

Pruebas Unitarias con NetBeans

Las pruebas unitarias nos permiten asegurarnos que los métodos individuales del código de un proyecto funcionen correctamente. JUnit es un marco de trabajo que nos permite crear, ejecutar y administrar las pruebas unitarias para un proyecto dado. Netbeans integra a su ambiente de desarrollo dos versiones de ese marco de trabajo. Las versiones 3 y 4. En este tutorial se utilizará la versión 4 ya que esta versión contiene una serie de elementos que facilita la creación, ejecución y administración de las pruebas unitarias.

Agregar la biblioteca JUnit 4 a un Proyecto

En este tutorial supondremos que ya tenemos un proyecto con al menos una clase de java a la que le queremos agregar pruebas unitarias y que ese proyecto ya está abierto en NetBeans. Cuando se crea un proyecto en NetBeans, automáticamente se le agrega al proyecto la biblioteca de la versión 3 de JUnit (al menos desde la versión 6.0 en adelante, de NetBeans). Como queremos usar la versión 4 de JUnit, tendremos que agregarle esa biblioteca al proyecto. Esa biblioteca ya se encuentra disponible en la instalación de NetBeans:

El procedimiento para agregarle a un proyecto la biblioteca de JUnit 4, es el siguiente:

1. Expanda el nodo **Test Libraries** (Bibliotecas de Prueba) del proyecto al que se le va a agregar la biblioteca, figura 1. Note que el proyecto ya tiene agregada una versión 3.X.X.X de la biblioteca de JUnit.

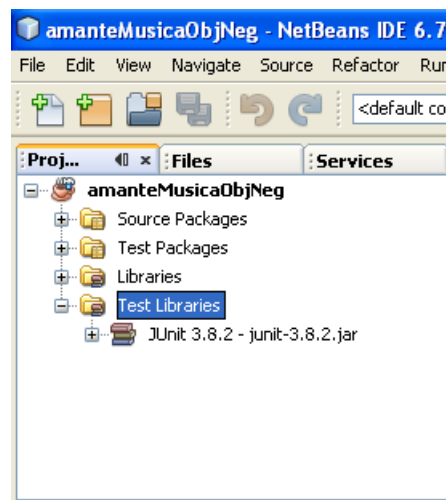


Figura 1

- Haga clic con el botón derecho sobre el nodo **Test Libraries** del proyecto y seleccione la opción **Add Library ...** del menú emergente, figura 2.

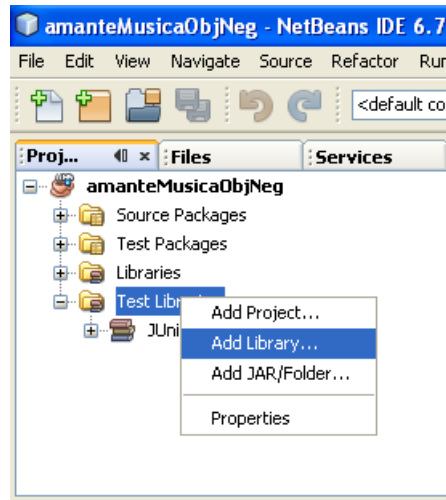


Figura 2

- Aparece un cuadro de diálogo para seleccionar la biblioteca a agregar, en este caso la biblioteca de JUnit 4, Figura 3. Lo seleccionamos y presionamos el botón **Add Library**.

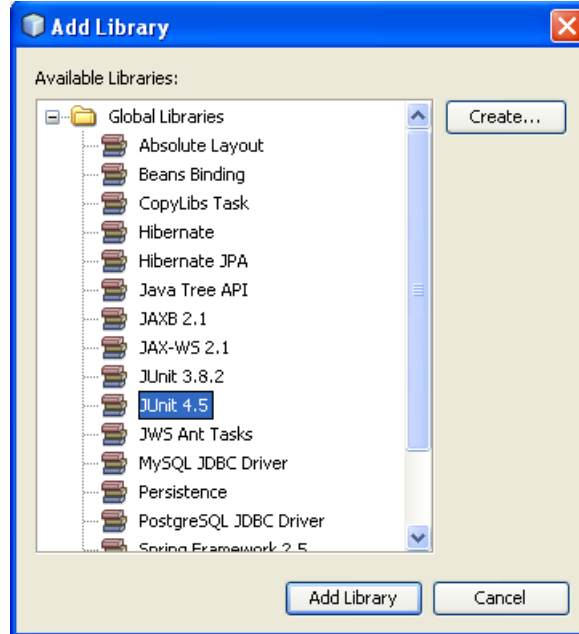


Figura 3

- El cuadro de diálogo desaparece y veremos que la biblioteca de JUnit 4 se ha agregado al nodo **Test Libraries** en el árbol del proyecto, figura 4.

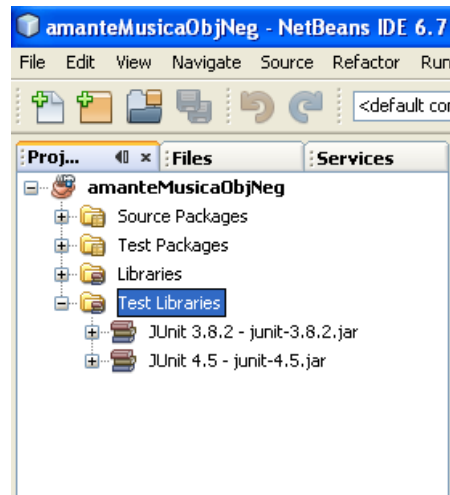


Figura 4

Creación Pruebas Unitarias JUnit 4

NetBeans nos permite crear una clase de prueba para una clase dada. La clase de prueba contiene un conjunto de esqueletos de pruebas unitarias para la clase. Una prueba unitaria para cada método de la clase. El procedimiento para crear una clase de prueba para una clase es el siguiente:

1. En el panel **Projects** expanda los nodos dentro del nodo **Source Packages** hasta encontrar la clase de la que se desean las pruebas unitarias, en este caso la clase `Medio`, figura 5.

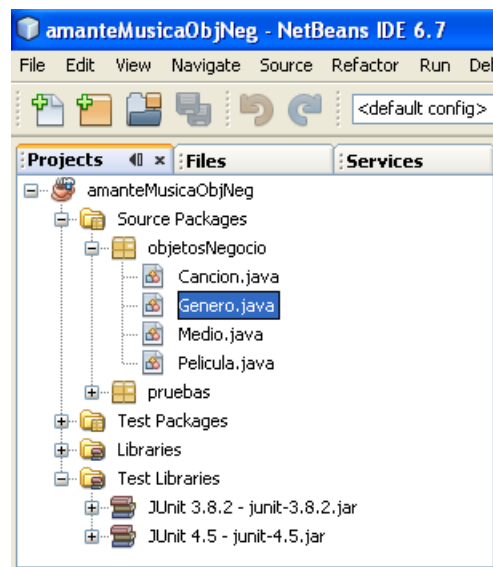


Figura 5

- Haz clic con el botón derecho sobre el nodo de la clase. Selecciona las opciones **Tools/Create JUnit Tests** de los menús emergentes, figura 6. Alternativamente se puede presionar las teclas rápidas **Ctrl+Mayúsculas+U**.

