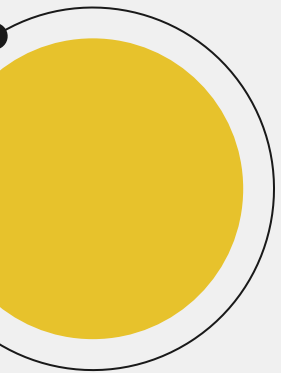




Java Swing

Interfaces gráficas



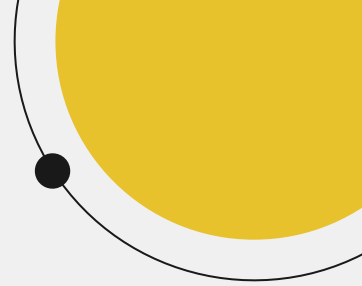


Interfaces graficas (GUI)

Son sistemas de interacción que permiten a los usuarios interactuar con una aplicación mediante la manipulación de elementos visuales en la pantalla, como botones, menús, cuadros de texto y ventanas.



Applet



Es un pequeño programa que se puede incrustar en el código HTML de una web para implementar una función determinada en una página web.

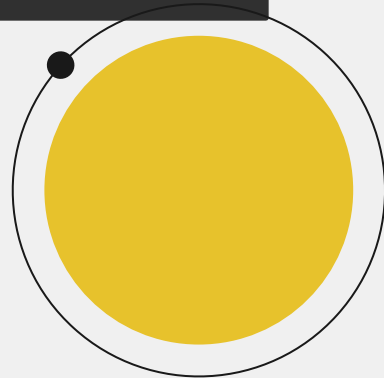
Sus principales características son:


- Son programas muy pequeños pues se deben ejecutar en página web por lo que su tamaño influye en el tiempo de transferencia.
- Hacen uso de las clases AWT y Swing lo que les proporciona gran operatividad y flexibilidad a la hora de desarrollar funciones.
- Actúan como componentes de la web que se ejecutan cuando son llamados desde el código HTML.





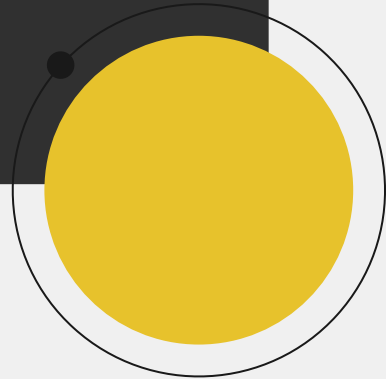
```
import java.applet.*;  
public class miApplet extends Applet {  
    //variables y métodos  
}
```





```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Color;

public class MiApplet extends Applet {
    public void paint(Graphics g) {
        g.setColor(Color.blue);
        g.drawString("!Hola, mundo!", 50, 50);
    }
}
```



Java AWT (Abstract Window Toolkit)

Es una biblioteca que permite hacer interfaces gráficas mediante artefactos de interacción con el usuario, como botones, menús, texto, botones para selección, barras de deslizamiento, etc.

El paquete `java.awt` proporciona una serie de clases e interfaces para realizar programas que se ejecuten en un entorno gráfico.

