

Tema 6

GUI, Swing y Java Beans

Interfaz de Usuario Gráfica

Para hacer más amigable la interacción con el usuario, actualmente la mayoría de los programas comerciales utilizan una interfaz de usuario gráfica (GUI).

El ejemplo más conocido es el sistema operativo Windows y sus aplicaciones. Sin embargo no fue el primero ni el único en usar una GUI.

Paquetes de Java para GUIs

Para facilitarnos la programación de las GUIs, Java nos provee de un conjunto de paquetes:

- **java.awt:** Contiene todas las clases para crear interfaces de usuario y para dibujar gráficas e imágenes.
- **javax.swing:** Provee un conjunto de componentes ligeros (escritos completamente en Java) que, en lo máximo posible, trabajan de la misma manera en todas las plataformas.
- **java.awt.event:** Provee interfaces y clases para manejar los diferentes tipos de eventos disparados por los componentes AWT.

Contenedores

Los diferentes componentes que forman la interfaz gráfica de una aplicación: Etiquetas, botones, cajas de texto, menús, etc. deben agruparse en otros componentes llamados contenedores. El paquete javax.swing tiene los siguientes contenedores para agrupar componentes.

- **JFrame:** Permite crear una ventana de una aplicación. Posee título y puede tener una barra de menús, barra de herramienta, barras de desplazamiento, su propio cursor, botones para maximizar, minimizar, etc.
- **JDialog:** Permite crear cuadros de diálogo.
- **JPanel:** Permite agrupar componentes. Podemos tener paneles dentro de paneles. Tienen un manejador de diseño.

Manejadores de Diseño

Un manejador de diseño es una clase que define la forma en que se acomodan los componentes insertados en un contenedor. El paquete `javax.swing` tiene los siguientes manejadores de diseño para acomodar componentes en un contenedor.

- **FlowLayout:** Coloca los componentes de izquierda a derecha por filas, con las filas ordenadas de arriba abajo.
- **BorderLayout:** El contenedor se divide en cinco regiones: North, East, West, South y Center.
- **GridLayout:** Acomoda a los componentes en filas y columnas con todas las regiones del mismo tamaño.
- **GridBagLayout:** Acomoda a los componentes en filas y columnas que no necesariamente tienen la misma altura y ancho.
- **CardLayout:** Cada uno de los componentes es tan grande como el contenedor. Sólo un componente es visible.

Componentes Activos

Los componentes activos del paquete Swing de Java permiten que el usuario interactúe con el programa:

- **JButton:** Implementa un botón.
- **JCheckBox:** Implementa una casilla de verificación.
- **JList:** Una componente que permite que el usuario seleccione uno o más objetos de una lista.

Componentes de Texto

Hay cuatro componentes que pueden usarse para el despliegue y captura de textos:

- **JLabel:** Despliega una línea de texto en la pantalla.
- **TextField:** Un cuadro que contiene una línea de texto. El usuario puede modificarla.
- **Password:** Un cuadro que permite la captura de una contraseña. No hace eco de los caracteres. En lugar despliega un carácter dado.
- **TextArea:** Un cuadro que contiene una o más líneas de texto. El usuario puede modificarlas.

Menús

El paquete `java.swing` de Java contiene una serie de clases que permiten crear diferentes menús:

- **JMenuBar:** Permite crear una barra de menús. Contiene un conjunto de objetos de tipo JMenu y debe ser parte de una ventana, JFrame.
- **JMenu:** Permite implementar menús. Contiene una colección de objetos JMenuItem y separadores.
- **JMenuItem:** Permite implementar las opciones de un menú.
- Un Jmenu también puede ser un JMenuItem permitiendo menús jerárquicos.

Programación por Eventos

En la programación orientada por eventos el flujo de un programa no sigue una secuencia de inicio a fin sino que está controlada por eventos externos.

Los programas con una interfaz de usuario gráfica por lo general son controlados por eventos externos como hacer un clic en una opción de un menú o en un botón, arrastrar el ratón, escribir en un campo de texto, etc.

Modelo de Eventos de Java

Un programa en Java maneja los eventos externos de la siguiente manera:

Toda componente puede reaccionar ante un evento externo generando una notificación llamada evento. Se dice que el componente puede ser la fuente de un evento.