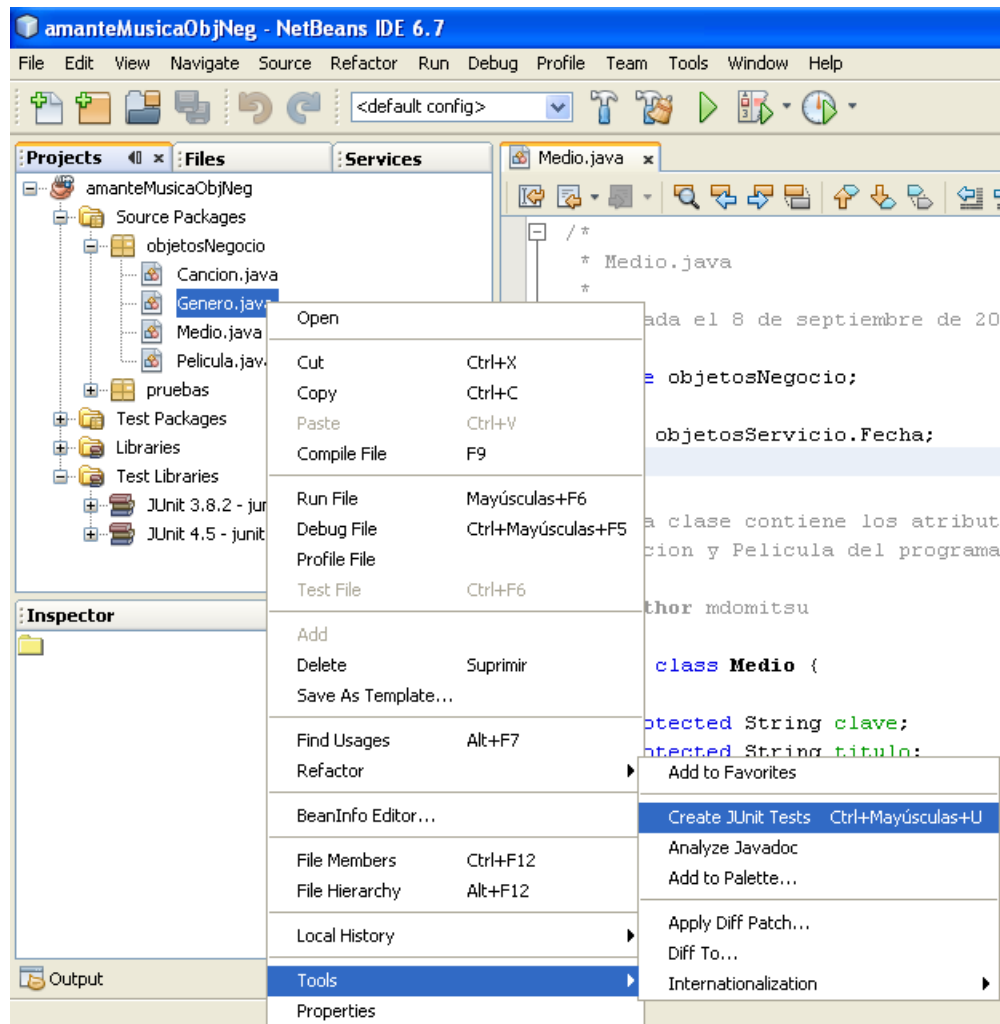


Figura 6

- Aparecerá un cuadro de diálogo para seleccionar la versión de la biblioteca de JUnit a usar, figura 7.

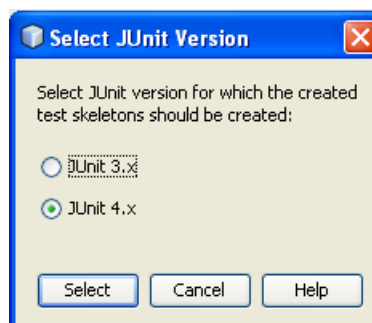


Figura 7

4. Selecciona la opción **JUnit 4.x** y presiona el botón **Select**. La versión 3.x de la biblioteca de JUnit se elimina del nodo **Test Libraries** en el árbol del proyecto, figura 8 y aparece un cuadro de diálogo para establecer el nombre y el paquete de la clase de prueba, así como seleccionar los niveles de visibilidad de los métodos de los que se generarán las pruebas, el código a generar y si se desean agregar comentarios de documentación, Figura 10:

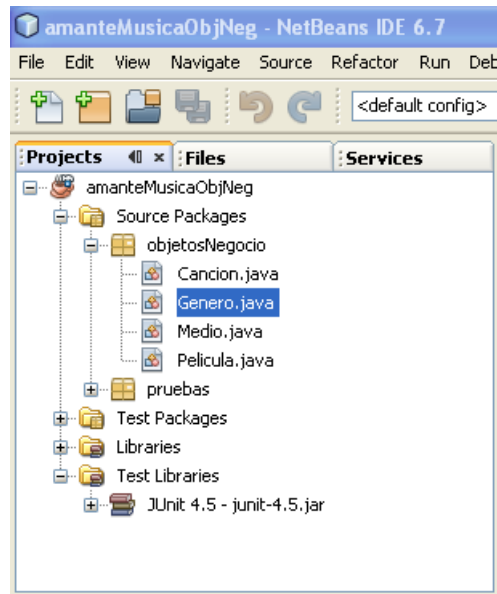


Figura 8

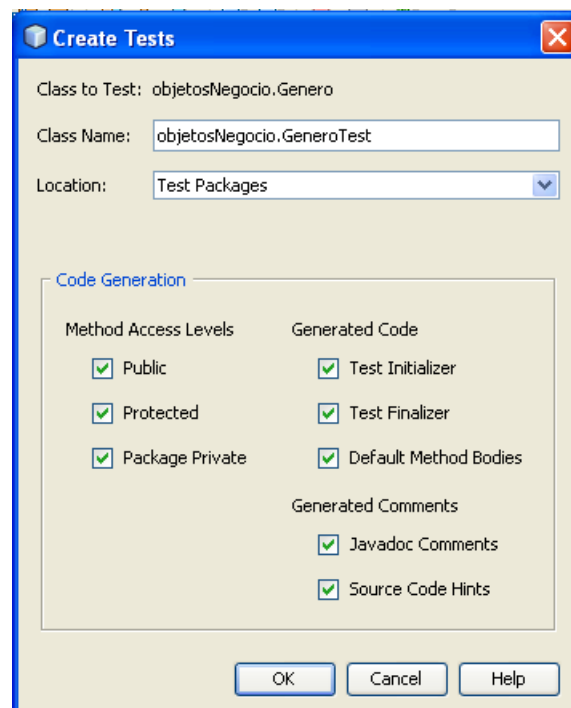


Figura 9

5. En este caso deje el nombre de la clase y el paquete sugeridos por Netbeans: `objetosNegocio.GeneroTest`, así como su ubicación: `Test Packages`. Desmarque las casillas de verificación **Test Initializer** y **Test Finalizer** y deje las demás casillas como están. Presione el botón **OK**. Se creará la clase de prueba con el nombre y en el paquete especificado, figura 10 y Netbeans despliega su código en el panel de edición, figura 11:

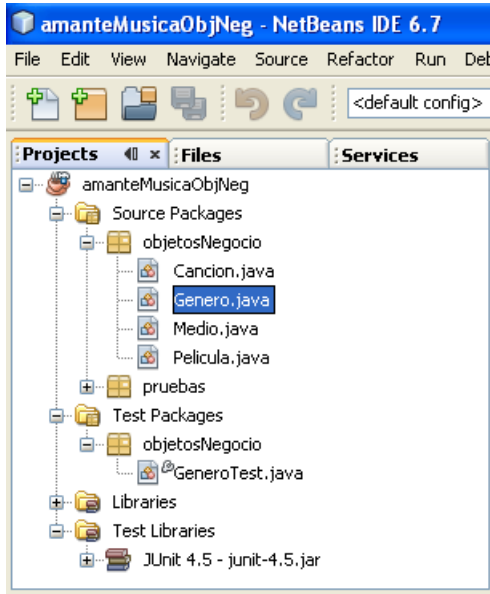


Figura 10

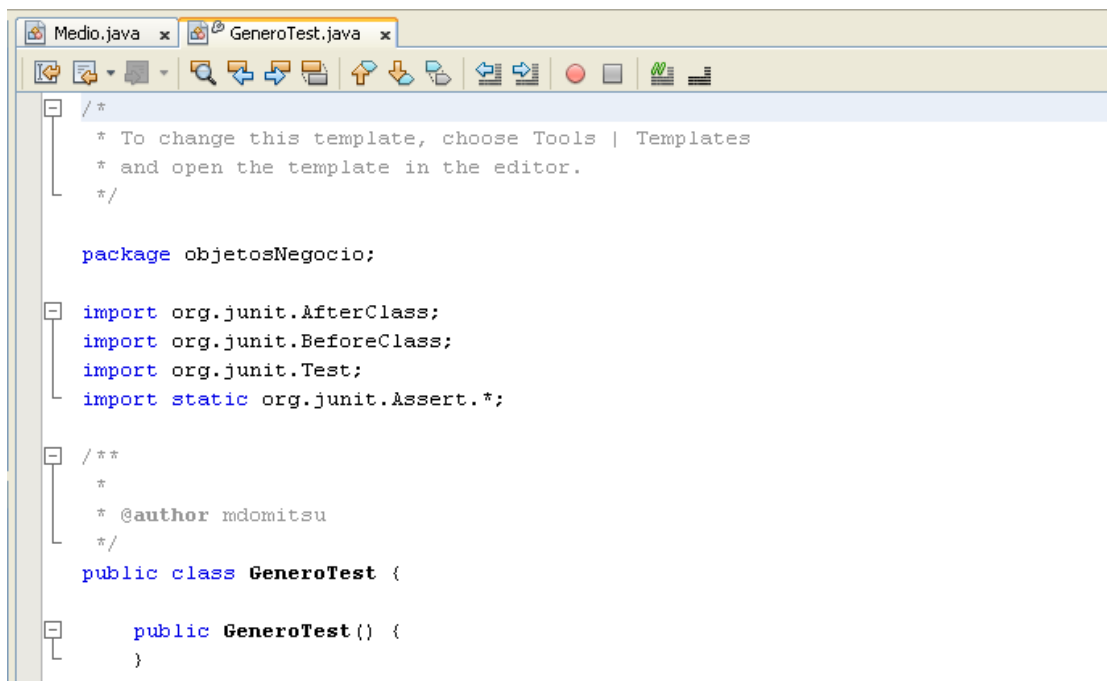


Figura 11

6. Modifique el código de la clase de prueba `GeneroTest` de la siguiente manera:

```
/*
 * GeneroTest.java
 *
 * @author mdomitsu
 */

package objetosNegocio;

import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

/**
 * Clase de prueba de la clase Genero del proyecto amanteMusicaObjneg
 */
public class GeneroTest {

    public GeneroTest() {
    }

    @BeforeClass
    public static void setUpClass() throws Exception {
    }

    @AfterClass
    public static void tearDownClass() throws Exception {
    }

    /**
     * Prueba del metodo getCveGenero, de la clase Genero.
     */
    @Test
    public void testGetCveGenero() {
        Genero instance = null;

        System.out.println("Prueba del método getCveGenero()");

        // Verifica si la clave del genero es null.
        // Uso de un constructor vacio
        instance = new Genero();
        assertNull(instance.getCveGenero());

        // Verifica si la clave de genero es de un valor dado.
        // Uso de otro constructor
        instance = new Genero("GC001", "Balada", 'C');
        assertEquals("GC001", instance.getCveGenero());
    }

    /**
     * Prueba del metodo setCveGenero, de la clase Genero.
     */
    @Test
    public void testSetCveGenero() {
        Genero instance = null;

        System.out.println("Prueba del método setCveGenero()");
    }
}
```

```
        // Verifica si la clave del genero toma el valor
        //dado por el método
        instance = new Genero();
        instance.setCveGenero("GC001");
        assertEquals("GC001", instance.getCveGenero());
    }

    /**
     * Prueba del metodo getNombre, de la clase Genero.
     */
    @Test
    public void testGetNombre() {
        Genero instance = null;

        System.out.println("Prueba del método getNombre()");

        // Verifica si el nombre es de un valor dado.
        instance = new Genero("GC001", "Balada", 'C');
        assertEquals("Balada", instance.getNombre());
    }

    /**
     * Prueba del metodo setNombre, de la clase Genero.
     */
    @Test
    public void testSetNombre() {
        Genero instance = null;

        System.out.println("Prueba del método setNombre()");

        // Verifica si el nombre toma el valor dado por el método
        instance = new Genero();
        instance.setNombre("Balada");
        assertEquals("Balada", instance.getNombre());
    }

    /**
     * Prueba del metodo getTipoMedio, de la clase Genero.
     */
    @Test
    public void testGetTipoMedio() {
        Genero instance = null;

        System.out.println("Prueba del método getTipoMedio()");

        // Verifica si el tipo de medio es de un valor dado.
        instance = new Genero("GC001", "Balada", 'C');
        assertEquals('C', instance.getTipoMedio());
    }

    /**
     * Prueba del metodo setTipoMedio, de la clase Genero.
     */
    @Test
    public void testSetTipoMedio() {
        Genero instance = null;
```



```
        System.out.println("Prueba del método setTipoMedio()");

        // Verifica si el tipo de medio toma el valor dado por el método
        instance = new Genero();
        instance.setTipoMedio('C');
        assertEquals('C', instance.getTipoMedio());
    }

    /**
     * Prueba del metodo equals, de la clase Genero.
     */
    @Test
    public void testEquals() {
        Genero instance = null;
        Object obj = null;

        System.out.println("Prueba del método equals()");

        instance = new Genero("GC001", "Balada", 'C');

        // Prueba la comparacion contra null
        boolean resultado = instance.equals(obj);
        assertFalse(resultado);

        // Prueba la comparacion contra un objeto con solo
        // la clave del genero
        obj = new Genero("GC001", null, ' ');
        resultado = instance.equals(obj);
        assertTrue(resultado);

        // Prueba la comparacion contra un objeto
        obj = new Genero("GC001", "Balada", 'C');
        resultado = instance.equals(obj);
        assertTrue(resultado);
    }

    /**
     * Prueba del metodo toString, de la clase Genero.
     */
    @Test
    public void testToString() {
        Genero instance = null;

        System.out.println("Prueba del método toString()");

        // Verifica si se genera la cadena correcta
        instance = new Genero("GC001", "Balada", 'C');
        assertEquals("GC001, Balada, C", instance.toString());
    }
}
```

Ejecución de las Pruebas Unitarias de una Clase

NetBeans nos permite ejecutar las pruebas unitarias de los métodos de una clase. Para ejecutar las pruebas unitarias de una clase se sigue el siguiente procedimiento:

1. Haz que el panel de la clase de prueba o de la clase a probar sea el panel activo en la ventana de edición.
2. Del menú principal de NetBeans seleccione la opción **Run/Test File** o presione las teclas **Ctrl+F6**, figura 12.

