



**Área Académica de Computación y  
Electrónica  
Licenciatura en Ciencias Computacionales  
Programación Orientada a Objetos**

# **Cuadros de diálogo JOptionPane en Java**

Elaboraron:

M.C.C. Iliana Castillo Pérez

Dra. Verónica Martínez Lazcano

Dra. Yira Muñoz Sánchez

Fecha de elaboración: octubre 2023



# Abstract

Most of the programs that are developed require interaction with the user, either by requesting the input of data or by showing the user the required information. In the Java language there is a library that provides components called dialog boxes used for this purpose. This material describes each of the different types of dialogs and the way in which they can be configured.

## Keywords:

Dialog boxes, data input, information output, user interaction.



# Resumen

La mayoría de programas que se desarrollan requieren de la interacción con el usuario, ya sea solicitando el ingreso de datos o mostrándole a éste la información requerida. En el lenguaje Java existe una librería que proporciona componentes llamados cuadros de diálogo utilizados para éste fin. En este material se describen cada uno de los diferentes tipos de cuadros de diálogo y la forma en la que pueden ser configurados.

## Palabras clave:

Cuadros de diálogo, entrada de datos, salida de información, interacción con el usuario.

# Objetivo de Aprendizaje



Implementar cuadros de diálogo en los programas bajo el paradigma orientado a objetos, a través de la definición de sus características y su configuración, para interactuar con el usuario final de una forma más efectiva.



# Cuadros de diálogo

## Introducción

Los cuadros de diálogo o ventanas emergentes son útiles para interactuar con el usuario, ya sea solicitando el ingreso de datos o para mostrar mensajes con información, advertencias, verificación de información, mensajes de error, entre otros.





# Cuadros de diálogo

## Introducción

El lenguaje de programación Java, proporciona la clase **JOptionPane** para la creación y manejo de cuadros de diálogo, dicha clase se localiza en el paquete **javax.swing**. Para utilizarla es necesario importarla, por lo que se debe agregar la siguiente línea de código:

```
import javax.swing.JOptionPane;
```