Toda clase puede recibir la notificación generada por un componente. Se dice que la clase es una oyente del evento.

Un componente que genera eventos, mantiene una lista de las oyentes interesadas en saber cuando ocurre el evento

El evento generado por un componente sólo se envía a las oyentes registradas. Las que están en la lista del componente.

Eventos

En Java, un evento es una subclase de la clase AWTEvent que se encuentran en el paquete java.awt.event.

Un objeto de tipo evento contiene la información del evento externo que lo generó
Cada componente genera uno o más tipos de eventos

Oyentes

Para que una clase pueda ser oyente de un evento debe implementar una o más interfaces oyente.

Una interfaz oyente declara uno o más métodos oyentes que deben ser implementados por la clase oyente.

Cuando un componente genera un evento, invoca a un método oyente de la clase oyente y le pasa como parámetro el evento.

El método oyente contiene el código que maneja al evento generado por el componente.

Tabla 6.1 Eventos, Componentes y Oyentes

Clase de Evento	Componentes que lo Generan	Significado	Interfaz Oyente	Métodos Oyentes
ActionEvent	JButton JList JMenuItem JTextField	El usuario hace clic sobre el botón. El usuario hace doble clic sobre la opción de la lista. El usuario selecciona una opción del menú. El usuario terminó de editar el texto.	ActionListener	actionPerformed()
FocusEvent	JComponent	El componente ganó o perdió el foco.	FocusListener	focusGained() focusLost()
ItemEvent	JCheckbox JCheckboxMenuItem JChoice JList.	El usuario selecciona o deselecciona una opción.	ItemListener	itemStateChanged()
KeyEvent	JComponent	El usuario presionó o soltó una tecla.	KeyListener	keyPressed() keyReleased() keyTyped()
MouseEvent	JComponent	El usuario presionó o soltó el botón del ratón.	MouseListener	mouseClicked() mouseEntered() mouseExited() mousePressed() mouseReleased()
		El ratón entró o salió del componente, o el usuario movió o arrastró al ratón.	MouseMotionListener	mouseDragged() mouseMoved()
TextEvent	JTextField JTextArea	El usuario cambió el texto	TextListener	textValueChanged()

Ejemplo sobre una Interfaz de Usuario Gráfica

Para el ejemplo de la aplicación del amante de la música y el cine, AmanteMusica, se tiene el diseño de la interfaz de usuario gráfica mostrada en la figura 6.1.

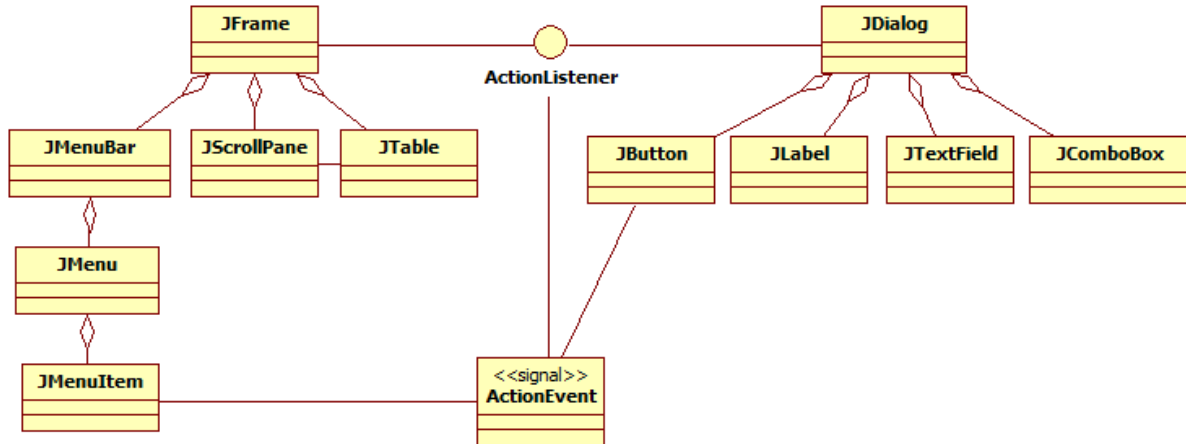


Figura 6.1. Diagrama de clases de la interfaz gráfica de la aplicación Amante de la Música

Interfaz de Usuario de la Aplicación AmanteMusica

La figura 6.2 muestra la ventana principal de la aplicación AmanteMusica, con su barra de menús.

