

ALERTAS Y DIALOGOS EN JAVA

La clase JOptionPane es una clase que hereda de JComponent. Esta clase nos permitirá crear alertas o cuadros de diálogo simples para poder **solicitar o mostrar** información al usuario.

Los métodos que veremos son:

1. JOptionPane.showMessageDialog(...);
2. JOptionPane.showConfirmDialog(...);
3. JOptionPane.showInputDialog(...);
4. JOptionPane.showOptionDialog(...);

Cada uno de estos métodos presenta una particularidad distinta pero todos ellos nos muestran una ventana pop up que nos permitirá captar información del usuario.

1. showMessageDialog

JOptionPane.showMessageDialog (Component componentePadre, Object mensaje, String titulo, int tipoDeMensaje)

Nos sirve para mostrar información, por ejemplo alguna alerta que queremos hacerle al usuario. Veamos cuales son los principales argumentos del método.

componentePadre: es el Frame desde el cual lo llamamos. Si queremos lo podemos poner null de momento. Si hubiera un padre, el dialogo se mostraría sobre él.

mensaje: es lo que queremos que diga el cuadro de dialogo.

titulo: el título de la ventana.

tipoDeMensaje: son constantes que le dirán a java qué tipo de mensaje queremos mostrar. De acuerdo a esto serán los iconos que se mostrarán en el cuadro de dialogo.

Las opciones son:

- ERROR_MESSAGE
- INFORMATION_MESSAGE
- WARNING_MESSAGE
- QUESTION_MESSAGE
- PLAIN_MESSAGE

Si quisiéramos mostrar un icono personalizado, podemos agregarlo al final como un argumento más.

Ejemplo:

```
JOptionPane.showMessageDialog(null, "Acceso Denegado", "Error",  
JOptionPane.ERROR_MESSAGE);
```

2. showConfirmDialog

Este método sirve para pedirle al usuario una confirmación. Por ejemplo, una confirmación de salida del sistema.

```
Int respuesta =JOptionPane. showConfirmDialog(Component componentePadre,  
Object mensaje, String titulo, int tipoDeOpcion);
```

Lo anterior es la versión corta de los argumentos del método. La versión larga incluye el tipo de mensaje y el icono, por si queremos personalizarlo.

Los argumentos son idénticos a los del método anterior. Excepto por el tipo de opción, que es otra constante y los valores pueden ser:

- **DEFAULT_OPTION**
- **YES_NO_OPTION**
- **YES_NO_CANCEL_OPTION**
- **OK_CANCEL_OPTION**

Como vemos abajo, el método devuelve un entero que nos permitirá captar cuál es la opción elegida por el usuario. Los valores serán **0** para **Sí**, **1** para **No**, **2** para **Cancelar** y **-1** para el **cierre de la ventana**. Así podremos preguntar cuál es el valor devuelto y realizar la acción que deseamos.

Ejemplo:

```
int i = JOptionPane.showConfirmDialog (this,"¿Realmente Desea Salir de Hola  
Swing?", "Confirmar Salida", JOptionPane.YES_NO_OPTION);  
  
if (i== 0) {  
    System.exit(0);  
}
```

3. ShowInputDialog

Este método nos muestra una ventana donde podremos insertar un String. Por ejemplo, cuando queremos que el usuario inserte su nombre. La versión corta del método es:

String respuesta = JOptionPane.showInputDialog(Object mensaje)

Este método devolverá un String para poder utilizarlo después. La versión larga de los argumentos del método es similar a los anteriores.

Ejemplo:

```
String nombre =JOptionPane.showInputDialog("Inserte su Nombre");
```

También podemos crear un cuadro de dialogo que contenga un combo con las opciones predeterminadas que le queremos dar al usuario.

Ejemplo:

```
Object[] valoresPosibles = {"Pedro", "Juan", "Carlos" };
```

```
Object jefe = JOptionPane.showInputDialog(null, "Seleccione cuál es su Jefe  
Inmediato", "Seleccionar Jefe", JOptionPane. QUESTION_MESSAGE, null,  
valoresPosibles, valoresPosibles[0]);
```

El array de valores posibles nos muestra en un combo cuáles serán los jefes que podemos mostrar. El último argumento del método nos muestra cuál será la opción seleccionada por defecto

4. **showOptionDialog**

Con este método podemos crear cuadros de diálogos con botones personalizados. Está bien para personalizar los cuadros de dialogo.

El método tiene la forma:

int res = JOptionPane.showOptionDialog(Component componentePadre, Object mensaje, String titulo, int tipoDeOpcion, int tipoMensaje, Icon icono, Object[] botones, Object botonDefault)

Aquí lo único que varía con el resto, es el array de botones que vamos a tener. No hace falta que creamos botones, solo tenemos que poner cuál será el texto que saldrá en él.

```
Object[] botones = {"No, nada", "Un poquito", "Me estalla" };  
  
int i = JOptionPane.showOptionDialog(null, "¿Te duele la cabeza?", "ventanita",  
JOptionPane.DEFAULT_OPTION, JOptionPane.WARNING_MESSAGE,  
null, botones, botones[0]);  
  
System.out.println(i);
```

Después podemos tomar la respuesta como lo hacíamos con el `confirmDialog`.

Se puede consultar el API y <https://serprogramador.es/programando-mensajes-de-dialogo-en-java-parte-1/>