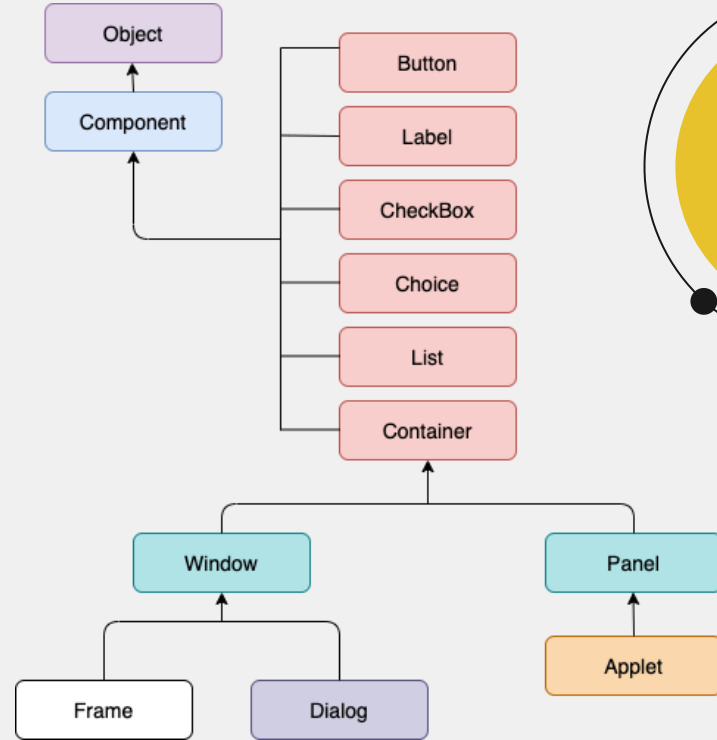


Diagrama de jerarquía de la clase Java AWT

Java.awt



Listado de algunas de las clases de java.awt

Component	Superclase abstracta de varios componentes del AWT. Tiene un método public Graphics getGraphics() para obtener el contexto gráfico sobre el que dibujar figuras o componentes. Declara más de 200 métodos.
Container	Subclase de la anterior que puede contener otros componentes. Tiene un método que nos permite añadir un componente: Component add(Component comp).
Color	Clase que representa un color.
Graphics	Encapsula el contexto gráfico que es utilizado por varios métodos para mostrar la salida en una ventana. Tiene además una serie de métodos para dibujar figuras.
Point	Clase para representar las coordenadas cartesianas "x" e "y"
JPanel	Representa un área rectangular sobre la que la aplicación puede dibujar.



Java Swing

Es una biblioteca de gráficos más moderna y sofisticada basada en AWT, pero ofrece una amplia gama de componentes personalizables, desde componentes básicos hasta componentes más complejos, como componentes de tablas, árboles y gráficos.



Java.swing

El paquete javax.swing proporciona una serie de clases e interfaces que amplían la funcionalidad del anterior. Están escritos en Java y son independientes de la plataforma.

La clase de este paquete es:

JFrame	Clase que representa una ventana. El método <code>Container.getContentPane()</code> debe utilizarse para obtener un objeto del tipo <code>Container</code> que se utilizará para añadir componentes a la ventana.
---------------	---

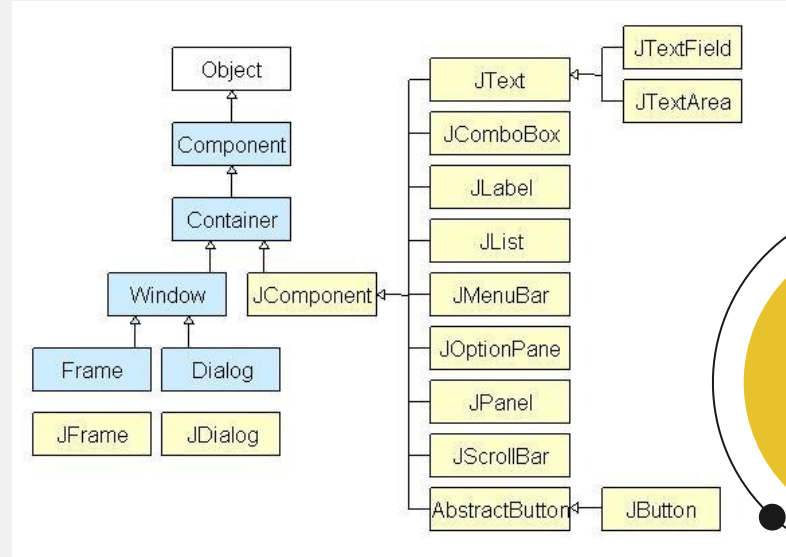
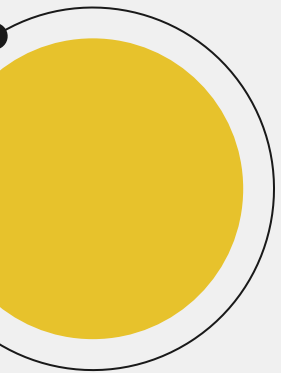


Diagrama de jerarquía de la clase Java Swing

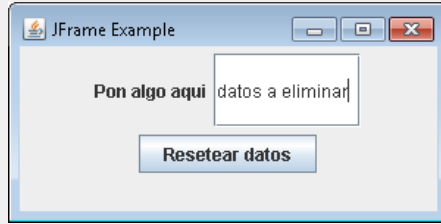


Contenedores

Los contenedores son componentes que permiten almacenar, alojar o contener otros elementos gráficos.