La figura 6.3 muestra la selección de la opción para agregar una canción al catálogo de canciones.

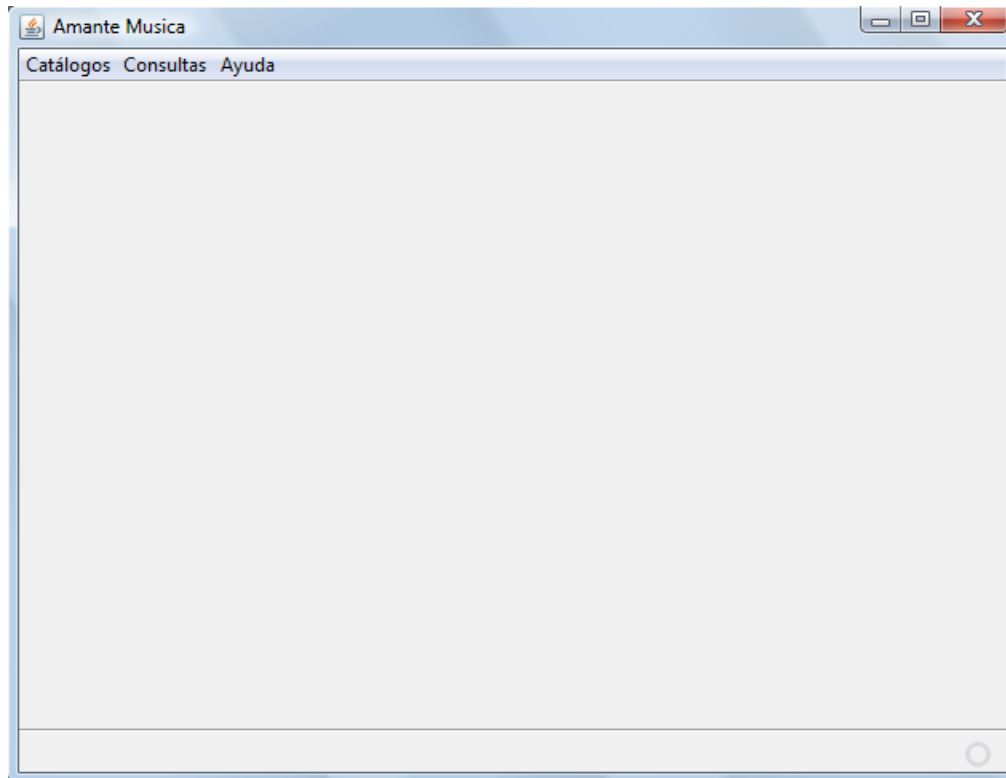


Figura 6.2. Ventana principal de la aplicación AmanteMusica

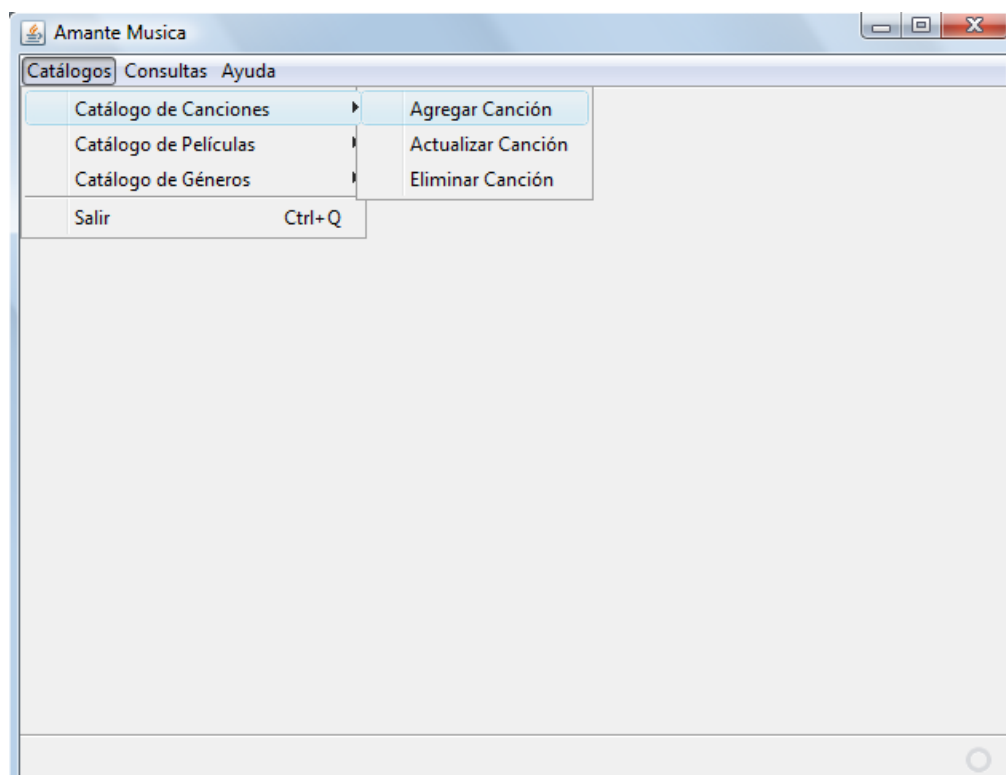


Figura 6.3. Selección de la opción de agregar una canción

Para agregar una canción, primero se captura la clave de una canción utilizando el método estático `showInputDialog()` de la clase `JOptionPane` que permite capturar una línea de texto utilizando un cuadro de diálogo como el mostrado en la figura 6.4:

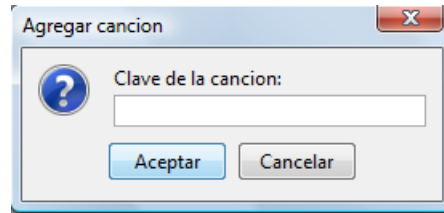


Figura 6.4 Cuadro de diálogo para capturar la clave de una canción.

