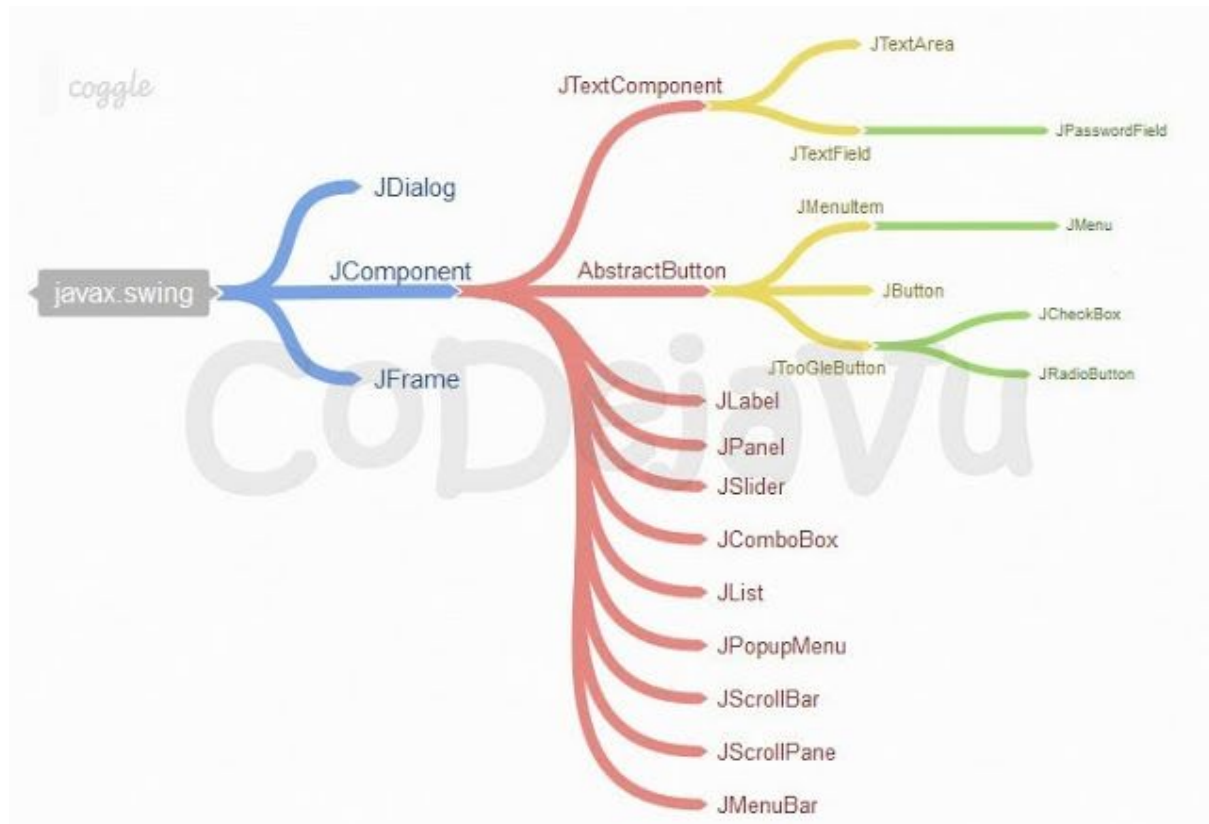


## JAVA [SWING BIBLIOTECA GRÁFICA]

**Java Swing**, es un paquete que provee herramientas o facilidades para la construcción de **GUI's** o interfaces Graficas de Usuario (graphical user interface).



Como vemos todos los componentes de **Swing** heredan del paquete **javax.swing**

**Para ver más en detalle busquemos el paquete swing en la api de java.**

Cuando vamos a construir aplicaciones utilizando Java Swing debemos tener al menos un contenedor que será la base para nuestra aplicación, es decir, será el lienzo donde pintaremos los demás componentes.

Normalmente podemos utilizar un JFrame o JDialog, estos serán la base para nuestra ventana y en ellos pintar los paneles, botones, cajas de texto, áreas entre otros.