Contenedores



JFrame

Representa la ventana Principal de nuestra aplicación swing, y en el podemos alojar otros contenedores.



JDialog

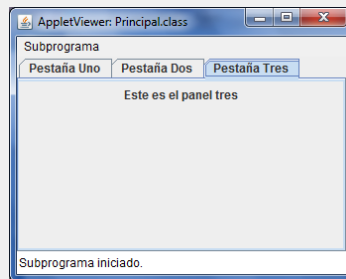
Ventana de tipo de diálogo, se utiliza para mostrar mensajes y otras acciones del usuario. Puede ser el contenedor principal, aunque se recomienda utilizarlo como ventana secundaria.

Contenedores



JPanel

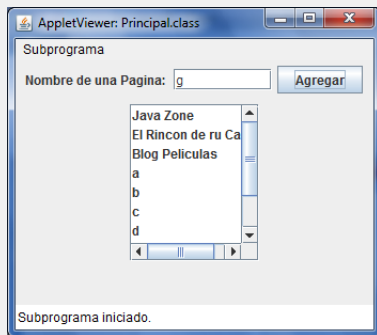
Es un contenedor que se utiliza para organizar y agrupar componentes relacionados en la interfaz de usuario. Este puede contener otros paneles, etiquetas, campos de texto, botones, etc.



JTabbedPane

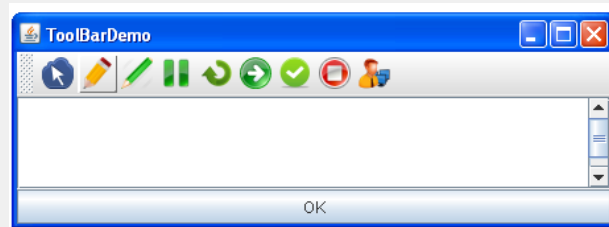
Permite crear pestañas en nuestra ventana, cada pestaña es un contenedor independiente donde podemos colocar paneles u otros elementos.

Contenedores



JScrollPane

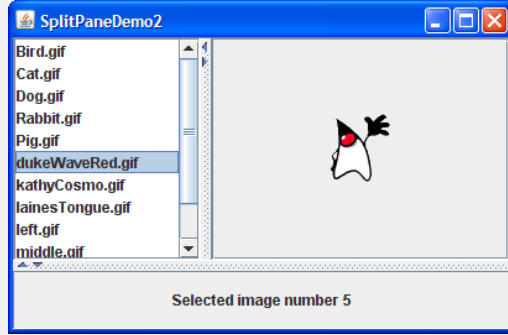
Es un contenedor para agregar barras de desplazamiento a otros componentes, como paneles y áreas de texto.



JToolBar

Es un contenedor que se utiliza para agregar herramientas, botones de acceso, check, radios, campos, etc. rápido a una aplicación.

Contenedores



JSplitPane

Permite la creación de un contenedor dividido en 2 secciones.



JDesktopPane

Permite crear ventanas dentro de una ventana principal.

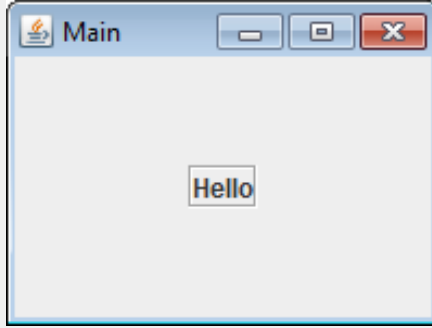


Componentes

Los componentes son estos elementos (botones, etiquetas, campos de texto, tablas, menús, entre otros) que permiten brindar una interacción con el usuario del sistema.

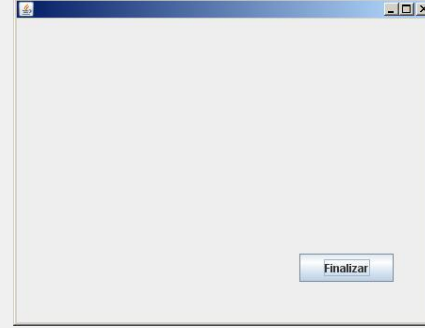


Contenedores



JLabel

Es una etiqueta que muestra un texto o una imagen en la interfaz de usuario.