A continuación se busca la canción de la clave en el catálogo de canciones. Si la canción no existe en el catálogo de canciones, se despliega el cuadro de diálogo mostrado en la figura 6.5 para capturar el resto de los datos de la canción:

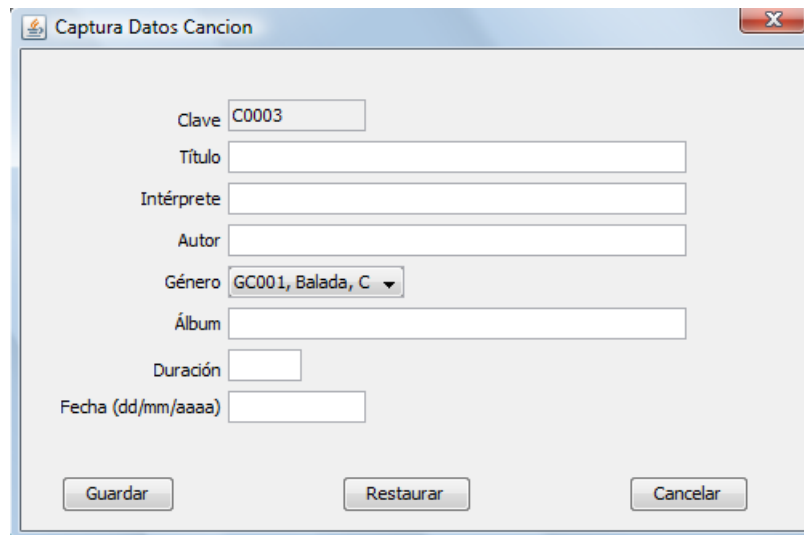


Figura 6.8. Cuadro de diálogo DlgCancion para capturar los datos de una nueva canción.

Después de obtener los datos de una canción, el programa la agrega al catálogo de canciones y posteriormente despliega una tabla con las canciones del catálogo, figura 6.9.

Si la canción ya existe en el catálogo de canciones, el programa despliega en un cuadro de diálogo los datos de la canción existente, figura 6.10.