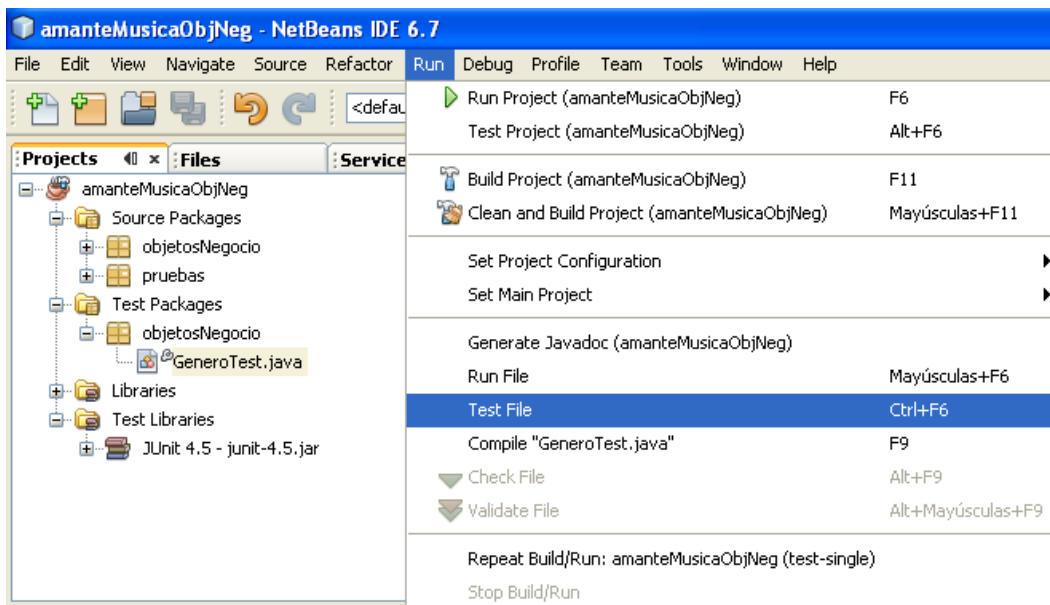


Figura 12

3. Como alternativa haga clic con el botón derecho sobre el nodo de la clase de prueba que desea ejecutar y seleccione la opción **Test File** del menú emergente o presione las teclas **Ctrl+F6**, figura 13.
4. Al hacerlo NetBeans compilará y ejecutará la clase de prueba desplegando en el Panel **Test Results** el resultado de la ejecución de las pruebas unitarias, figura 14. Si en alguno de los métodos hubiera un error y la prueba unitaria fallara, Netbeans desplegaría un mensaje indicando que esa prueba unitaria falló y la causa, figura 15.

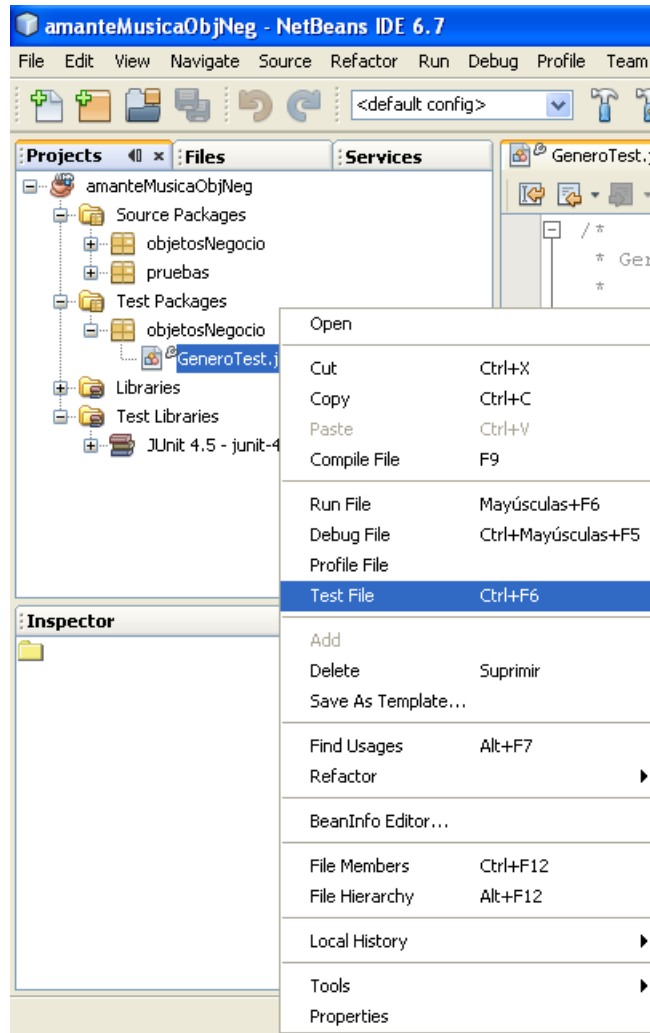


Figura 13

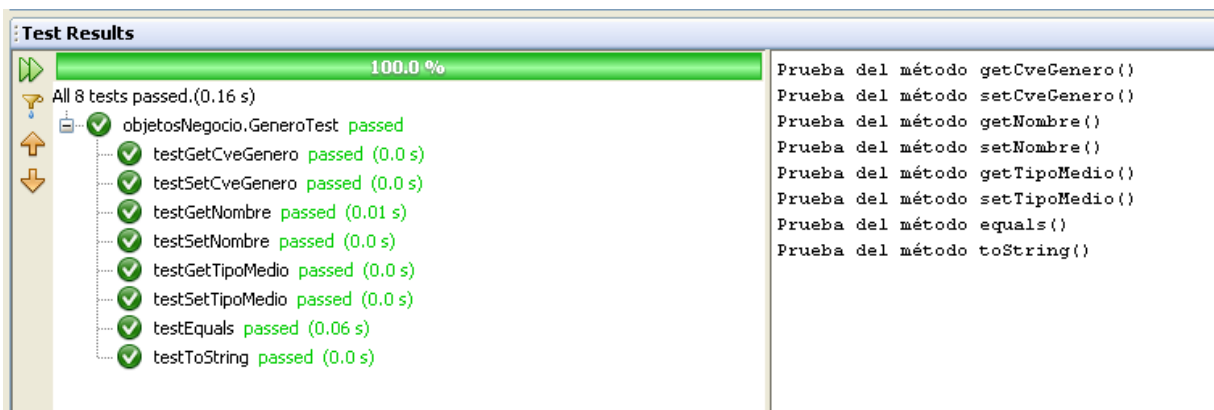


Figura 14

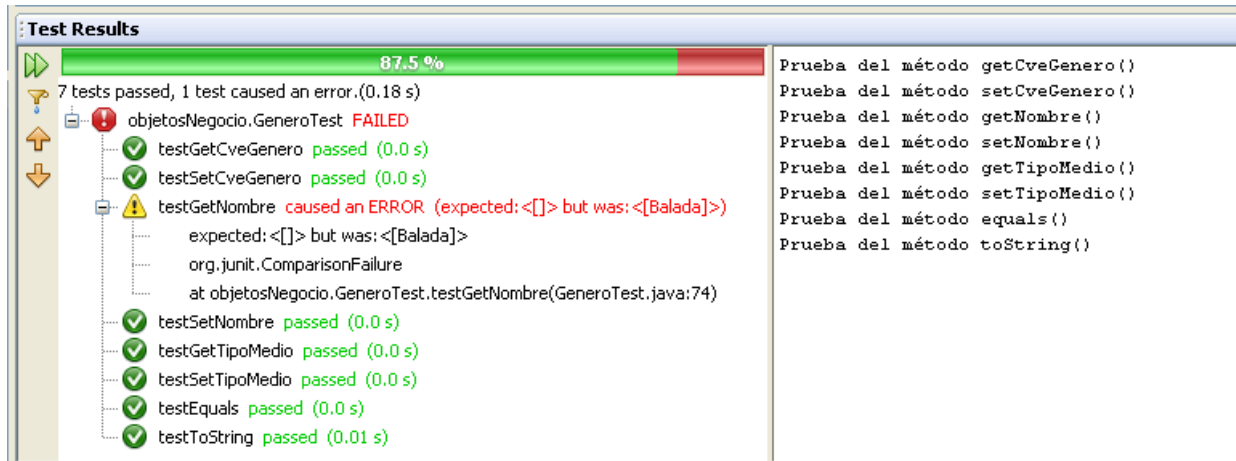


Figura 15

Ejecución de las Pruebas Unitarias de las Clases de un Proyecto

NetBeans, también nos permite ejecutar las pruebas unitarias de los métodos de las clases de un proyecto. Para ejecutar las pruebas unitarias de las clases de un proyecto se sigue el siguiente procedimiento:

1. Del menú principal de NetBeans seleccione la opción **Run/Test Proyecto (nomProyecto)**, donde *nomProyecto* es el nombre del proyecto a probar, o presione las teclas **Alt+F6**, figura 16.