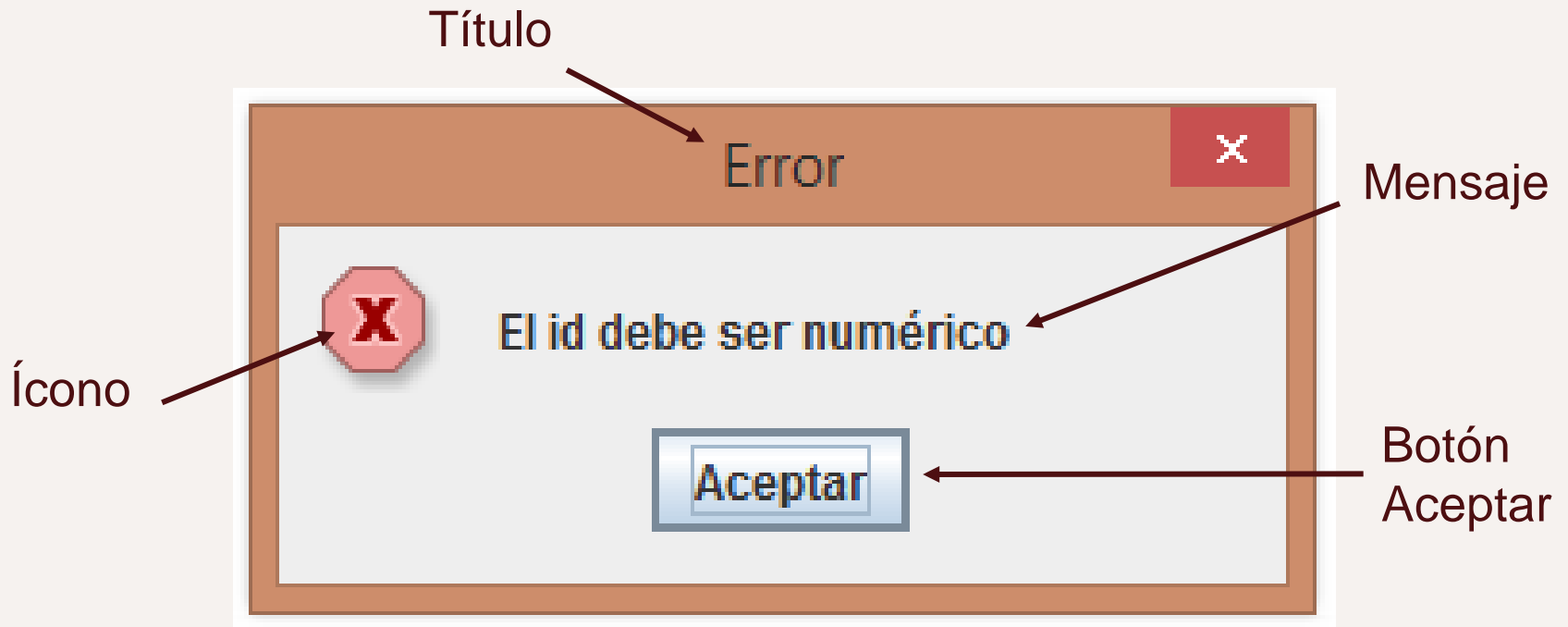
# Clase JOptionPane

La clase `JOptionPane` proporciona 4 métodos que determinarán el tipo y funcionalidad del cuadro de diálogo:

- `showMessageDialog()`
- `showInputDialog()`
- `showConfirmDialog()`
- `showOptionDialog()`

# showMessageDialog()

- El método `showMessageDialog` mostrará al usuario un cuadro de diálogo de tipo informativo con las siguientes características:





# showMessageDialog()

Este método no devuelve ningún valor.

Su sintaxis puede ser alguna de las siguientes:

- 1) `JOptionPane.showMessageDialog(Component componentePadre, Object mensaje);`
- 2) `JOptionPane.showMessageDialog(Component componentePadre, Object mensaje, String titulo, int tipoDeMensaje);`

# showMessageDialog()

```
3) JOptionPane.showMessageDialog(Component  
    componentePadre, Object mensaje, String  
    titulo, int tipoDeMensaje, Icon icono);
```

Este método puede recibir un mínimo de 2 parámetros: el componentePadre y el mensaje.



# Descripción de parámetros

- `Component componentePadre` -> indica el componente al que pertenece el cuadro de diálogo.
- `Object mensaje` -> mensaje de texto dirigido al usuario.
- `String titulo` -> título del cuadro de diálogo.
- `int tipoDeMensaje` -> definido por una constante de la clase `JOptionPane`.
- `Icon icono` -> la imagen del icono a mostrar.



# Ejemplo de showMessageDialog

```
JOptionPane.showMessageDialog(null,  
    "Bienvenido a esta aplicación");
```

El código anterior, genera el siguiente cuadro de diálogo:



## Otro ejemplo de showMessageDialog





```
JOptionPane.showMessageDialog(null,  
"Aplicación bajo licencia\n Creative  
commons", "Licencia", 0, new  
ImageIcon("C:\\\\imagenes\\\\icono-cc.png"));
```

El código anterior, genera el siguiente cuadro de diálogo:



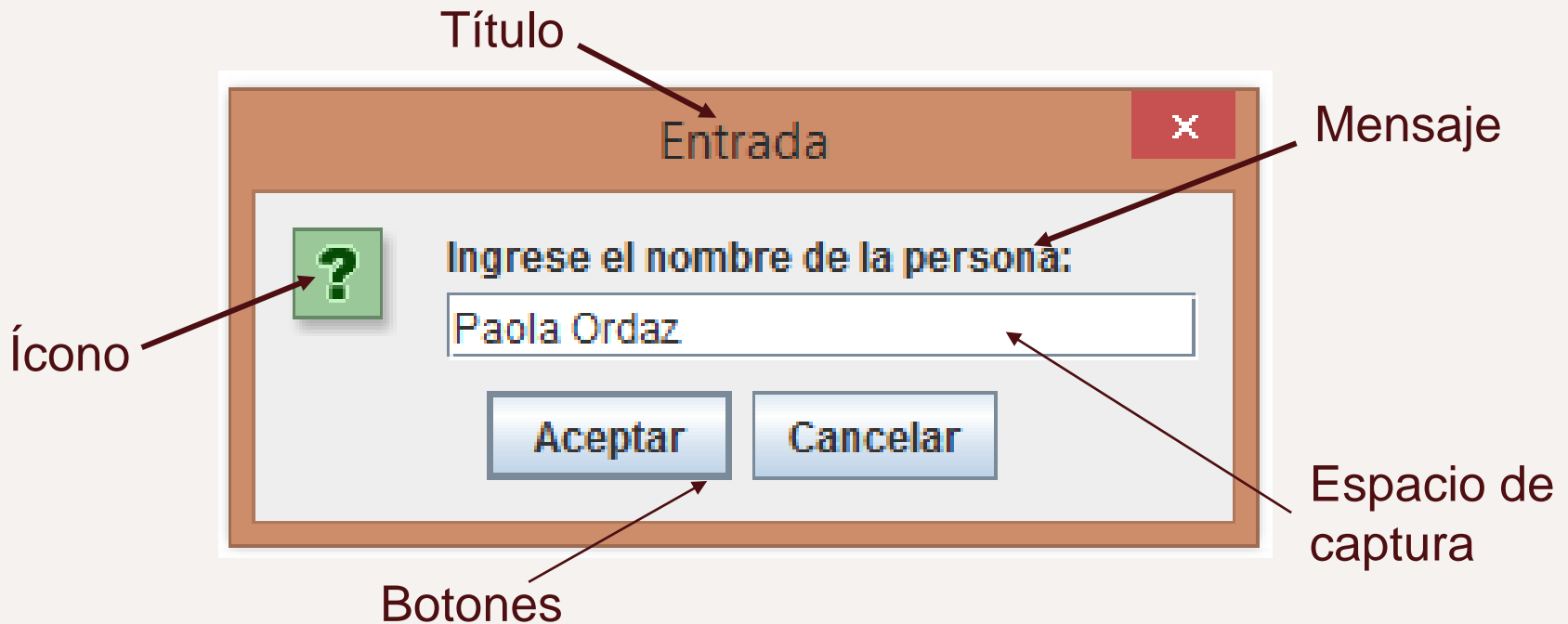
# Constantes de JOptionPane

- El parámetro `tipoDeMensaje` se pueden definir con una constante proporcionada por la clase `JOptionPane` y puede ser alguna de las siguientes:

Icono	Código	Valor constante
Sin icono	<code>JOptionPane.PLAIN_MESSAGE</code>	-1
	<code>JOptionPane.ERROR_MESSAGE</code>	0
	<code>JOptionPane.INFORMATION_MESSAGE</code>	1
	<code>JOptionPane.WARNING_MESSAGE</code>	2
	<code>JOptionPane.QUESTION_MESSAGE</code>	3

# showInputDialog()

- El método `showInputDialog` permite al usuario ingresar información mediante un cuadro de diálogo con las siguientes características:



# showInputDialog()

Este tipo de componente, devuelve un **String** o un **Object** con la información capturada por el usuario.

Su sintaxis puede ser alguna de las siguientes:

- 1) `JOptionPane.showInputDialog(Object mensaje);`  
*//devuelve un String*
- 2) `JOptionPane.showInputDialog(Component  
componentePadre, Object mensaje);` *//devuelve  
un String*

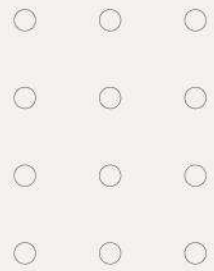




# showInputDialog()

- ○ ○
- ○ ○
- ○ ○
- ○ ○
- 3) `JOptionPane.showInputDialog(Object mensaje, Object valorDeSeleccionInicial); //devuelve un String`
- 4) `JOptionPane.showInputDialog(Component componentePadre, Object mensaje, Object valorDeSeleccionInicial); //devuelve un String`
- 5) `JOptionPane.showInputDialog(Component componentePadre, Object mensaje, String titulo, int tipoDeMensaje); //devuelve un String`