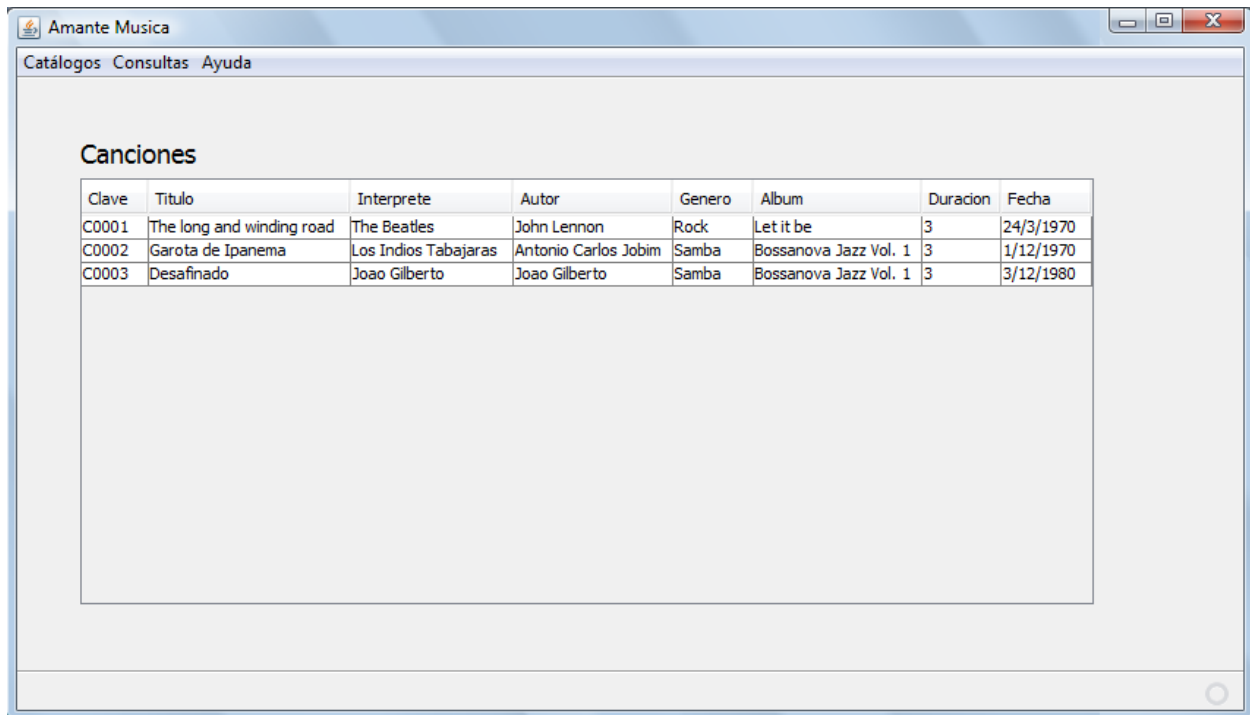


Figura 6.9. Listado de las calificaciones en el catálogo de calificaciones

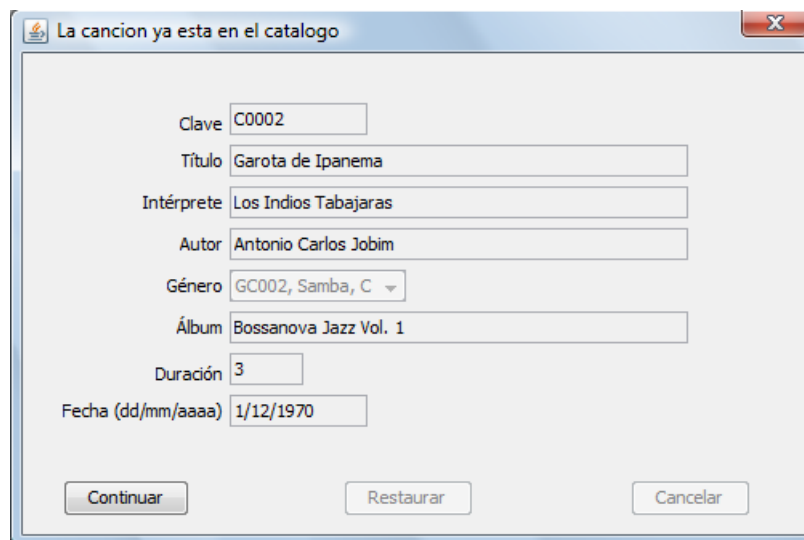


Figura 6.10 Cuadro de diálogoDlgCancion para mostrar los datos de una canción existente.