Con **Swing** le daremos vida a nuestro sistema, ya que se crearán las vistas de la aplicación, por medio de las cuales el Usuario interactuará con el sistema, veremos que se tiene una gran cantidad de posibilidades para estructurar nuestros desarrollos, se pueden manejar los eventos de cada componente dependiendo de nuestras necesidades

### **Veamos un ejemplo:**

Creación de ventanas

- Es usual extender la clase `JFrame`, y realizar las operaciones de inicialización en su constructor:

```
public class MiVentana extends JFrame {  
    public MiVentana() {  
        super("Titulo de ventana");  
        setSize(400, 300);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```

```
public class VentanaTest {  
    public static void main(String[] args) {  
        MiVentana v = new MiVentana();  
        v.setVisible(true);  
    }  
}
```

Con el método **setSize** determino el ancho y alto de la ventana (width height).

Para colocarle un título a la ventana **setTitle("Titulo de ventana");**

Todas las ventanas que se crean por defecto están invisible por eso ponemos **v.setVisible(true);**

**setDefaultCloseOperation()** pertenece a la clase `frame` y nos permite cerrar nuestra ventana:

- Acciones de cierre:

- `JFrame.EXIT_ON_CLOSE`: Abandona aplicación.
- `JFrame.DISPOSE_ON_CLOSE`: Libera los recursos asociados a la ventana.
- `JFrame.DO_NOTHING_ON_CLOSE`: No hace nada.
- `JFrame.HIDE_ON_CLOSE`: Cierra la ventana, sin liberar sus recursos.

También podemos crear ventanas de la siguiente manera:

```
import javax.swing.*;

public class VentanaTest {
    public static void main(String[] args) {
        JFrame f = new JFrame("Titulo de ventana");
        f.setSize(400, 300);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setVisible(true);
    }
}
```

Podemos cambiar la posición de la ventana cuando se ejecuta ya que por defecto viene en la coordenada (0,0):

**setLocation(int x, int y)** donde hacemos uso de las coordenadas (x,y). Ejemplo: setLocation(100,200).

Nota: las medidas están en px.

**setBounds(int x, int y, int ancho, int largo):** Este método nos permite hacer lo mismo que hace setLocation y setSize juntos.

**setLocationRelativeTo(null)** establece la ventana en el centro de la pantalla.

# Componentes de una ventana



**.JButton**

**JLabel**

**.JTextField**

**JCheckBox**

**JRadioButton**

- Tras crear uno de estos componentes con new, ha de añadirse al contentPane de la ventana correspondiente mediante su método add.

```
public class MiVentana extends JFrame {  
    public MiVentana() {  
        super("Título de ventana");  
        setSize(400, 300);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        Container cp = getContentPane();  
        cp.setLayout(new FlowLayout());  
        JLabel etiqueta = new JLabel("Nombre: ");  
        JTextField texto = new JTextField(20);  
        JButton boton = new JButton("Saludar");  
        cp.add(etiqueta);  
        cp.add(texto);  
        cp.add(boton);  
    }  
}
```

## Interfaces complejas: JPanel

Un panel es un componente con un layout manager propio, y que puede contener varios componentes en su interior.

- Constructor: JPanel()



Es recomendable para organizar nuestros componentes usar paneles. y no agregar directamente nuestros componentes a la ventana como en el ejemplo anterior.

```
public MiVentana3() {
    super("Añadir usuario");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // Panel de fecha
    JPanel panelFecha = new JPanel();
    panelFecha.setLayout(new FlowLayout());
    panelFecha.add(new JTextField(2));
    panelFecha.add(new JLabel("/"));
    panelFecha.add(new JTextField(2));
    panelFecha.add(new JLabel("/"));
    panelFecha.add(new JTextField(2));
    // Panel de datos
    JPanel panelDatos = new JPanel();
    GridLayout gl = new GridLayout(3,2,0,5);
    panelDatos.setLayout(gl);
    panelDatos.add(new JLabel("Nombre:"));
    panelDatos.add(new JTextField(10));
    panelDatos.add(new JLabel("DNI:"));
    panelDatos.add(new JTextField(10));
    panelDatos.add(new JLabel("Fecha de nacimiento: "));
    panelDatos.add(panelFecha);
    ...
}
```

### **Veamos un ejemplo completo:**

<http://francho.org/2009/04/13/java-calculadora-sencilla-en-modo-grafico/>