## showInputDialog()

6) `JOptionPane.showInputDialog(Component componentePadre, Object mensaje, String titulo, int tipoDeMensaje, Icon icono, Object[] valoresDeSeleccion, Object valorDeSeleccionInicial);` //devuelve un *Object*



# Descripción de parámetros

- `Component componentePadre` -> indica el componente al que pertenece el cuadro de diálogo.
- `Object mensaje` -> mensaje de texto dirigido al usuario.
- `String titulo` -> título del cuadro de diálogo.
- `int tipoDeMensaje` -> definido por una constante de la clase `JOptionPane`.
- `Icon icono` -> la imagen del icono a mostrar.



## Descripción de parámetros

- `Object[] valoresDeSeleccion` -> arreglo de objetos con las opciones a elegir.
- `Object valorDeSeleccionInicial` -> valor por default que inicializa el campo de entrada.

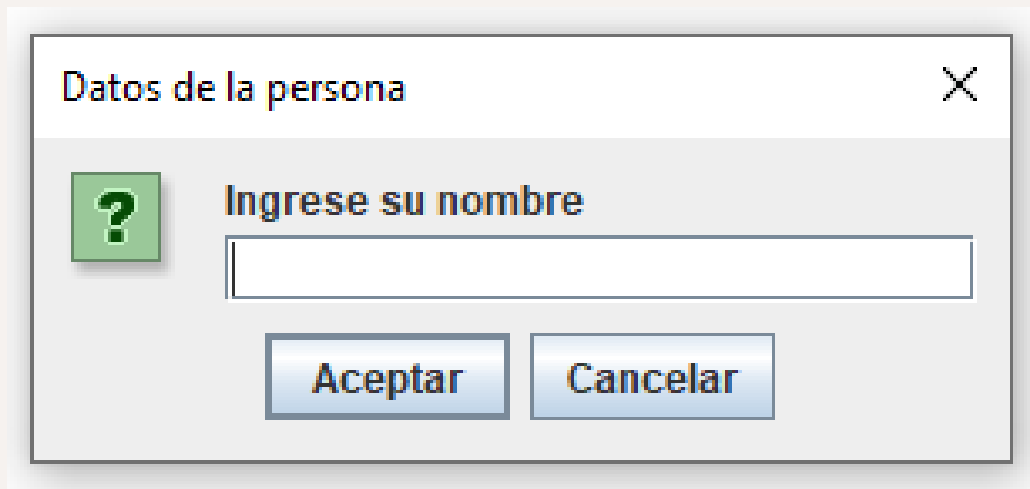
Éste método puede recibir como mínimo 1 parámetro: el mensaje.



## Ejemplo de showInputDialog

```
String nombre = JOptionPane.showInputDialog(null,  
    "Ingrese su nombre", "Datos de la persona",  
    JOptionPane.QUESTION_MESSAGE);
```

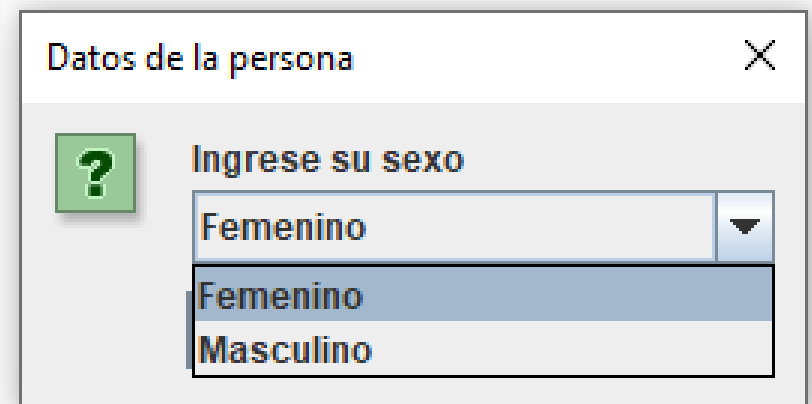
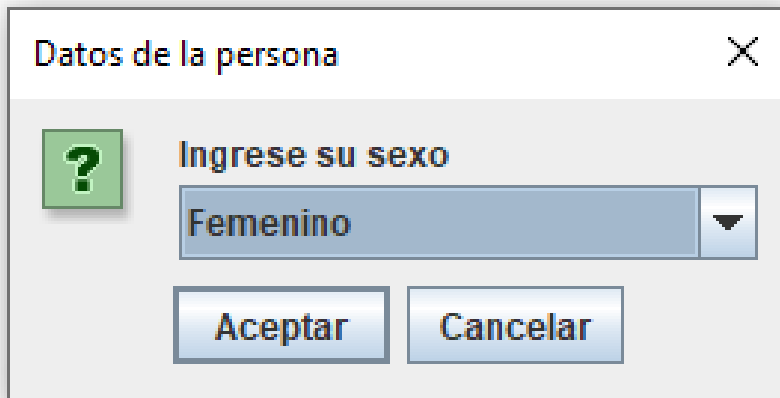
El código anterior, genera el siguiente cuadro de diálogo:



## Otro ejemplo de showInputDialog

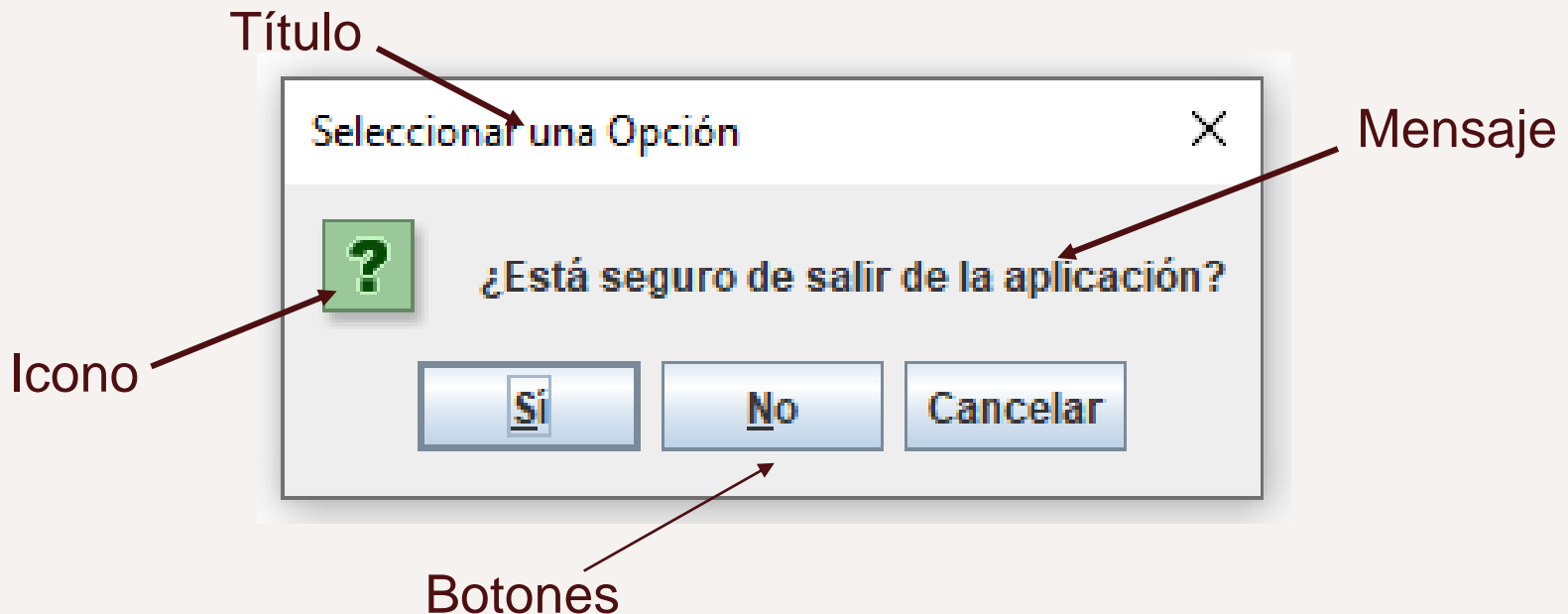
```
Object seleccion = JOptionPane.showInputDialog(null,
    "Ingrese su sexo", "Datos de la persona",
    JOptionPane.QUESTION_MESSAGE, null, new Object[]
    {"Femenino", "Masculino"}, "Femenino");
```

