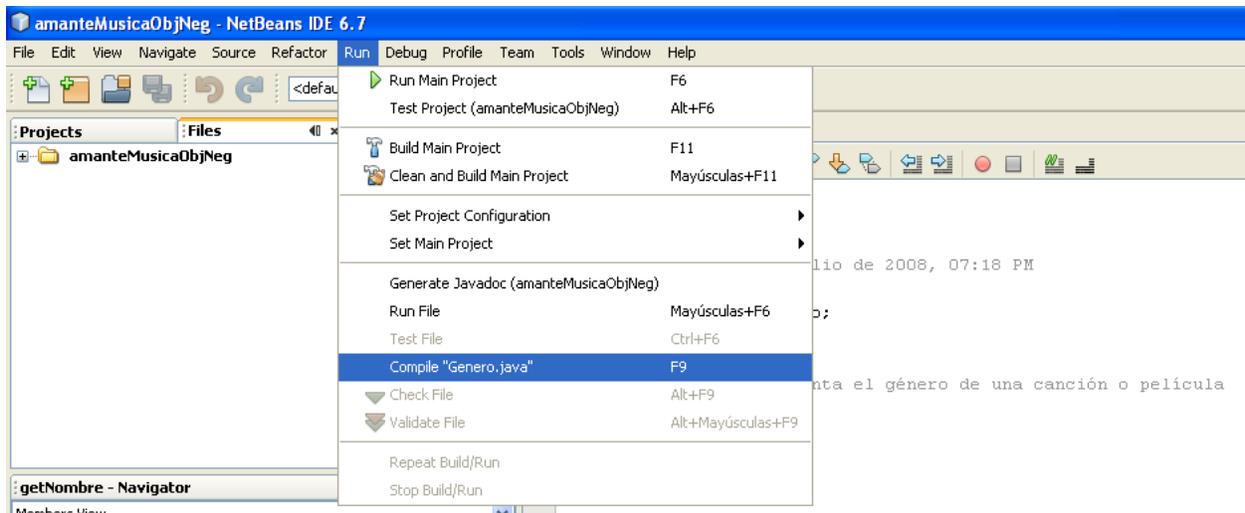


Figura 26

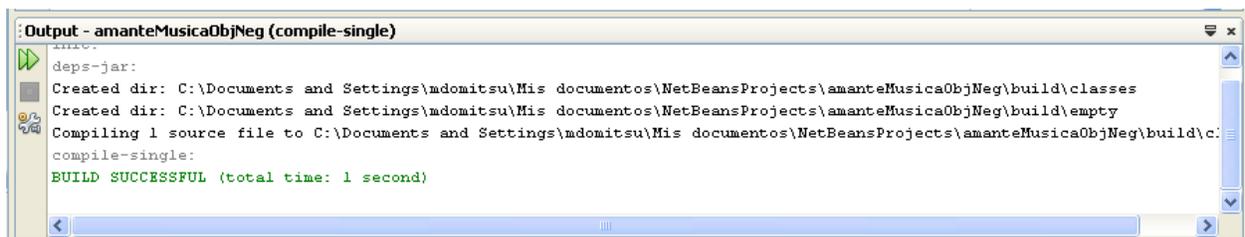


Figura 27

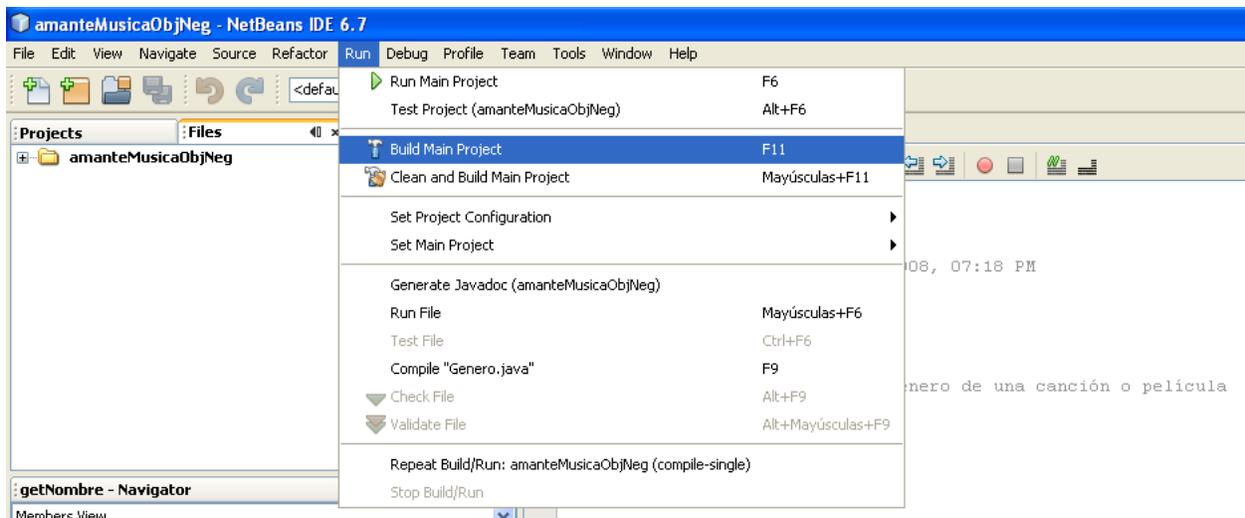


Figura 28

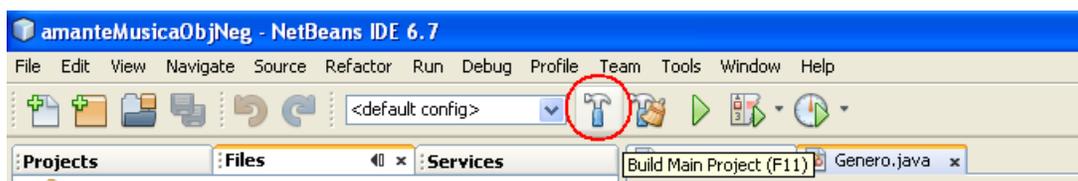


Figura 29

Ubicación de los Archivos con el Código “byteCode” de las Clases

La figura 30, muestra que el código “byteCode”, generado al compilar las clases, se guarda en archivos que tienen el mismo nombre de la clase y con la extensión “.class”: “Genero.class” dentro de la carpeta “objetosNegocio” (el nombre del paquete) que a su vez está dentro de la carpeta “classes” (donde se almacenan los archivos con el código “byteCode” de este proyecto) y que a su vez está dentro de la carpeta “amanteMusica” (que contiene todos los archivos del proyecto). El archivo “Prueba.class” está en la carpeta “amanteMusica\classes\pruebas”