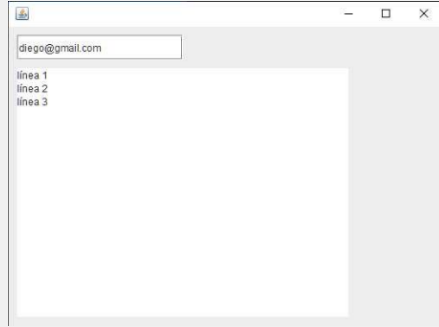


JButton

Es un botón en el que se puede hacer clic para realizar una acción.

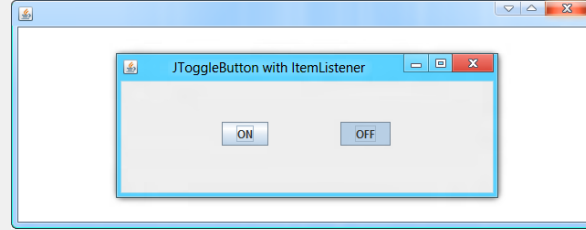


Contenedores



JTextArea

Permite vincular un área de texto donde el usuario ingresara información o simplemente para presentar cadenas de texto.



JToggleButton

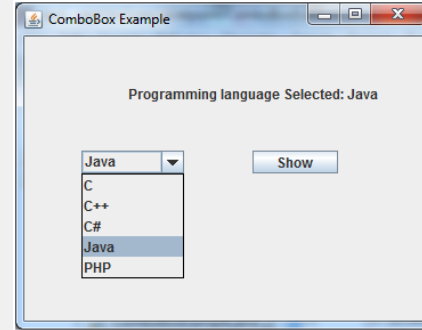
Botón que al oprimirlo se quedará presionado hasta que se ejecute otro evento.

Contenedores



JTextField

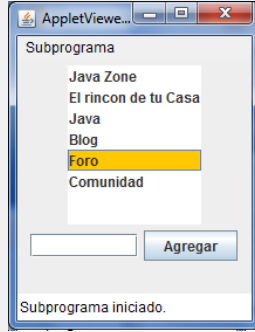
Es una caja que permite introducir un campo de texto simple.



JComboBox

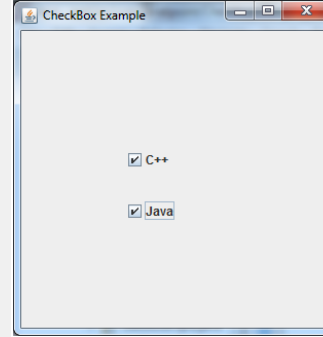
Es un componente que se utiliza para permitir que el usuario seleccione una opción de una lista desplegable.

Contenedores



JList

Es un componente que se utiliza para mostrar una lista de elementos para que el usuario seleccione.



JCheckBox

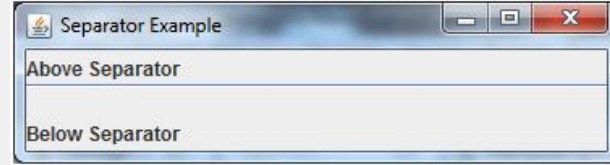
Es un componente de selección que se utiliza para permitir que el usuario seleccione opciones múltiples.

Contenedores



JRadioButton

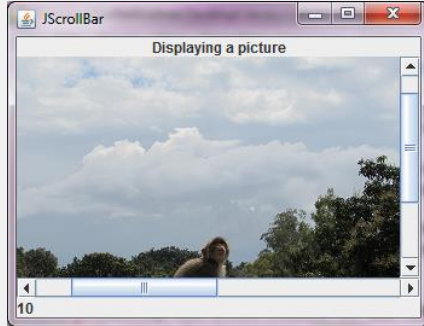
Es un componente de selección que se utiliza para permitir que el usuario seleccione una opción de una lista.



JSeparator

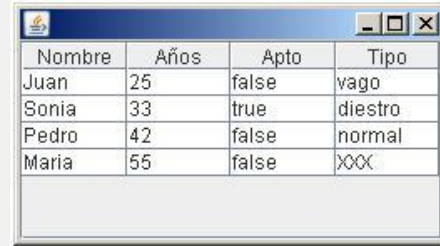
Permite separar opciones, es una barra simple.

