

Documento Explicativo – Aplicación GUI “Cartel”

1. Descripción general

El programa es una aplicación gráfica desarrollada en **Java Swing** que muestra una ventana con un cartel personalizable.

El usuario puede:

- Escribir un texto para el cartel.
- Cambiar el color de fondo.
- Modificar el texto haciendo clic sobre el cartel.
- Salir de la aplicación mediante un botón.

La aplicación está organizada en tres clases dentro del paquete **guiada**:

- **Main**
- **Ventana**
- **PanelCartel**

2. Clase Main

```
public class Main {  
    public static void main(String[] args) {  
        new Ventana();  
    }  
}
```

Función:

Es el punto de entrada del programa.

Su único objetivo es **crear una instancia de la clase **Ventana****, lo que inicia la aplicación gráfica.

3. Clase Ventana

```
public class Ventana extends JFrame
```

Función:

Representa la ventana principal del programa.

Configuración principal:

- `setTitle("sí")` → Define el título de la ventana.
- `setSize(900, 400)` → Define el tamaño.
- `setDefaultCloseOperation(EXIT_ON_CLOSE)` → Cierra el programa al cerrar la ventana.
- `setLocationRelativeTo(null)` → Centra la ventana en pantalla.
- Crea e inserta un objeto `PanelCartel`.
- `setVisible(true)` → Hace visible la ventana.

Esta clase actúa como contenedor principal del panel donde se dibuja el cartel.

4. Clase PanelCartel

```
public class PanelCartel extends JPanel implements ActionListener
```

Es la clase más importante del programa.

Se encarga de:

- Dibujar el cartel.
- Gestionar los botones.
- Detectar clics del ratón.
- Cambiar el color de fondo.

- Modificar el texto.

4.1 Variables principales

- `String texto` → Almacena el texto que se mostrará en el cartel.
- `JButton btnColor` → Botón para cambiar el color.
- `JButton btnTexto` → Botón para cambiar el texto.
- `JButton btnSalir` → Botón para cerrar el programa.

4.2 Constructor

En el constructor se realizan varias acciones:

1 Solicitud inicial del texto

Al iniciar el programa se muestra un `JOptionPane` para que el usuario escriba el texto del cartel.

Si el usuario cancela o no escribe nada, se asigna el texto por defecto:

`"Cartelsintexto"`

2 Creación de botones

Se crean tres botones:

- Cambiar Color
- Cambiar Texto
- Salir

Se añaden al panel inferior usando un `BorderLayout` en la posición `SOUTH`.

3 Evento de ratón

Se añade un `MouseListener` que detecta cuando el usuario hace clic sobre la zona aproximada donde está el texto (coordenadas específicas).

Si el clic ocurre dentro de esa zona:

- Se vuelve a pedir un nuevo texto.
- Se actualiza la variable.
- Se llama a `repaint()` para redibujar el panel.

4 Color inicial

El fondo del panel comienza siendo rojo:

```
setBackground(Color.RED);
```

4.3 Método `actionPerformed`

Este método se ejecuta cuando se pulsa alguno de los botones.

♦ Botón Salir

```
System.exit(0);
```

Cierra la aplicación correctamente.

♦ Botón Cambiar Texto

- Muestra un `JOptionPane`.
- Guarda el nuevo texto.
- Llama a `repaint()` para actualizar el dibujo.

♦ Botón Cambiar Color

- Genera valores RGB aleatorios usando `Random`.
- Crea un nuevo objeto `Color`.
- Cambia el fondo del panel con `setBackground()`.

4.4 Método `paintComponent`

```
protected void paintComponent(Graphics g)
```

Este método se encarga de dibujar el contenido gráfico del panel.

Acciones que realiza:

1. Llama a `super.paintComponent(g)` para limpiar el panel.
2. Establece una fuente de texto.
3. Dibuja un óvalo (simulando un cartel).
4. Dibuja el texto dentro del óvalo.
5. Dibuja el nombre "Alberto" en otra posición.

Este método se ejecuta automáticamente cada vez que se llama a `repaint()`.

5. Funcionamiento general paso a paso

1. Se ejecuta `Main`.
2. Se crea la ventana (`Ventana`).
3. Se crea el panel (`PanelCartel`).
4. Se solicita el texto inicial.

5. Se muestra el cartel con fondo rojo.

6. El usuario puede:

- Cambiar el texto con el botón o haciendo clic en el cartel.
- Cambiar el color de fondo.
- Salir del programa.