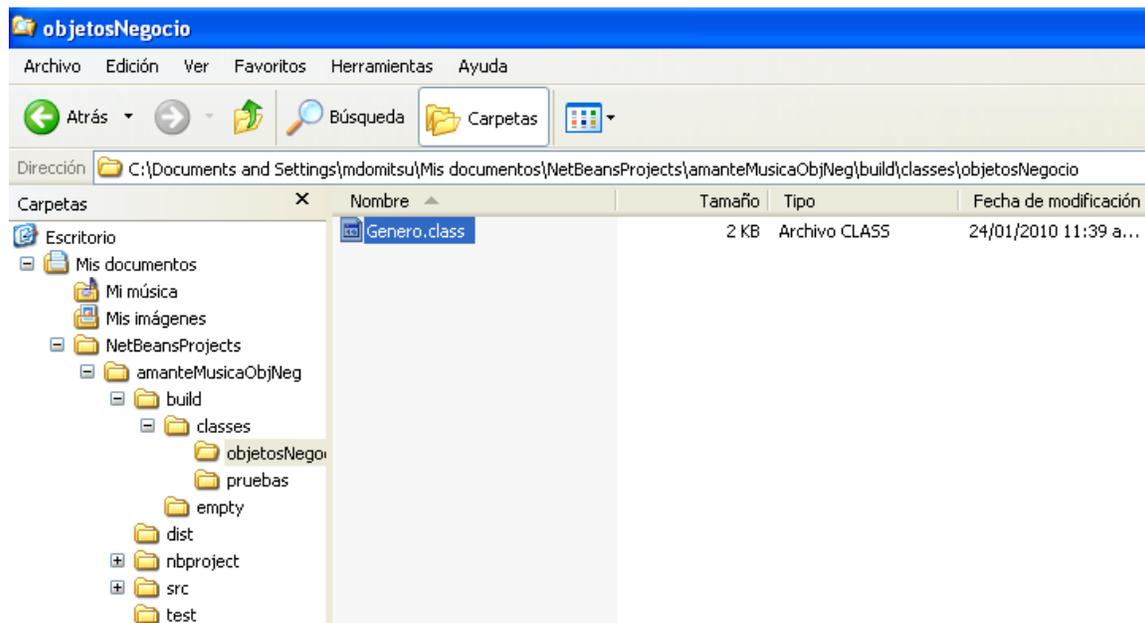


Figura 30

Cuando el proyecto tiene una clase principal (una con el método `main()`). NetBeans empaqueta los archivos con el código “byteCode” en un archivo con el nombre del proyecto y la extensión “.jar” y lo almacena en el directorio “dist” dentro del directorio del proyecto, figura 31.

Ejecución de una aplicación

1. Para ejecutar la aplicación dentro de NetBeans, seleccione del menú principal la opción **Run/Run Main Project**, figura 32, presione la tecla **F6** o haga clic en el icono **Run Main Project**, mostrado en la figura 33.