El código anterior, genera el siguiente cuadro de diálogo:



# showConfirmDialog()

- El método `showConfirmDialog` permite dar al usuario alternativas para ejecutar una instrucción a través de un cuadro de diálogo como el siguiente:



# showConfirmDialog()

Este tipo de componente, devuelve un **int** con la elección del usuario.

Su sintaxis puede ser alguna de las siguientes:

- 1) `JOptionPane.showConfirmDialog(Component componentePadre, Object mensaje);`  
*//devuelve un int*
- 2) `JOptionPane.showConfirmDialog(Component componentePadre, Object mensaje, String titulo, int tipoDeOpciones);` *//devuelve un int*



# showConfirmDialog()

- ○ ○
- ○ ○
- ○ ○
- ○ ○
- 3) `JOptionPane.showConfirmDialog(Component componentePadre, Object mensaje, String titulo, int tipoDeOpciones, int tipoDeMensaje);`
- 4) `JOptionPane.showConfirmDialog(Component componentePadre, Object mensaje, String titulo, int tipoDeOpciones, int tipoDeMensaje, Icon icono);`

Éste método puede recibir como mínimo 2 parámetros: el componente padre y el mensaje.



# Descripción de parámetros

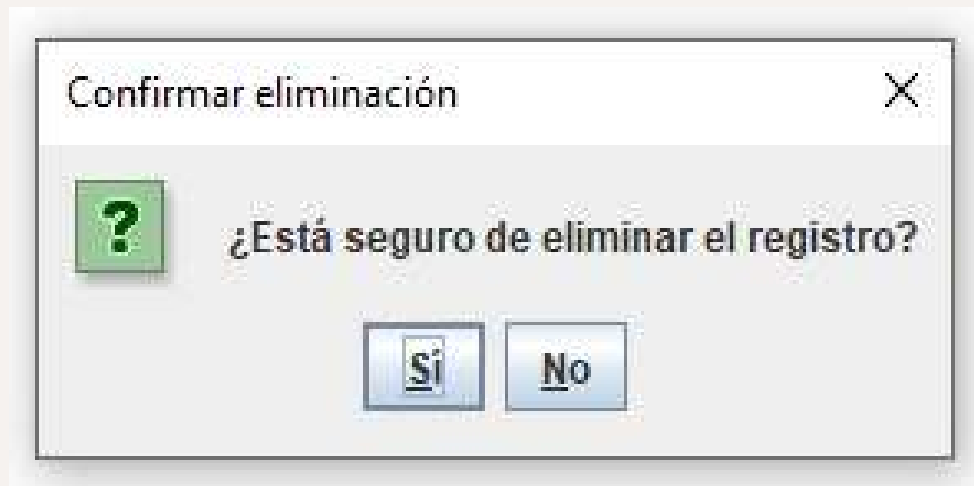
- `Component componentePadre` -> indica el componente al que pertenece el cuadro de diálogo.
- `Object mensaje` -> mensaje de texto dirigido al usuario.
- `String titulo` -> título del cuadro de diálogo.
- `int tipoDeOpciones` -> constante que indica las opciones a mostrar en el cuadro de diálogo.
- `int tipoDeMensaje` -> definido por una constante de la clase `JOptionPane`.
- `Icon icono` -> la imagen del icono a mostrar.