Contenedores



JScrollbar

Es un componente que permite mostrar barras de desplazamiento, generalmente para áreas de texto o paneles cuyo contenido es más grande que el tamaño del componente.

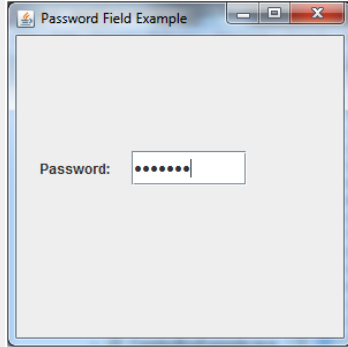


Nombre	Años	Apto	Tipo
Juan	25	false	vago
Sonia	33	true	diestro
Pedro	42	false	normal
Maria	55	false	XXX

JTable

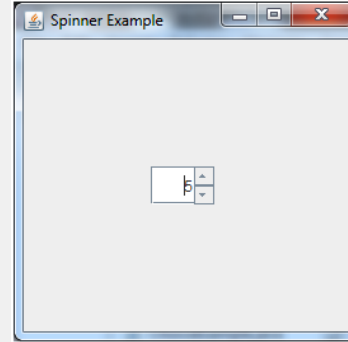
Es un componente que muestra datos en forma de tabla con filas y columnas.

Contenedores



JPasswordField

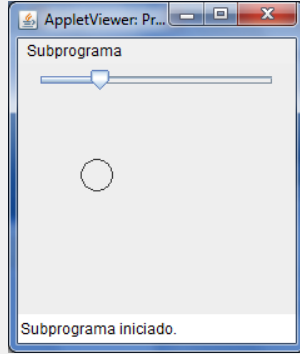
Campo de texto que oculta los caracteres ingresados.



JSpinner

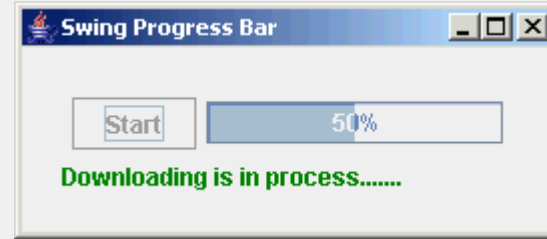
permite vincular una caja de texto con botones integrados para seleccionar algún valor.

Contenedores



JSlider

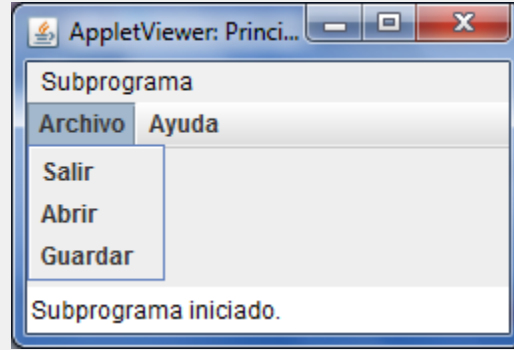
Permite vincular un Deslizador en nuestra ventana.



JProgressBar

Establece una barra de progreso.

Contenedores



JSlider

Se utilizan para crear menús desplegables en la interfaz de usuario.

Ejercicio

Cree un programa que genere una interfaz que nos salude



Ejercicio

Elabora un programa que dentro del JFrame que realice una evaluación 2 números y visualice como resultado el mayor de los 2 números.

Pautas del ejercicio

- Crea una clase llamada "Numbers" que extienda de la clase "JFrame" de Swing.
- Defina dos variables enteras, "numberOne" Y "numberTwo" para almacenar los números ingresados por el usuario.
- Crea cuatro objetos JLabel y dos objetos JTextField para mostrar el texto y permitir que el usuario ingrese los datos.
- Crea un objeto JButton con el texto "Calcular" que se utilizara para iniciar el calculo del número más grande.
- Crea un objeto JLabel para mostrar el resultado del calculo
- Crea un ActionListener para el botón "Calcular" que lee los números ingresados por el usuario, los compare y muestra el numero mas grande en el JLabel de resultado.
- Define en el método Main que crea una instancia de la clase "Numbers", hace visible la ventana, establece el tamaño y la ubicación en la pantalla.