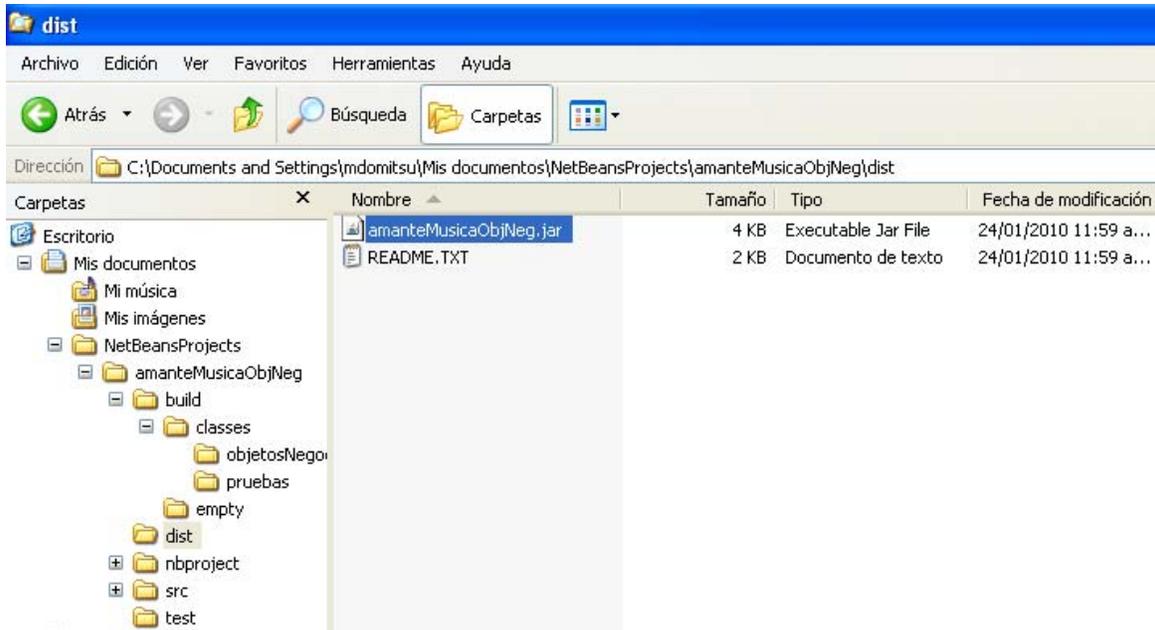


Figura 31

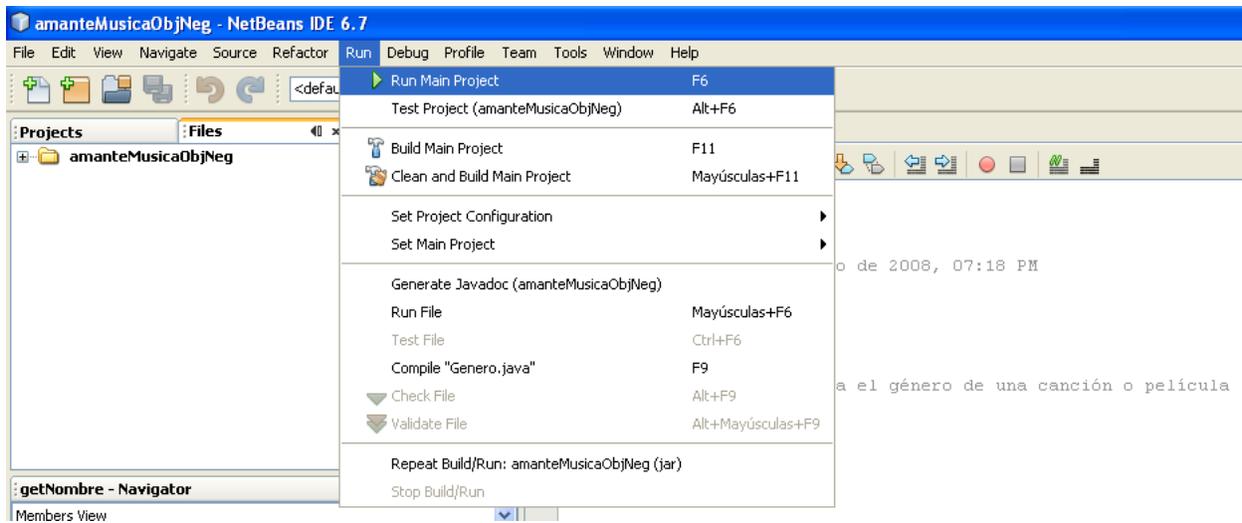


Figura 32



Figura 33

- Los resultados arrojados al ejecutar la aplicación se muestran la ventana: **Output**, en la parte inferior de la ventana de NetBeans, como se muestra en la figura 34.



Figura 34

- Para ejecutar una aplicación desde una consola del sistema, teclearemos en la línea de comando:

```
java -jar "ruta\nombreProyecto.jar"
```

donde *ruta\nombreProyecto.jar* es la ruta y nombre del archivo .jar con el código bytecode de la aplicación. Para este ejemplo *ruta\nombreProyecto.jar* es:

```
"C:\Documents and Settings\mdomitsu\Mis
documentos\NetBeansProjects\amanteMusicaObjNeg\dist\amanteMusicaObjN
eg.jar"
```

- Al hacerlo, veremos los mismos resultados observados en el paso 2, figura 35.

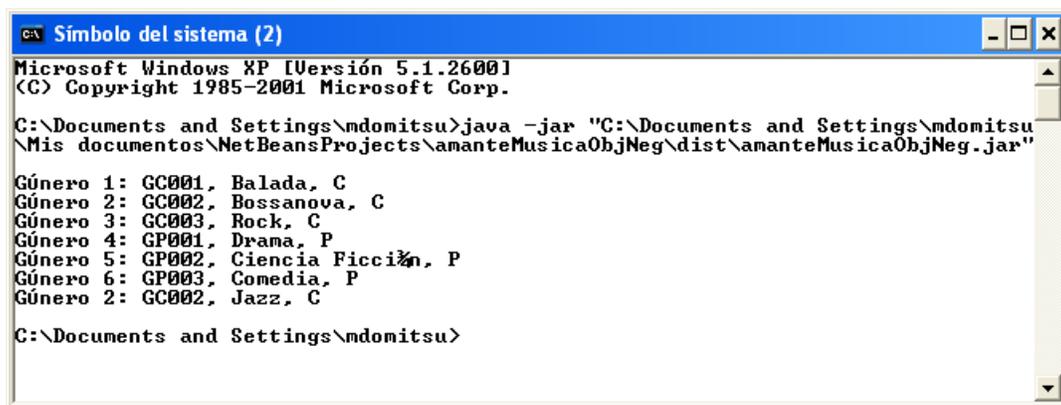


Figura 35

Cierre de un Proyecto

NetBeans nos permite cerrar un proyecto cuando ya no se desea trabajar con él. Para cerrar un proyecto en NetBeans seleccione de la barra de menú de NetBeans la opción **File/Close Project (*NombreProyecto*)**, donde ***NombreProyecto*** es el nombre del proyecto. Por ejemplo *amanteMusicaObjNeg*, figura 36.

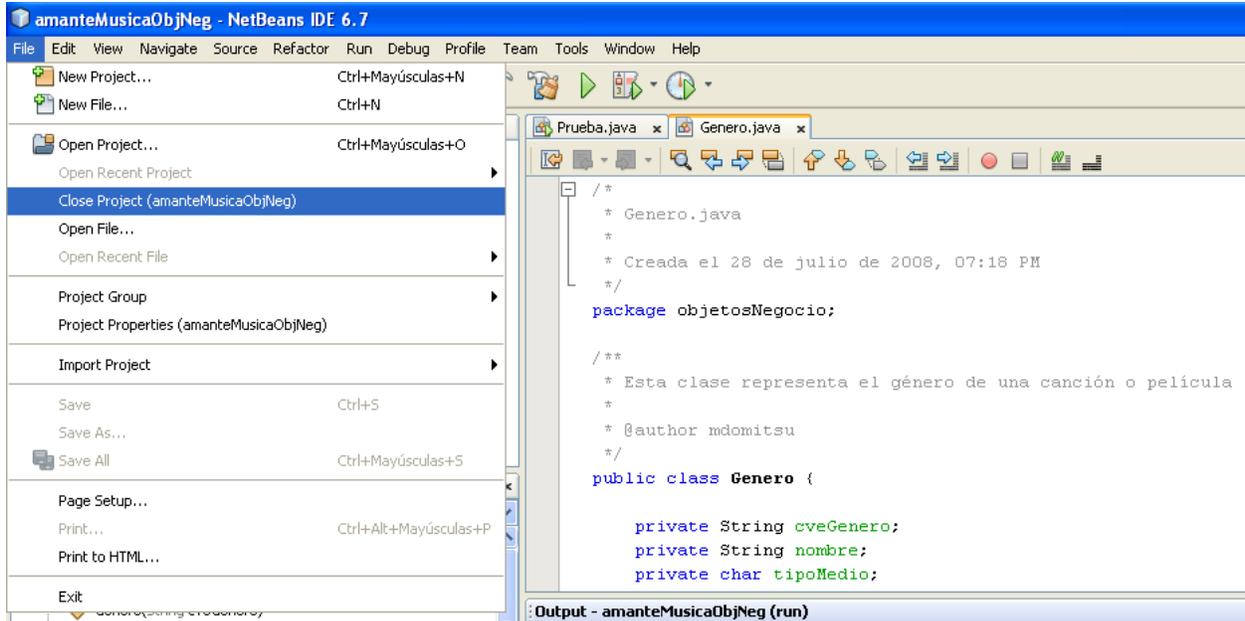


Figura 36

Otra forma de cerrar un proyecto es hacer clic con el botón derecho sobre el nodo del proyecto en el panel Projects y seleccionar la opción **Close**, figura 37.