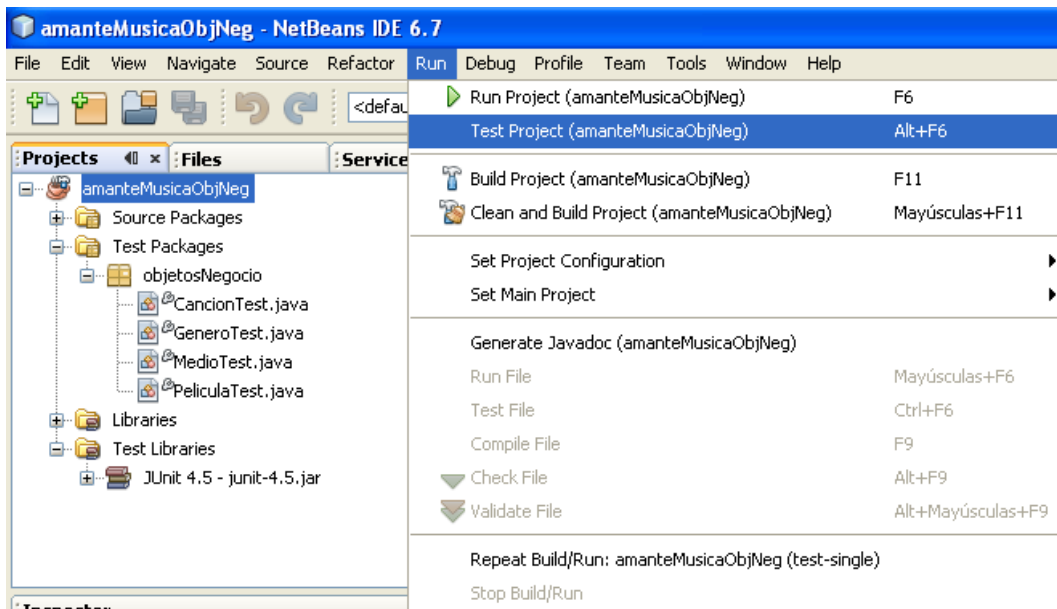


Figura 16

- Como alternativa haga clic con el botón derecho sobre el nodo del proyecto a probar y seleccione la opción **Test** o presione las teclas **Alt+F6**, figura 17.

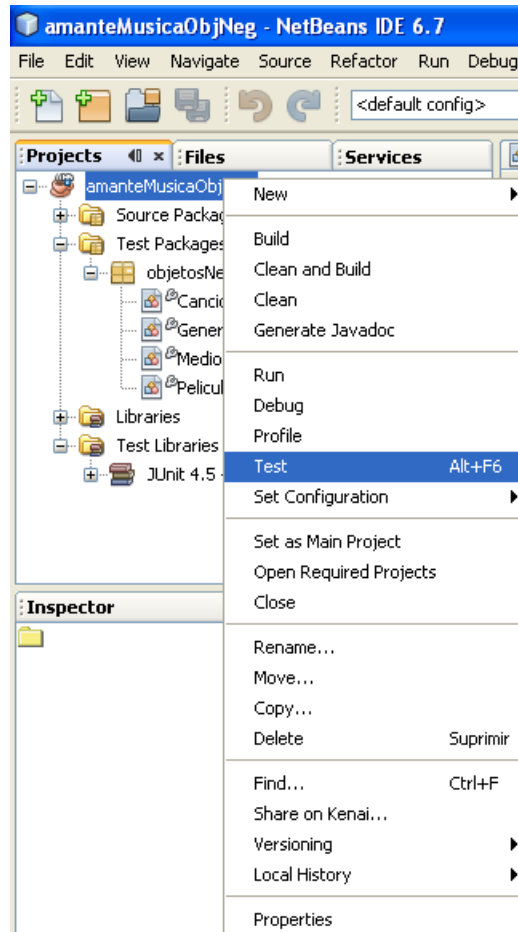


Figura 17

- Al hacerlo NetBeans compilará y ejecutará las clases de prueba del proyecto desplegando en el Panel **Test Results** los resultados de la ejecución de las pruebas unitarias de las clases, figura 18. Si en alguno de los métodos hubiera un error y la prueba unitaria fallara, Netbeans desplegaría un mensaje indicando que esa prueba unitaria falló y la causa, figura 19.

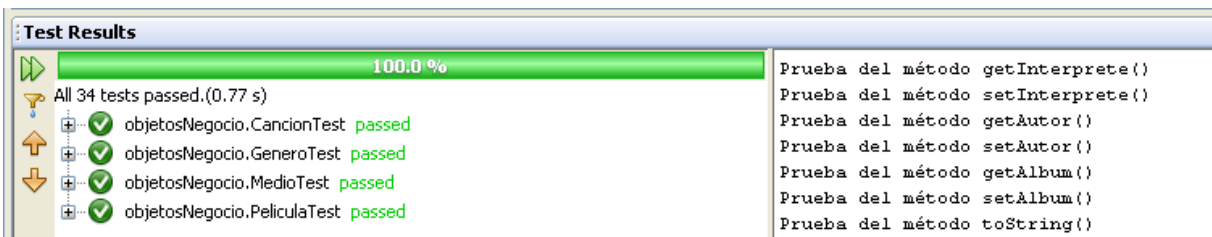


Figura 18

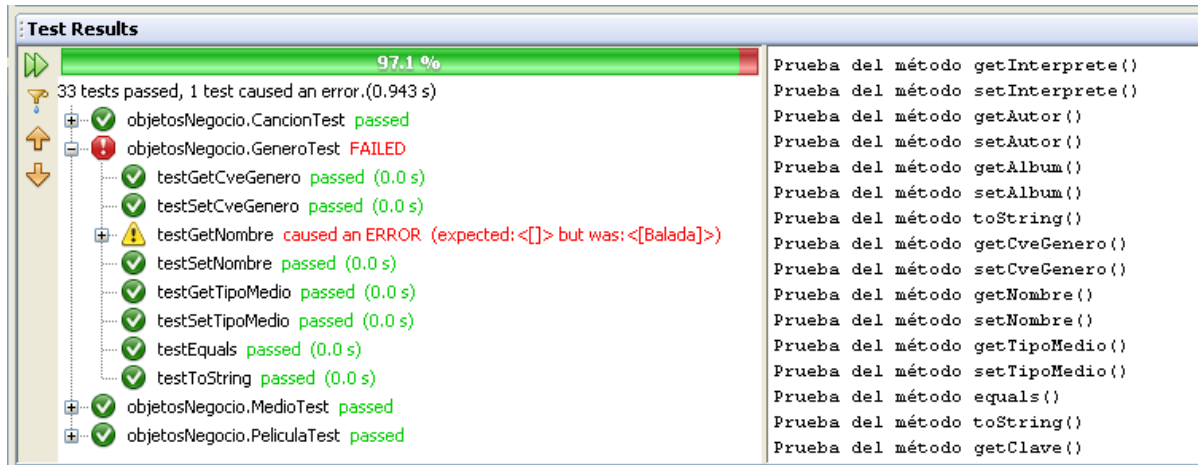


Figura 19

Creación de un Conjunto de Pruebas Unitarias JUnit 4

Aparte de ejecutar las pruebas unitarias de una clase o de un proyecto, hay ocasiones en que nos interesa ejecutar un subconjunto de pruebas o ejecutar pruebas en un orden específico. Para ello, NetBeans nos permite crear uno o más conjunto de pruebas. Un conjunto de pruebas es una clase con un método que invoca los casos de prueba específicos, como podrían ser clases de prueba, métodos de prueba de clases de prueba y otros conjuntos de prueba. El procedimiento para crear un conjunto de pruebas es el siguiente:

1. Haga clic con el botón derecho en el nodo del proyecto con las clases de las que se desea hacer el conjunto de pruebas. Selecciona las opciones **New/Other** de los menús emergentes, figura 20.
2. Aparecerá el cuadro de diálogo de la figura 21. Seleccione del panel **Categories** la opción **JUnit** y del panel **File Types** la opción **Test Suite**. Presione el botón **Next**.
3. En el cuadro de diálogo de la figura 22 establezca el nombre de la clase del conjunto de pruebas: **Class Name** a amanteMusicaObjNegTestSuite y su paquete: **Package** a conjuntoPruebas. Desmarque las casillas de verificación **Test Initializer** y **Test Finalizer** y deje las demás casillas como están. Presione el botón **Finish**. Se creará la clase del conjunto de pruebas con el nombre y en el paquete especificado, figura 23 y Netbeans despliega su código en el panel de edición, figura 24:

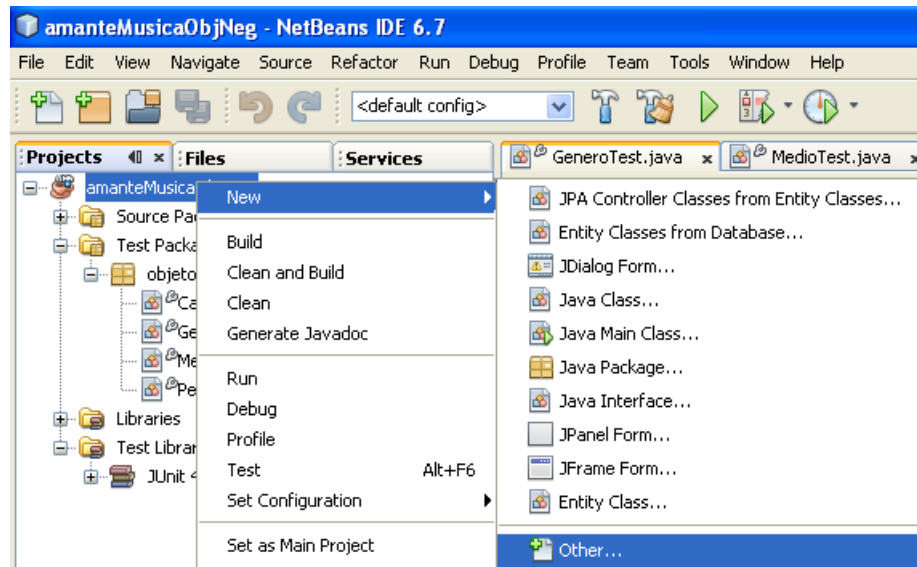


Figura 20

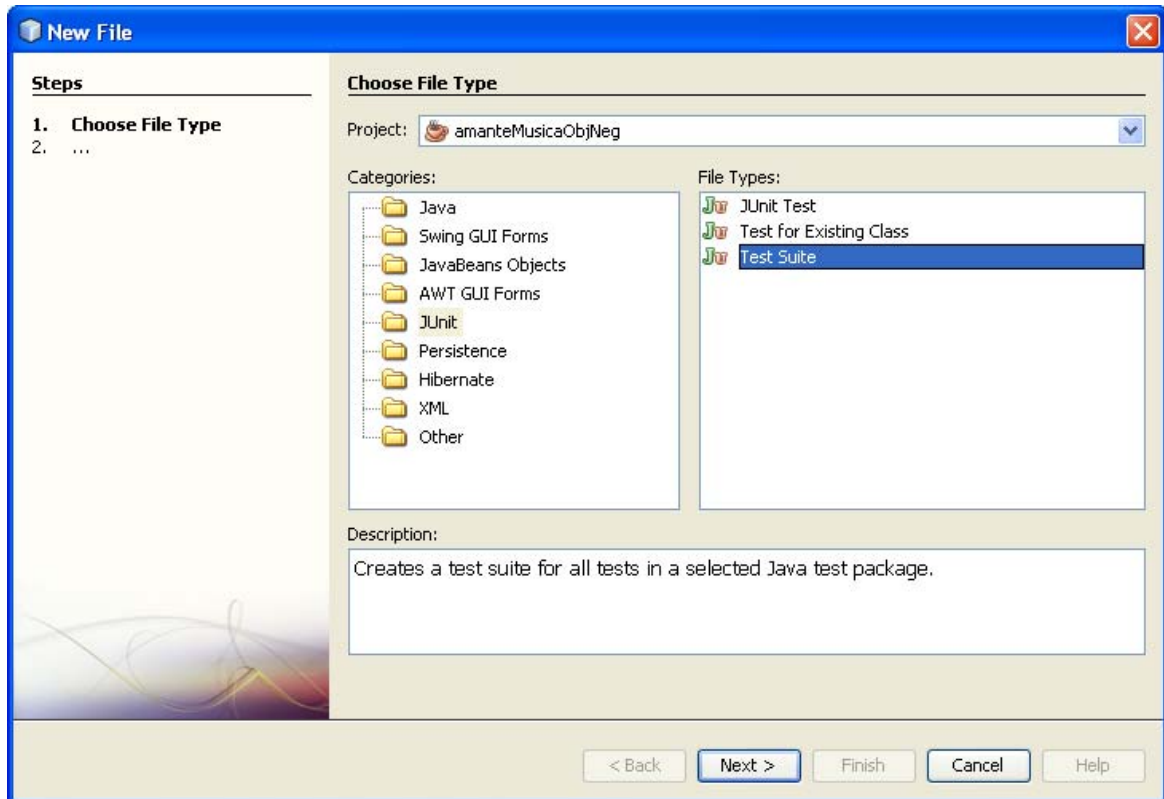


Figura 21

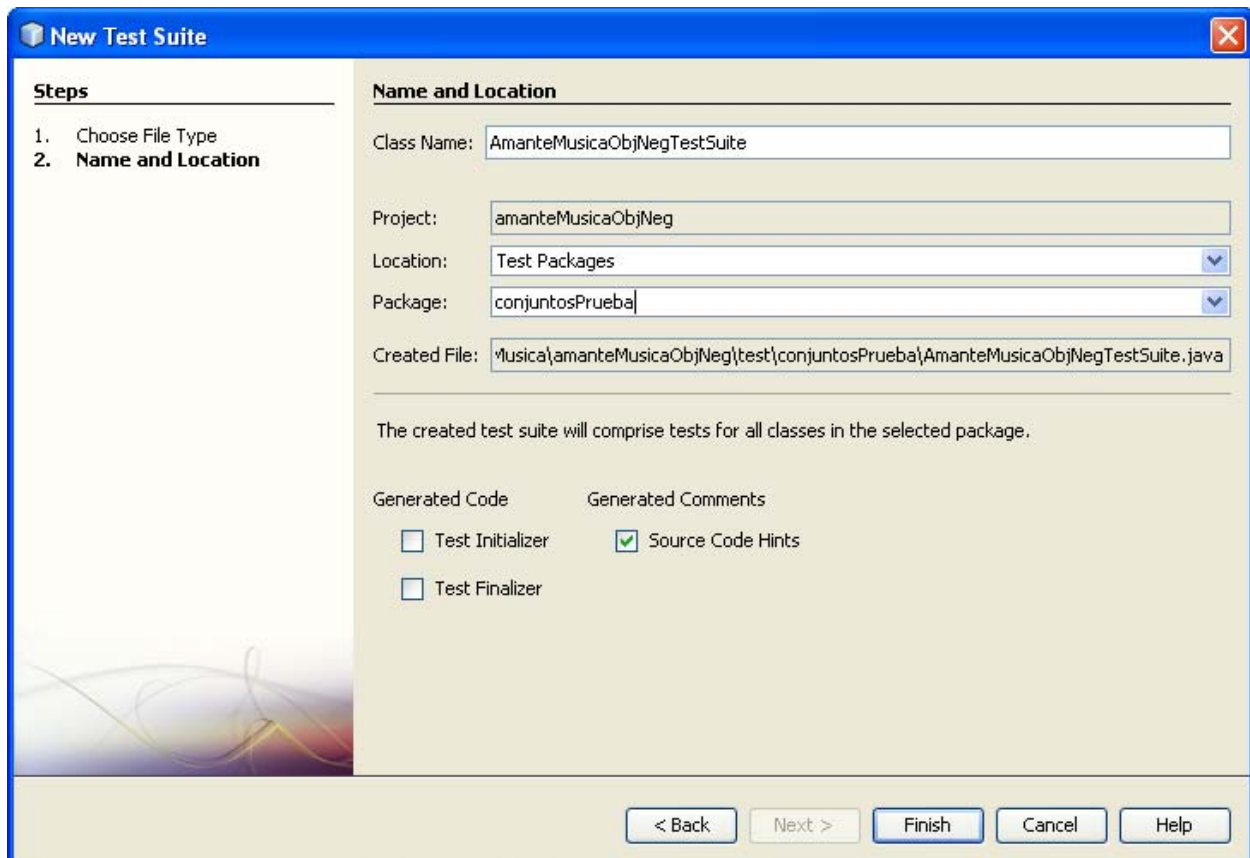