Para actualizar los datos de una canción, primero se captura la clave de una canción en forma similar a cuando se quiere agregar una canción. A continuación se busca la canción de la clave en el catálogo de canciones. Si la canción existe en el catálogo de canciones, se despliega el cuadro de diálogo mostrado en la figura 6.11 para modificar los datos de la canción:

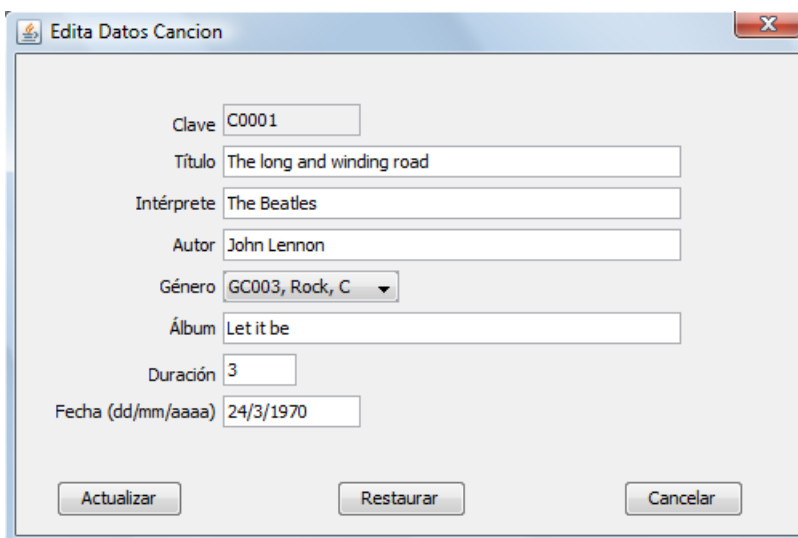


Figura 6.11. Cuadro de diálogo DlgCancion para modificar los datos de una canción.

Después de modificar los datos de una canción, el programa la actualiza en el catálogo de canciones y posteriormente despliega una tabla con las canciones del catálogo.

Si la canción no existe en el catálogo de canciones, el programa despliega un cuadro de diálogo con un mensaje de error, utilizando el método estático `showErrorMessageDialog()` de la clase `JOptionPane` que permite desplegar un mensaje utilizando un cuadro de diálogo como el mostrado en la figura 6.12:

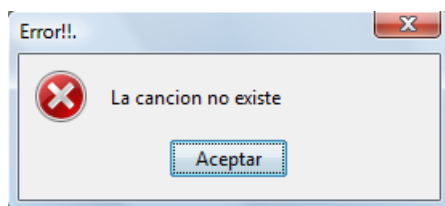


Figura 6.12 Cuadro de diálogo con mensaje de error.

Para eliminar los datos de una canción, primero se captura la clave de una canción en forma similar a cuando se quiere agregar una canción. A continuación se busca la canción de la clave en el catálogo de canciones. Si la canción existe en el catálogo de canciones, se despliega el cuadro de diálogo mostrado en la figura 6.13 para confirmar la operación.

Después de eliminar una canción, el programa despliega una tabla con las canciones del catálogo.

Si la canción no existe en el catálogo de canciones, el programa despliega un cuadro de diálogo con un mensaje de error similar al de la figura 6.12.

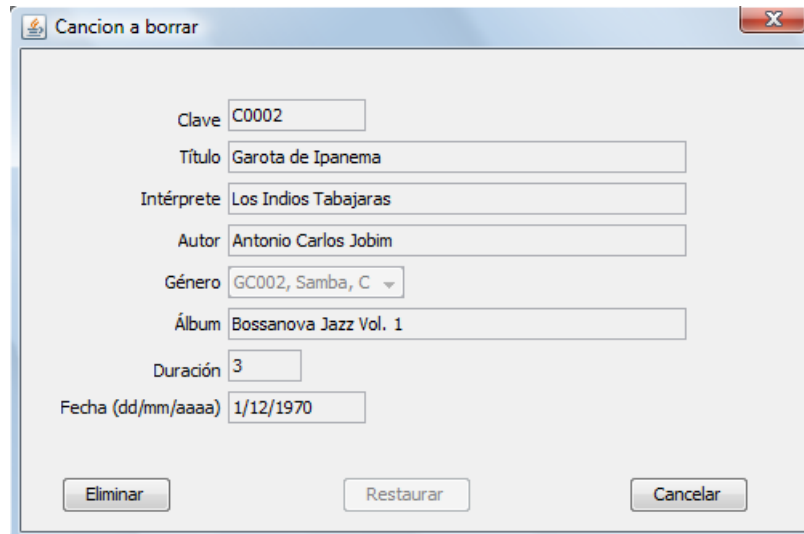


Figura 6.13. Cuadro de diálogo DlgCancion para confirmar la eliminación de una canción