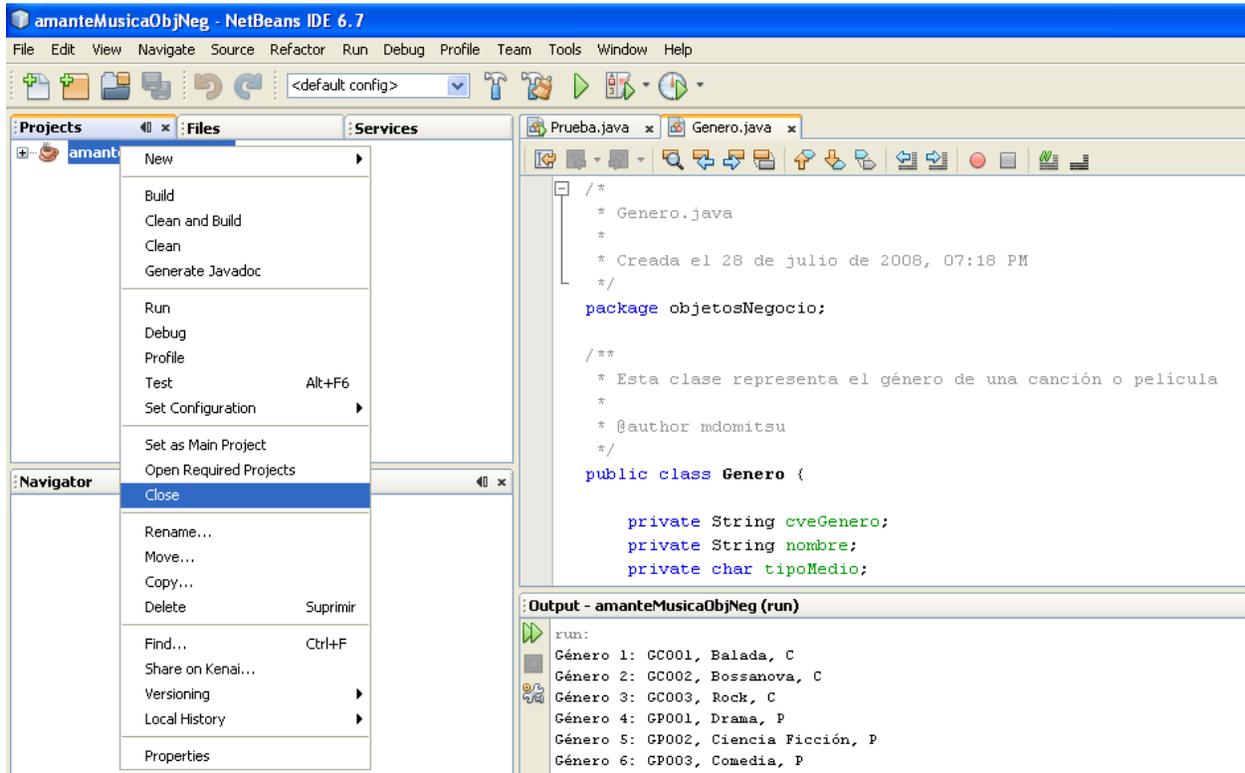


Figura 37

Apertura de un Proyecto

NetBeans nos permite abrir un proyecto existente para modificar una de sus clases o agregarle clases nuevas. Para abrir un proyecto en NetBeans se sigue el siguiente procedimiento:

1. Seleccione de la barra de menú de NetBeans la opción **File/Open Project ...**, figura 38, presione las teclas **Ctrl+Mayúsculas+O** o haga clic en el icono **Open Project**, mostrado en la figura 39.

