

# USO DE CUADRO DE DIALOGOS

**Carrera:** Computación

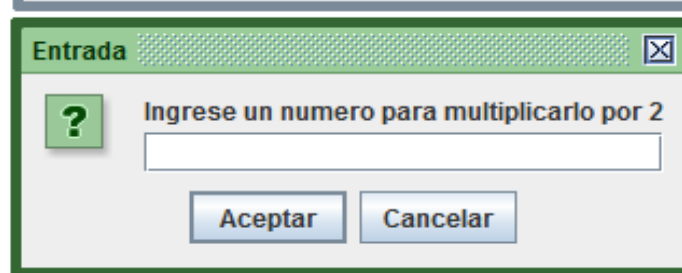
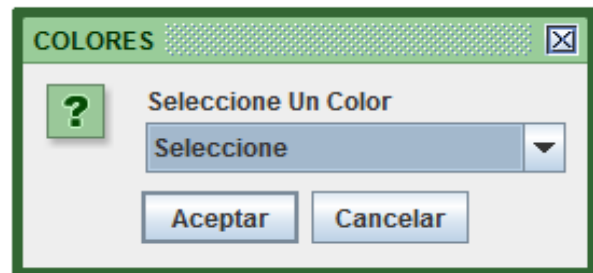
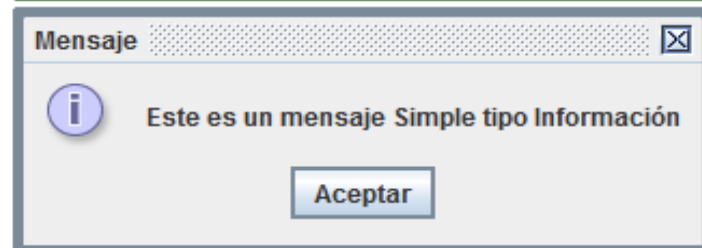
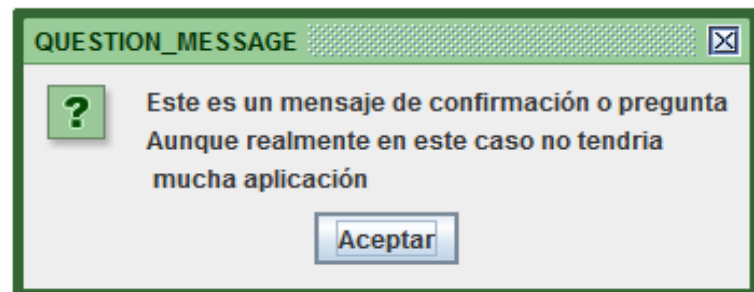
**Semestre:** III

**Nombre de Unidad Didáctica:** Análisis y técnicas de algoritmo.

**Logro Esperado:** Desarrolla aplicaciones haciendo uso del contenedor Swing Controls



# ¿Qué observas en la imagen?



# JOptionPane

Un diálogo es una ventana independiente también llamada subventana que pretende mostrar un aviso temporal al margen de la ventana de la aplicación principal de Swing. Los Diálogos más comunes son mostrar un mensaje de error o de advertencia a un usuario.



# Librerías

Para poder usar estos cuadros de dialogo hay que importar la librería necesaria para que esta se ejecute es:

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



# Tipos de Cuadros de Diálogos

Entre los tipos de cuadros dialogo encontraremos:

- `showConfirmDialog`
- `showInputDialog`
- `showMessageDialog`
- `showOptionDialog`.

Los mas usados son los tres primeros.