Las figura 6.14 muestra las opciones de las diferentes consultas que se pueden realizar en el catálogo de canciones.

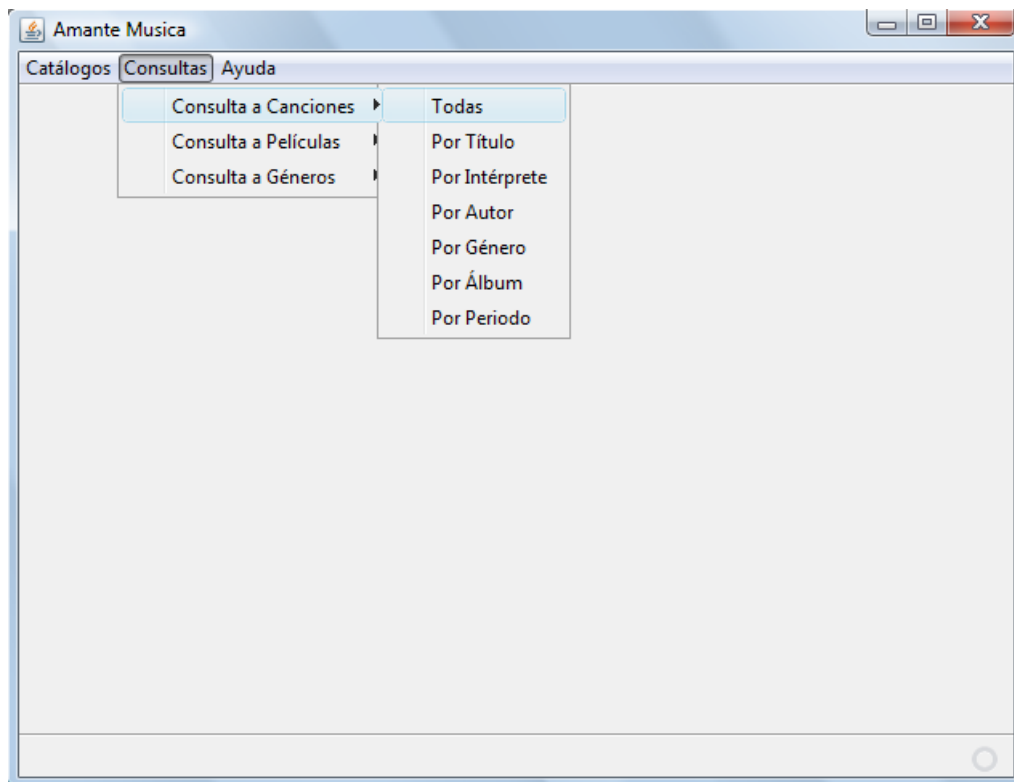


Figura 6.14. Selección de la opción de consultar todas las canciones

Cada consulta produce una tabla como la mostrada en la figura 6.9.

Para capturar, editar o desplegar los datos de un periodo se emplea el cuadro de diálogo mostrado en la figura 6.15:

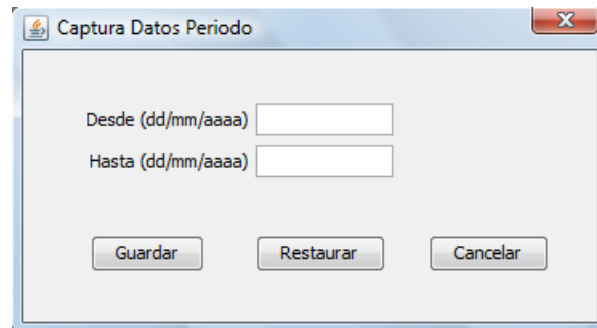


Figura 6.15. Cuadro de Diálogo DlgPeriodo