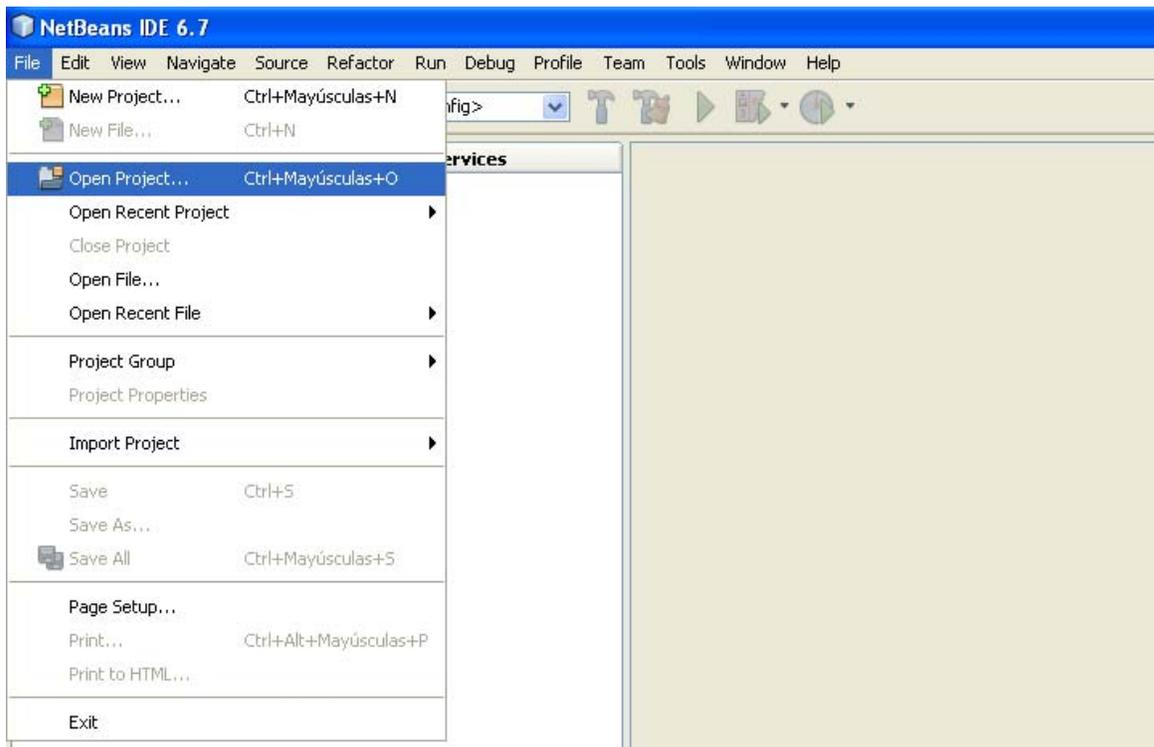


Figura 38

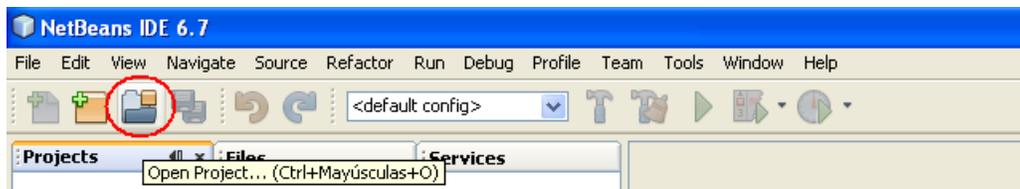


Figura 39

2. Al hacerlo NetBeans nos presenta un cuadro de diálogo que nos permite seleccionar el proyecto que deseamos abrir, figura 40. En la parte superior, hay una caja combinada para seleccionar la carpeta con los proyectos existentes, mismos que se muestran en el recuadro central (cada uno en su propia carpeta). Al seleccionar un proyecto, su nombre aparece del lado derecho. La casilla de selección: **Open as Main Project** (Abrir como proyecto principal) nos

permite establecer que el proyecto a abrir sea el proyecto principal, el que será compilado y ejecutado al seleccionar las opciones: **Build Main Project** (Compilar Proyecto Principal) y **Run Main Project** (Ejecutar Proyecto Principal). La casilla de selección: **Open Required Projects** (Abrir Proyectos Requeridos) sólo estará activa si el proyecto a abrir depende de otros proyectos, si se selecciona, hace que NetBeans abra el proyecto deseado además de abrir los proyectos de los que depende. Una vez seleccionadas las opciones deseadas se presiona el botón **Open Project Folder**.

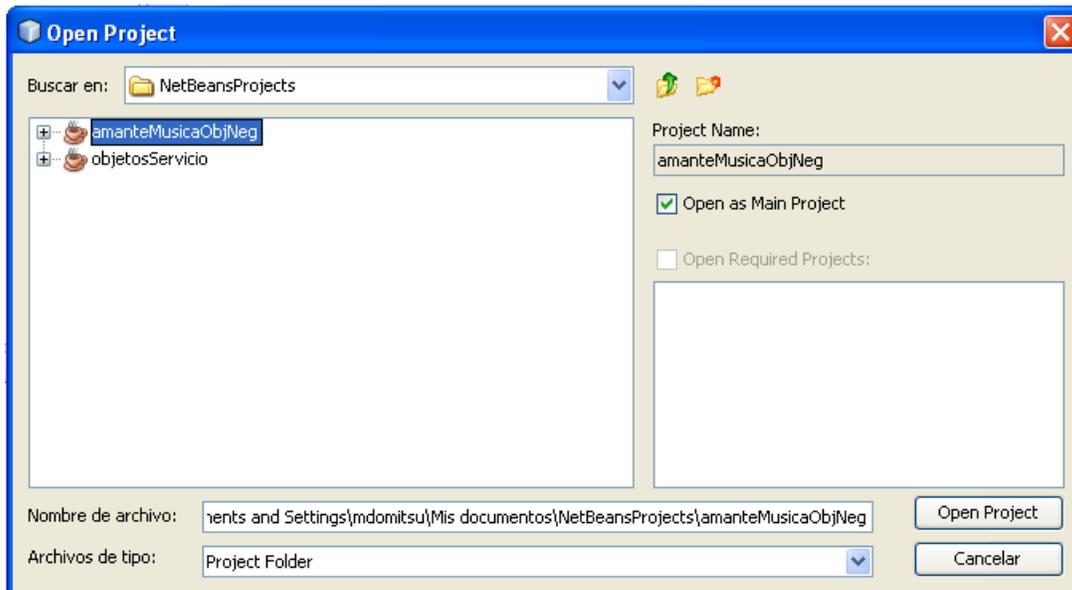


Figura 40

3. Al hacerlo, NetBeans abrirá el proyecto seleccionado, figura 41.

NetBeans y Múltiples Proyectos

NetBeans nos permite tener abiertos varios proyectos y crear, editar, compilar clases en ellos. Sólo uno de ellos será nominado el proyecto principal. Sin embargo, podemos cambiar la designación del proyecto principal entre los diferentes proyectos. Cada uno de los proyectos se abre siguiendo el procedimiento previamente visto, inclusive podemos seleccionar más de un proyecto a la vez para abrirlos al mismo tiempo, figura 42.

