

## Ejemplos de FlowLayout

Recordamos que el FlowLayout va a hacer que todos los componentes queden en fila y hace que todos quepan en la ventana en la medida de lo posible.

Ejemplo 1.- Vamos a crear un JFrame con 10 botones y los vamos a alinear a la izquierda y con un espacio entre ellos de 3 px horizontal y 3px vertical.

Ejemplo 2.- Vamos a crear un JFrame con 15 botones y los vamos a alinear a la derecha y con un espacio entre ellos de 20px horizontal y 30 px vertical.

Ejemplo 3.- Vamos a crear un JFrame con 8 botones y los vamos a centrar con la separación por defecto (5px horizontal y vertical)

Ejemplo 4.- Vamos a crear un JFrame con un botón que se llame botonEnviar y cuyo texto sea “enviar”. Un textfield que se llame cajaTexto y contenga el valor “” (Blancos) y una JLabel que se llame etiqueta y que contenga “Escribe tu nombre”. Ahora vamos a crear un flowLayout Centrado y con separación de 10 px tanto horizontal como en vertical.

Vamos a hacer más grande la ventana para ir jugando y viendo que los componentes se quedan siempre centrados.

Ejemplo 5.- Dado el siguiente código:

```
import java.awt.FlowLayout;

import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.WindowConstants;

public class PruebaFlowLayout {
    public static void main (String [] args)
    {
        JFrame v = new JFrame();

        // Se pone el FlowLayout en la ventana
        v.getContentPane().setLayout(new FlowLayout());

        // Se instancian los componentes
```

```

        JButton boton = new JButton("boton");
        JTextField textField = new JTextField(10);
        JCheckBox checkBox = new JCheckBox("check box");
        JLabel etiqueta = new JLabel("Etiqueta");

        // Se añaden los componentes a la ventana
        v.getContentPane().add(boton);
        v.getContentPane().add(textField);
        v.getContentPane().add(checkBox);
        v.getContentPane().add(etiqueta);

        // Se muestra la ventana

v.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        // pack() hace que la ventana coja el tamaño
necesario para mostrar
        // los componentes que hay en ella.
        v.pack();
        v.setVisible(true);
    }
}

```

- 1) Explicar el código. ¿Qué hace cada cosa?
- 2) Transformar este código para que haga lo mismo en Netbeans con un JFrame.