# ShowMessageDialog

Este cuadro de dialogo es ideal para mostrar información de cualquier tipo, este reemplaza el `System.out.print()`; La sintaxis es:

```
JOptionPane.showMessageDialog(null,"Bienvenidos");
```



## ShowInputDialog

Este cuadro de dialogo es ideal para ingresar datos, hay que tener en cuenta que todo lo que se reciba se tomara como un String, pero esto no es problema para trabajar con otros tipos de datos, en tal caso es necesario convertir la cadena que se recibe en el tipo de dato que necesitamos (int, Float, double, etc). La sintaxis es la siguiente:

```
String num = JOptionPane.showInputDialog("ingrese un numero");
```

```
JOptionPane.showMessageDialog(null, "El numero ingresado es: " + num);
```





## ShowConfirmDialog

Este cuadro de dialogo es de tipo entero, por lo tanto hay que crear una variable de tipo int para manejar las opciones que este cuadro de dialogo ofrece las cuales son: Si, No y Cancelar. Sintaxis es la siguiente:

```
int num = JOptionPane.showConfirmDialog(null, "Estas en java?");  
if(num == JOptionPane.YES_OPTION)  
    JOptionPane.showMessageDialog(null, "Has seleccionado SI.");  
else if(num == JOptionPane.NO_OPTION)  
    JOptionPane.showMessageDialog(null, "Has seleccionado NO.");
```





# ShowOptionDialog

Esta ventana se usa para poder personalizar los cuadros de diálogos con nuestras propias opciones.

# Componente Swing

**JLabel:** Un objeto de control JLabel permite dibujar en el formulario una etiqueta, entendiéndose como etiqueta una expresión estética que se quiere colocar. También es usado para mostrar los resultados de un proceso. Este componente se utiliza para desplegar textos o mensajes estéticos dentro de las formas, textos tales como encabezados, solicitud al usuario del programa para que proporcione algún dato o información.



# Componente Swing

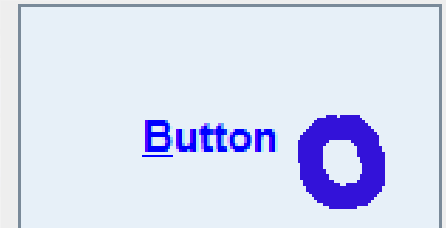
**JTextField:** Un objeto de control JTextField permite dibujar en el formulario un cuadro de texto, es decir, una caja que permite la introducción de un dato o valor. Este objeto es utilizado para el ingreso de datos. En Java este componente es un importante componente visual, su función principal es manejar todos los procesos de entrada y salida (input/output) al programa. Su principal propiedad es TEXT(), que es la propiedad que contiene el dato de tipo STRING que se manipulara en los programas o aplicaciones.



Text Field

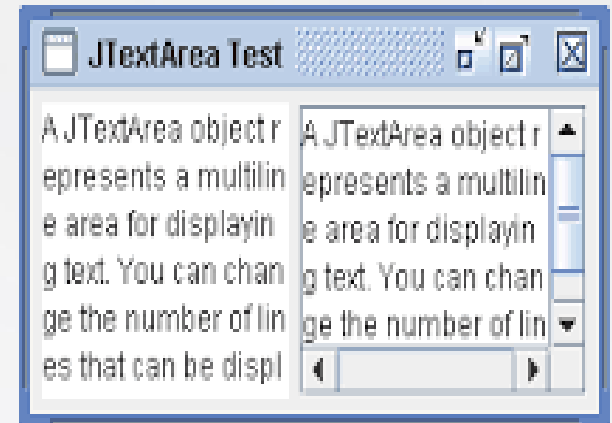
# Componente Swing

**JButton:** Es uno de los controles Swing más comunes y más utilizados, algunos de los usos básicos: deshabilitar y habilitar botones, establecer un botón por defecto que recibe foco al cargar el formulario y utilizar eventos para ejecutar métodos o líneas de códigos al hacer clic en el botón.



# Componente Swing

**JTextArea:** Es un componente básico del Swing de Java y su función principal es la de capturar texto ingresado desde teclado por el usuario. Sin embargo, entre sus métodos está el de asignarle texto para mostrarlo además.



# Método Get y Set

Un método es un conjunto de instrucciones encerradas entre 2 llaves, un nombre que lo identifica y parámetros (opcionales), devuelven algún tipo de datos. Sirve para realizar una función específica y su código es reutilizable.

## **Método Set ():**

Se usa para asignar datos. No devuelve nada.

## **Método Get ():**

Se usa para recuperar datos. Devuelve un valor.



# EJERCICIOS DE APLICACIÓN





# ¡GRACIAS!