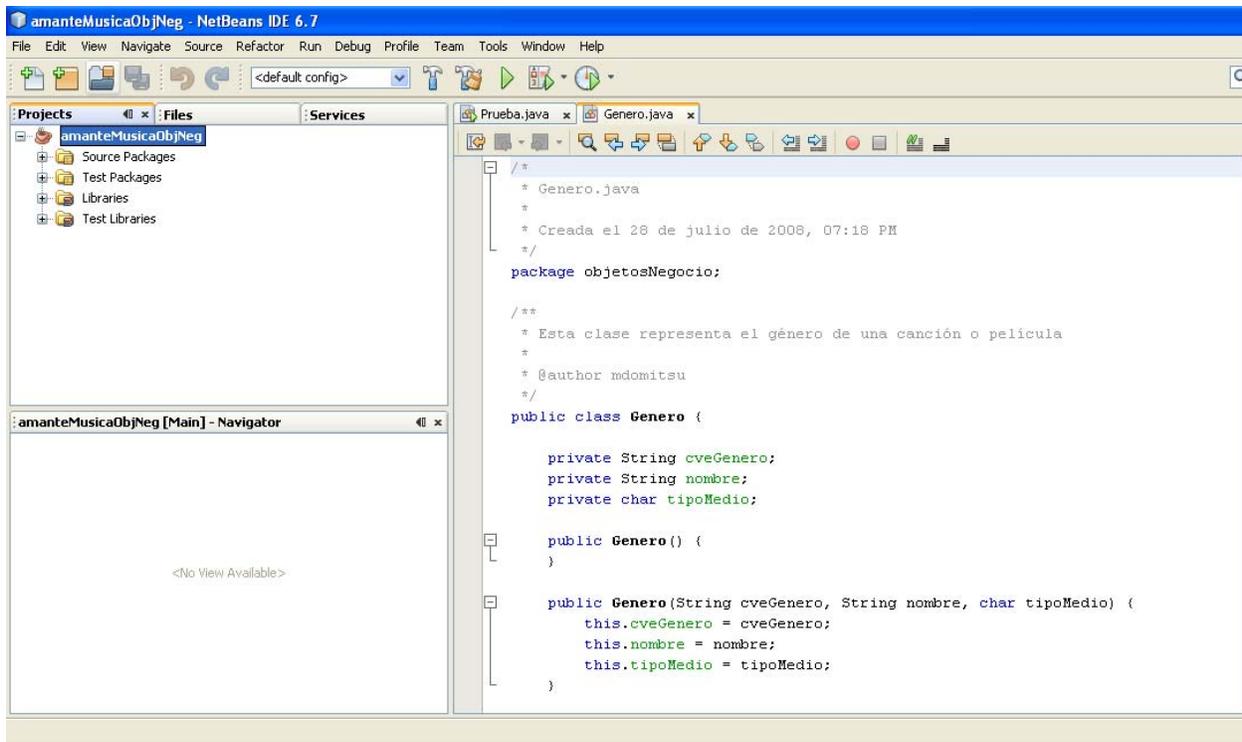


Figura 41

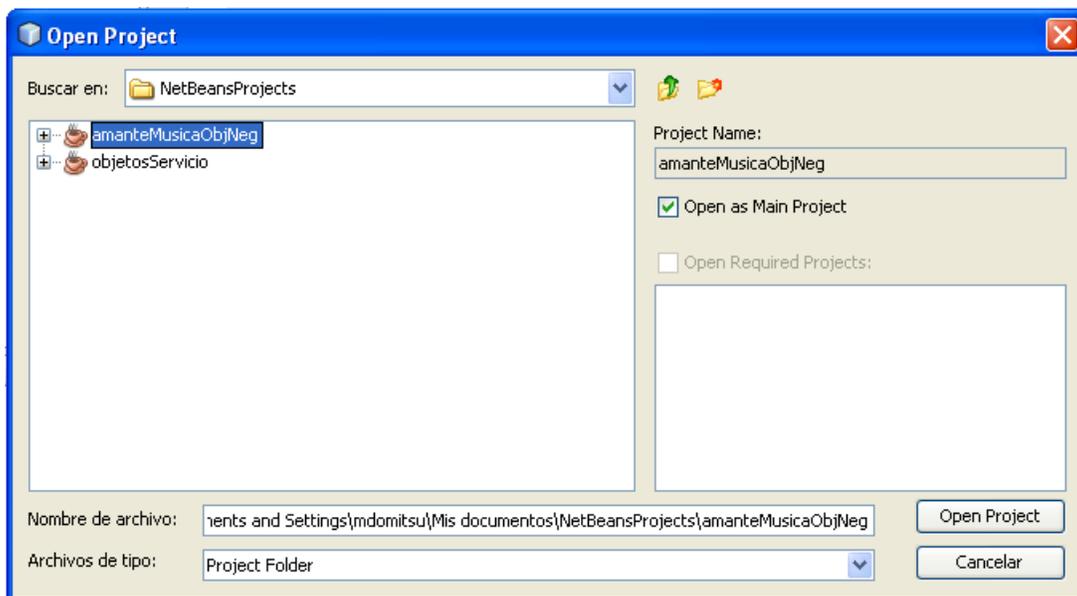


Figura 42

Los proyectos abiertos se muestran en el árbol de proyectos, figura 43.

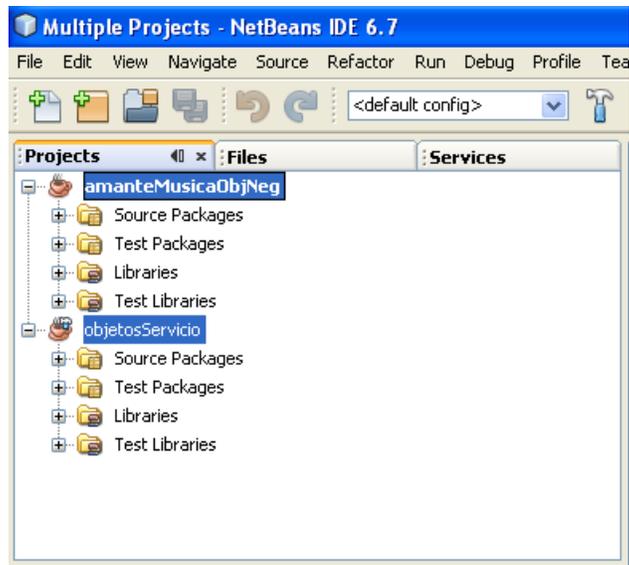


Figura 43

Para cambiar la designación de proyecto principal, de un proyecto a otro, hacemos clic con el botón derecho en el nodo del proyecto que deseamos que sea el proyecto principal y seleccionamos la opción **Set Main Project**, del menú emergente, figura 44.