Figura 22

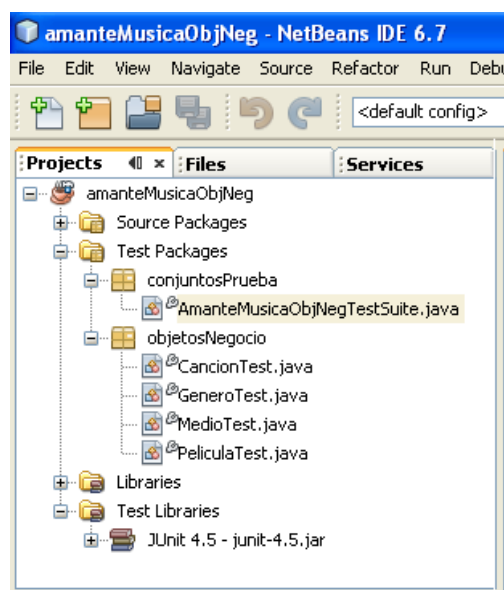


Figura 23

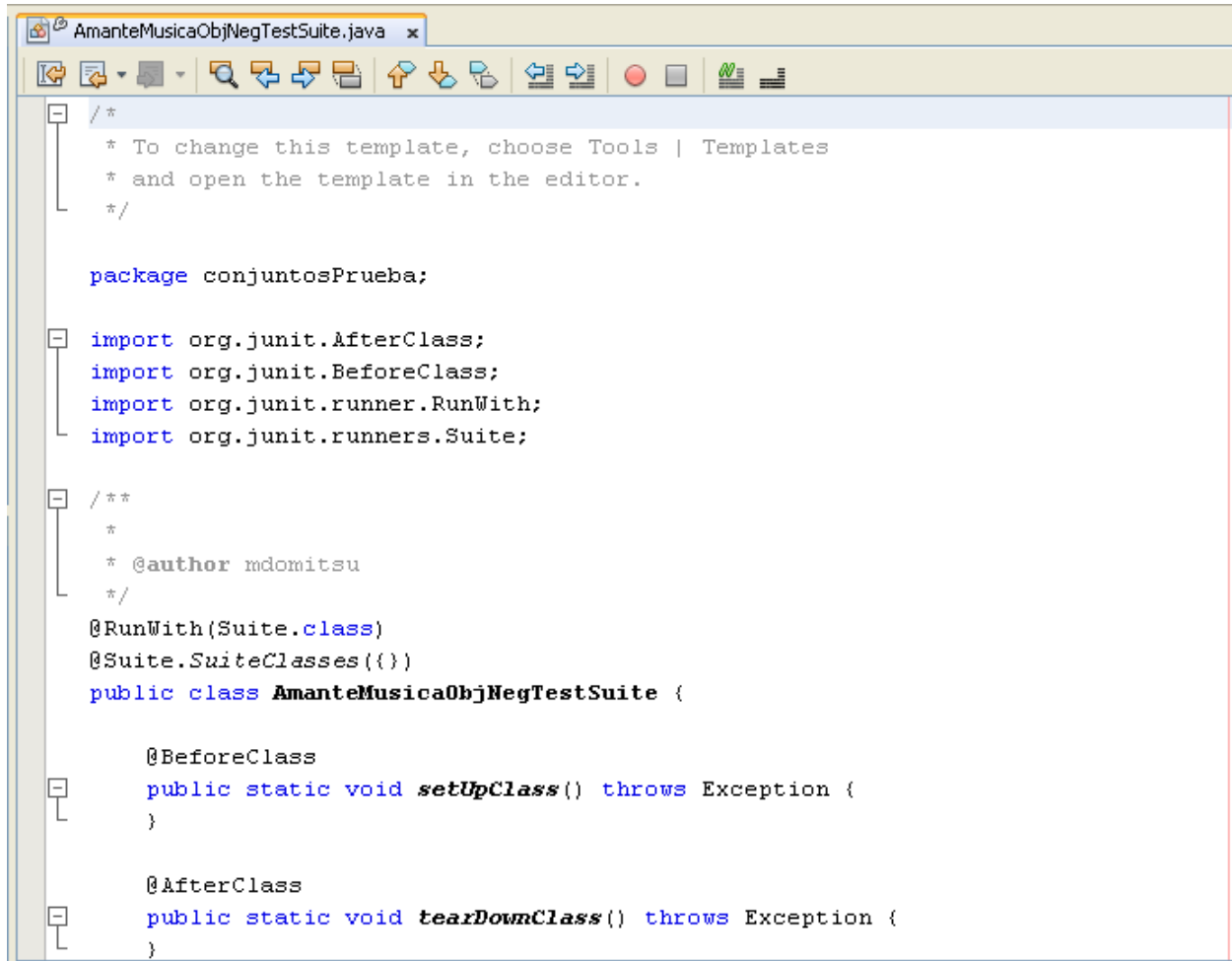


Figura 24

4. La línea de código con la anotación:

```
@Suite.SuiteClasses({})
```

Permite establecer las clases de prueba que se desean ejecutar y el orden de ejecución. Por ejemplo, si se desean ejecutar las clases de prueba `GeneroTest` y `MedioTest` en ese orden debemos modificar la línea de código anterior a:

```
@Suite.SuiteClasses(value={objetosNegocio.GeneroTest.class,
                        objetosNegocio.MedioTest.class})
```

Note que en la lista de clases de prueba van los nombres de los archivos con el código de bytes (`.class`) incluyendo el nombre del paquete en que se encuentran, separados por comas.

Ejecución de un Conjunto de Pruebas Unitarias

Para ejecutar un conjunto de pruebas unitarias se sigue el siguiente procedimiento:

1. Haga clic con el botón derecho en el nodo del conjunto de pruebas y seleccione la opción **Test File** o presione las teclas **Ctrl+F6**, figura 25.