## Ejemplo de showConfirmDialog

```
int r = JOptionPane.showConfirmDialog(null,  
    "¿Está seguro de eliminar el registro?",  
    "Confirmar eliminación",  
    JOptionPane.YES_NO_OPTION);
```

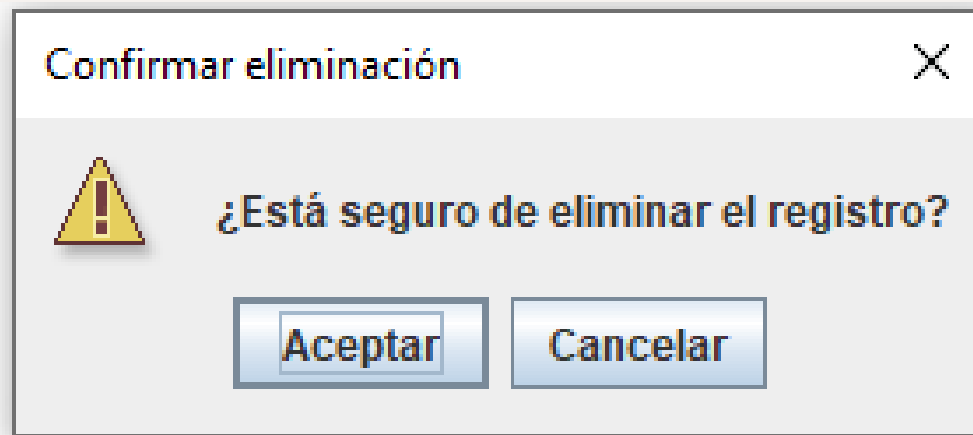
El código anterior, genera el siguiente cuadro de diálogo:



## Otro ejemplo de showConfirmDialog

```
int r = JOptionPane.showConfirmDialog(null,  
    "¿Está seguro de eliminar el registro?",  
    "Confirmar eliminación",  
    JOptionPane.OK_CANCEL_OPTION,  
    JOptionPane.WARNING_MESSAGE);
```

El código anterior, genera el siguiente cuadro de diálogo:



# showOptionDialog()

- ○ ○
- ○ ○
- ○ ○

○ El método `showOptionDialog` permite configurar el cuadro de diálogo a mostrar. Se basa en los 3 métodos anteriores.



# showOptionDialog()

Este tipo de componente, devuelve un **int** con la elección del usuario.

Su sintaxis es la siguiente:

```
1) JOptionPane.showOptionDialog(Component  
componentePadre, Object mensaje, String  
titulo, int tipoDeOpciones, int  
tipoDeMensaje, Icon icono, Object[]  
opciones, Object valorInicial);
```

# Descripción de parámetros

- `Component componentePadre` -> indica el componente en el que se muestra el cuadro de diálogo.
- `Object mensaje` -> mensaje de texto a mostrar.
- `String titulo` -> título del cuadro de diálogo.
- `int tipoDeOpciones` -> entero que designa las opciones disponibles en el cuadro de diálogo.
- `int tipoDeMensaje` -> definido por una constante de la clase `JOptionPane`, que determina el tipo de mensaje.
- `Icon icono` -> la imagen del icono a mostrar.



## Descripción de parámetros

- `Object[] opciones` -> arreglo de objetos con las opciones a elegir por el usuario.
- `Object valorInicial` -> valor por default que inicializa el campo de entrada para el cuadro de diálogo.

Éste método devolverá **CLOSE\_OPTION** en caso de que el usuario cierre el cuadro de diálogo.