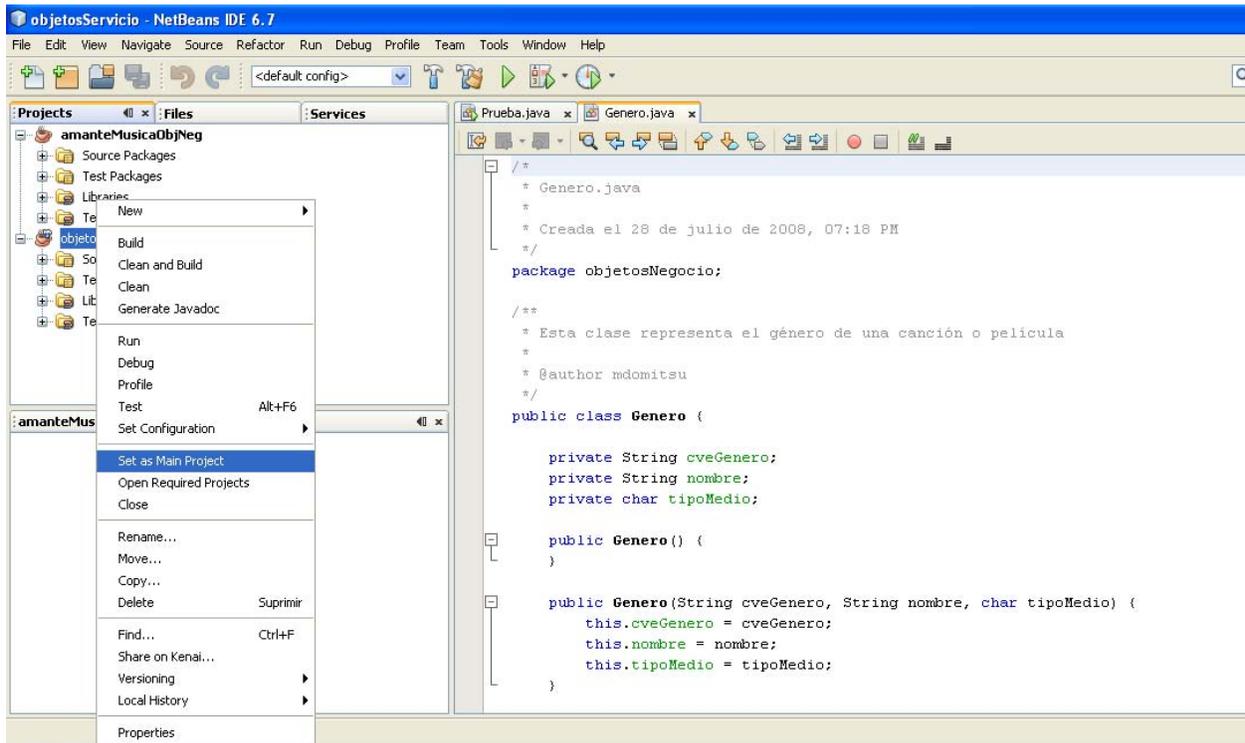


Figura 44

Dependencias entre Proyectos

Al desarrollar una aplicación grande con un gran número de paquetes y clases, es común separar el código en varios proyectos. Cada proyecto podría tener su propia clase principal, posiblemente para probar las clases de ese proyecto. Un proyecto puede depender de uno o más proyectos, esto es, utiliza las clases de esos proyectos.

NetBeans nos permite establecer que un proyecto depende de otro, dándonos acceso al código byteCode de las clases de éste último. El código byteCode se encuentra en el archivo de distribución o JAR (el archivo con extensión “.jar”) del otro proyecto. Recuerde que ese archivo JAR se genera al compilar el proyecto.

Para hacer que un proyecto dependa de otro se sigue el siguiente procedimiento:

1. En el árbol de proyectos expanda el nodo del proyecto dependiente y haga clic con el botón derecho en el nodo **Libraries** y seleccione la opción **Add Project ...**, del menú emergente, figura 45.

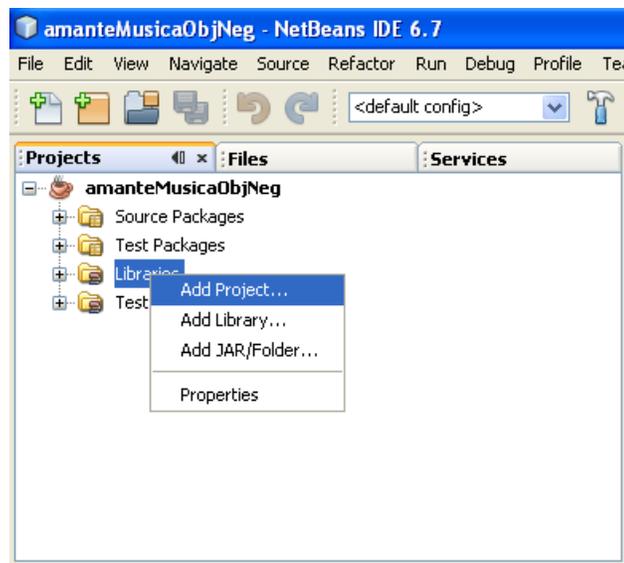


Figura 45

2. Aparecerá el cuadro de diálogo para seleccionar el proyecto del que dependerá nuestro proyecto, figura 46.

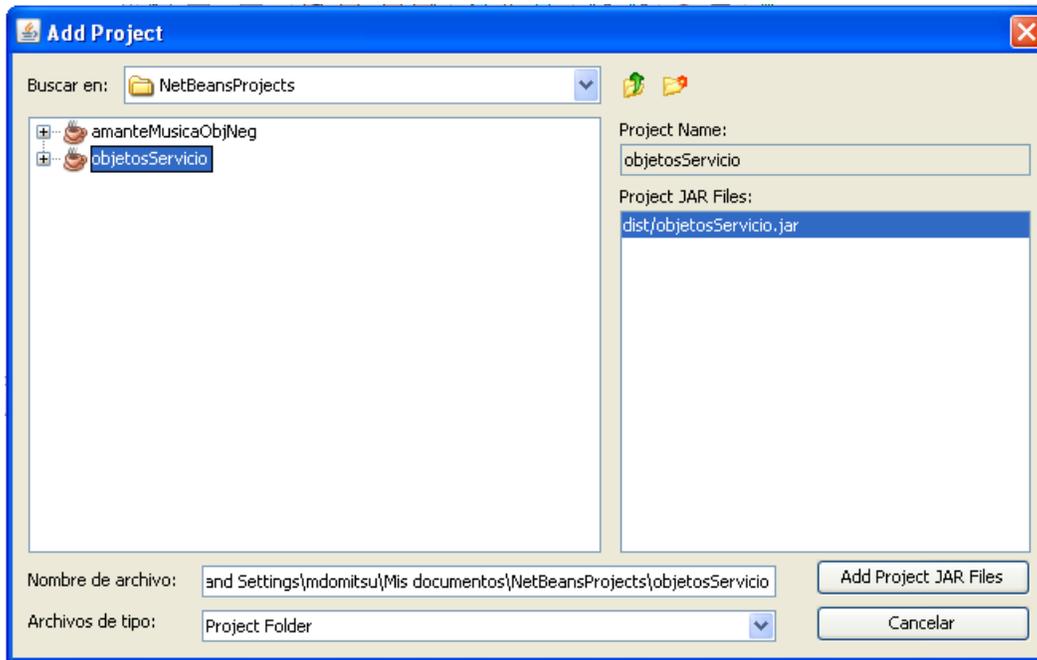


Figura 46

- En la parte superior, hay una caja combinada para seleccionar la carpeta con los proyectos existentes, mismos que se muestran en el recuadro central (cada uno en su propia carpeta). Al seleccionar un proyecto, su nombre aparece del lado derecho y debajo de éste la lista de archivos JAR que hay en este proyecto. Se selecciona el archivo JAR deseado y se presiona el botón **Add Project JAR Files**. Al hacerlo, NetBeans crea la dependencia de nuestro proyecto con el proyecto seleccionado y crea un nodo dentro del nodo **Libraries**, con el nombre del proyecto del que depende nuestro proyecto, figura 47.