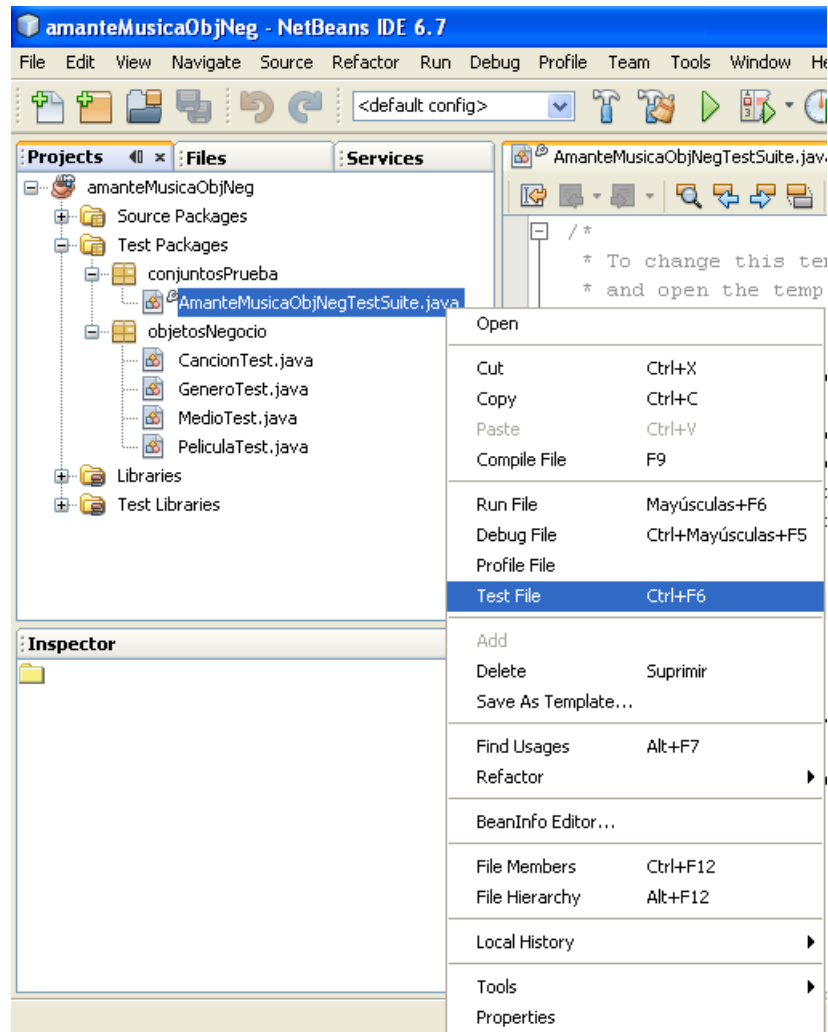


Figura 25

2. Al hacerlo NetBeans compilará y ejecutará las clases de prueba del conjunto de pruebas desplegando en el Panel **Test Results** los resultados de la ejecución de las pruebas unitarias de las clases del conjunto de pruebas, figura 26. Si en alguno de los métodos hubiera un error y la prueba unitaria fallara, Netbeans desplegaría un mensaje indicando que esa prueba unitaria falló y la causa, figura 27.

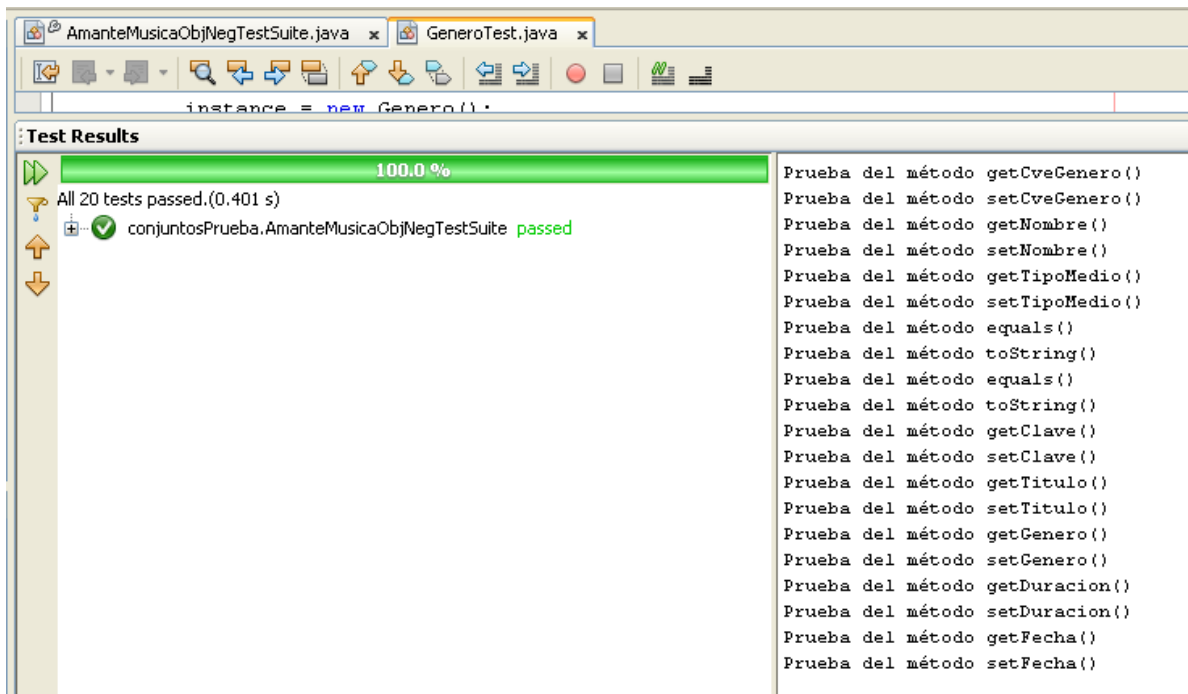


Figura 26

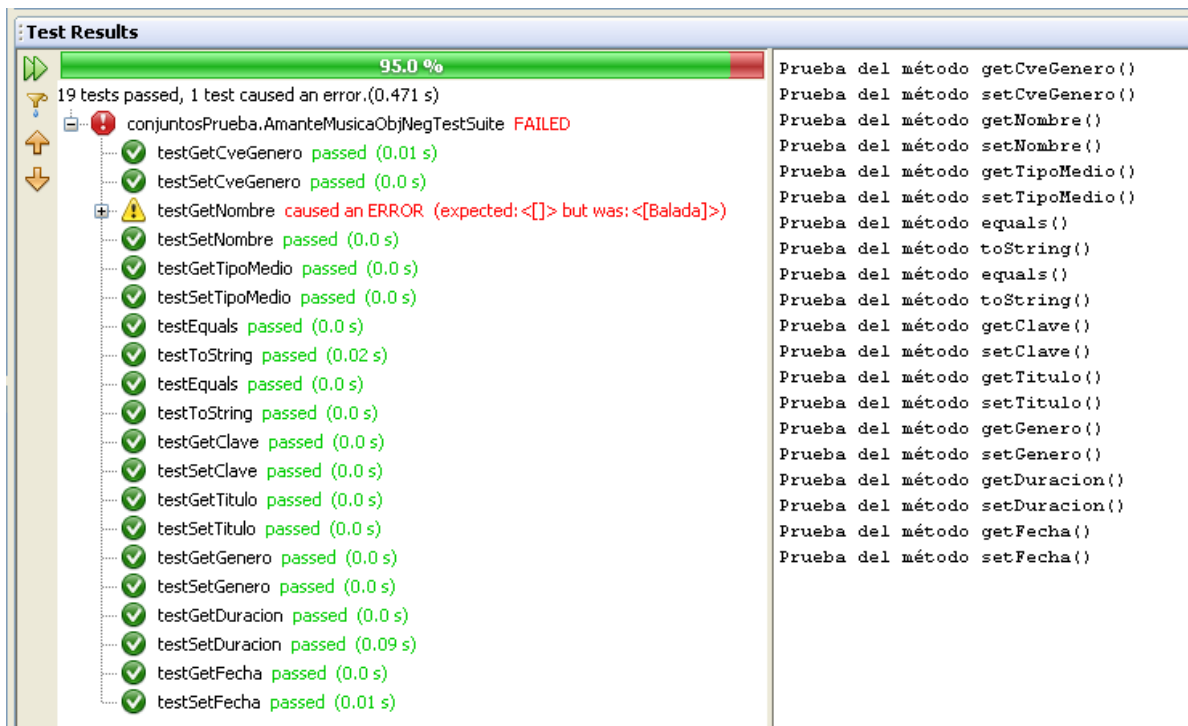


Figura 27