# Ejemplo de showOptionDialog

```
String[] colores
```

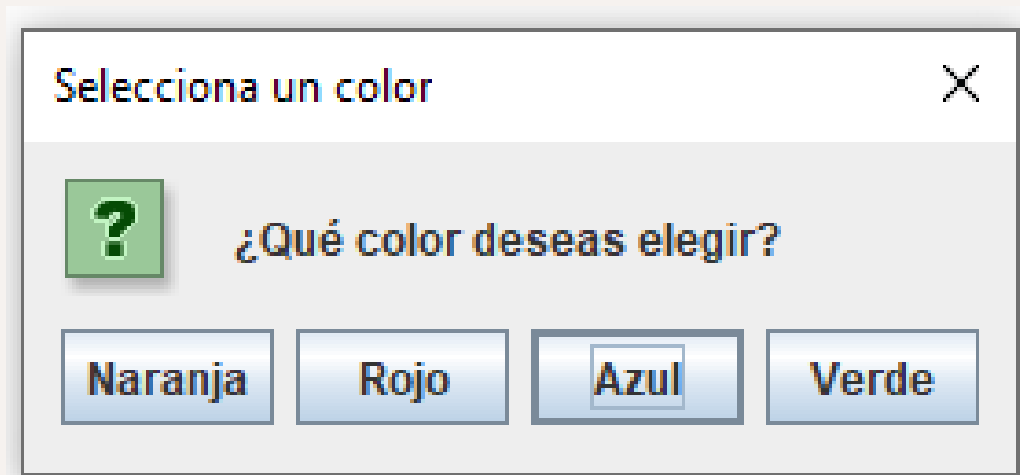
```
={"Naranja", "Rojo", "Azul", "Verde"};
```

```
int opcion = JOptionPane.showOptionDialog(null,  
    "?Qué color deseas elegir?", "Selecciona un  
    color", JOptionPane.DEFAULT_OPTION,  
    JOptionPane.QUESTION_MESSAGE, null, colores,  
    colores[2]);
```



# Ejemplo de showOptionDialog

El código de la diapositiva anterior, genera el siguiente cuadro de diálogo:



## Otro ejemplo de showDialog

```
String[] opciones = {"Brownie", "Pay", "Pastel"};

int postre = JOptionPane.showMessageDialog(null,
    "¿Qué postre te gustaría?", "Selecciona un
    postre", JOptionPane.DEFAULT_OPTION,
    JOptionPane.QUESTION_MESSAGE, new
    ImageIcon("C:\\imagenes\\Duke6.png"),
    opciones, opciones[0]);
```

Ejemplo tomado y modificado de: (McKenzie, 2022)



# Otro ejemplo de showOptionDialog

El código de la diapositiva anterior, genera el siguiente cuadro de diálogo:



# Bibliografía

- Ceballos, F. (2007), Java 2. Lenguaje y Aplicaciones, 1ra. Ed, Editorial Ra-Ma
- Chuidiang. (2014). JOptionPane y diálogos modales. Obtenido de Chuwiki:  
[https://chuidiang.org/index.php?title=JOptionPane\\_y\\_di%C3%A1logos\\_modales&redirect=no](https://chuidiang.org/index.php?title=JOptionPane_y_di%C3%A1logos_modales&redirect=no)
- Deitel, H. y Deitel, P. (2016). Como Programar en Java, 9na. Edición, Pearson Prentice Hall.
- Joyanes, L. (2011), Programación en Java 6. Algoritmos, programación orientada a objetos e interfaz gráfica de usuarios, McGraw-Hill.
- McKenzie, C. (2022). Java's JOptionPane showDialog by Example. Obtenido de TechTarget - TheServerSide-Your Enterprise Java Community:  
<https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/ Javas-JOptionPane-showOptionDialog-by-Example>

# Datos de contacto

M.C.C. Iliana Castillo Pérez

Profesora Investigadora

Área Académica de Computación y Electrónica

Instituto de Ciencias Básicas e Ingeniería

Universidad Autónoma del Estado de Hidalgo

Correo-e: [ilianac@uaeh.edu.mx](mailto:ilianac@uaeh.edu.mx)

Dra. Verónica Martínez Lazcano

Profesora Investigadora

Área Académica de Computación y Electrónica

Instituto de Ciencias Básicas e Ingeniería

Universidad Autónoma del Estado de Hidalgo

Correo-e: [vlazcano@uaeh.edu.mx](mailto:vlazcano@uaeh.edu.mx)

Dra. Yira Muñoz Sánchez

Profesora Investigadora

Escuela Superior de Cd. Sahagún

Universidad Autónoma del Estado de Hidalgo

Teléfono: 77171 72000 ext. 5300

Correo-e: [yira@uaeh.edu.mx](mailto:yira@uaeh.edu.mx)



The image features a vibrant orange background with a repeating pattern of large, overlapping geometric shapes, including circles and semi-circles. In the center, the text "LAEH" is displayed in a bold, white, sans-serif font. A thin white diagonal line runs through the letter 'A'. A registered trademark symbol (®) is positioned to the upper right of the 'H'.

**LAEH**®