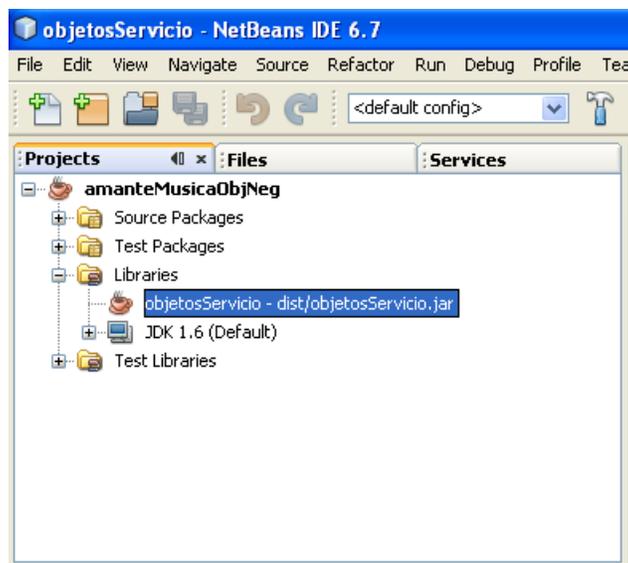


Figura 47

Agregar a un Proyecto Archivos JAR

El procedimiento anterior de hacer que un proyecto A dependa de otro B, sólo se puede emplear si se dispone del proyecto B. En muchas ocasiones, sobre todo cuando adquirimos clases desarrolladas por terceros, sólo tenemos uno o más archivos JAR con el código byteCode de las clases. NetBeans nos permite establecer que un proyecto utilice uno o más archivos JAR, dándonos acceso al código byteCode de las clases de estos archivos.

Para hacer que un proyecto utilice un archivo JAR se sigue el siguiente procedimiento:

1. En el árbol de proyectos expanda el nodo del proyecto dependiente y haga clic con el botón derecho en el nodo **Libraries** y seleccione la opción **Add JAR/Folder ...**, del menú emergente, figura 48.

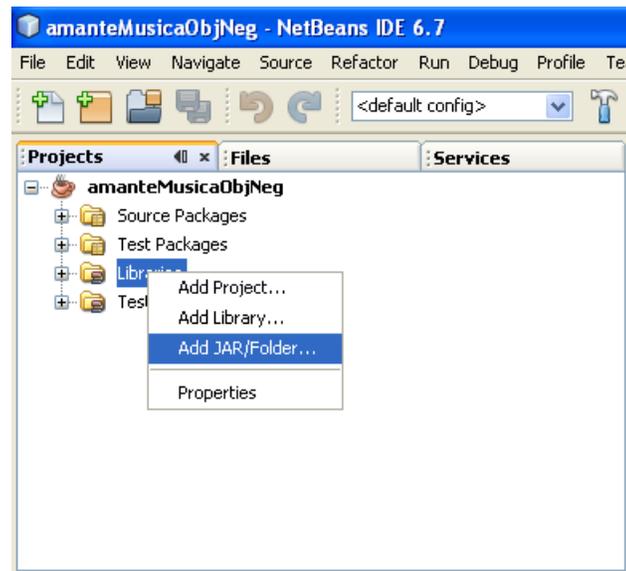


Figura 48

2. Aparecerá el cuadro de diálogo para seleccionar el archivo JAR que utilizará nuestro proyecto, figura 49.
 - a. Encuentre el archivo JAR deseado y selecciónelo.
 - b. Asegúrese que el botón de radio **Relative Path:** de la sección **Reference as** esté seleccionado.
 - c. Haga clic en el botón **Abrir**.

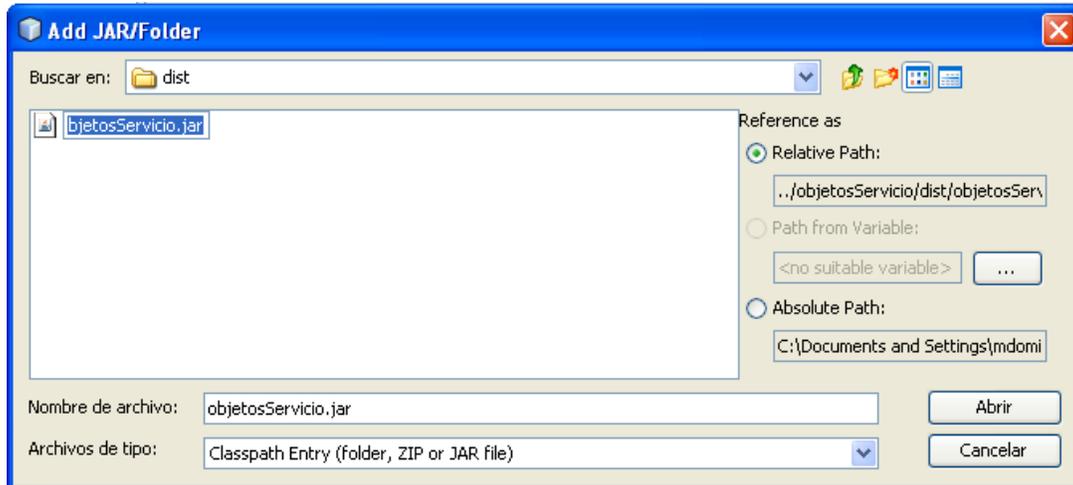


Figura 49

- Al hacerlo, NetBeans crea permite el acceso de nuestro proyecto al código del archivo JAR seleccionado y crea un nodo dentro del nodo **Libraries**, con el nombre del archivo JAR agregado, figura 50.

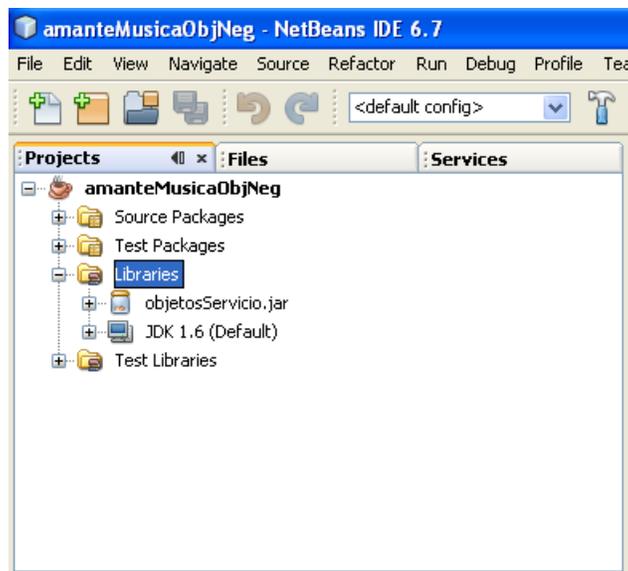


Figura 50