Video tutorial

<https://www.youtube.com/watch?v=Fc4uFeMXBS8&t=1012s>





JApplet

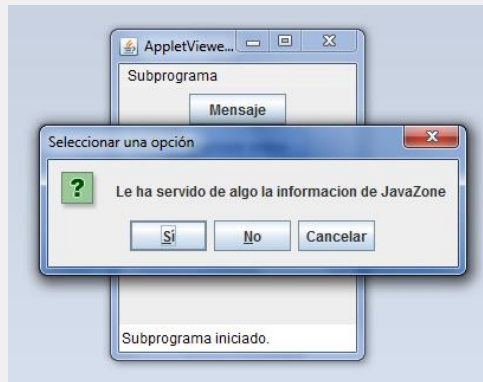
Es una extensión de la clase de applet de Java que permite a los desarrolladores crear applets y aplicaciones de escritorio más complejas y avanzadas utilizando el lenguaje de programación Java.





JOptionPane

Es una clase de java que muestra cuadros de diálogo, lo que permite mostrar mensajes, solicitar entrada del usuario y realizar acciones específicas.





Eventos

Los eventos son acciones que ocurren en un programa que el programa puede detectar y procesar. Los eventos pueden ser generados por el usuario (por ejemplo, haciendo clic en un botón o escribiendo en un cuadro de texto), o por el sistema operativo u otros componentes del programa.




ActionEvent



Es un evento que ocurre cuando un usuario realiza una acción en un componente interactivo en una interfaz gráfica de usuario. Se utiliza para procesar las acciones del usuario, como hacer clic en un botón o presionar "Entrar" en un campo de texto, y realizar una acción apropiada en respuesta a esa acción del usuario.







```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class EjemploActionEvent extends JFrame implements ActionListener {
    JButton boton;

    public EjemploActionEvent() {
        setTitle("Ejemplo ActionEvent");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        boton = new JButton("Haz clic aquí");
        boton.addActionListener(this);
    }
}
```





```
add(boton, BorderLayout.CENTER);
```

```
setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent e) {
```

```
    if (e.getSource() == boton) {
```

```
        JOptionPane.showMessageDialog(this, "¡Hola, mundo!");
```

```
    }
```

```
}
```

```
public static void main(String[] args) {
```

```
    new EjemploActionEvent();
```

```
}
```

```
}
```






ActionListener

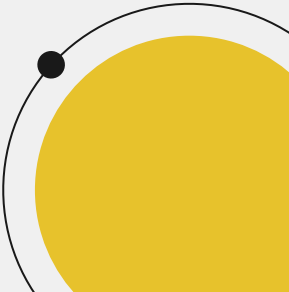
Es un controlador de eventos que ejecuta una tarea cuando una determinada acción se lleva a cabo por el usuario. Esta acción puede ser cualquier cosa que el usuario pueda hacer, como mover el ratón o pulsar una tecla en el teclado.

- Es una interfaz del grupo de los Listeners (escuchadores)
- Se usa para manejar y detectar eventos de la acción (ActionEvent)
- Tiene un solo método



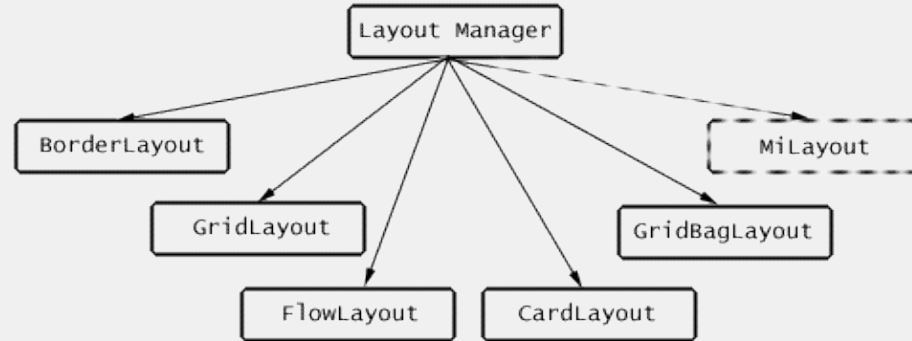


```
Public void actionPerformed(ActionEvent e){  
    //Accion a realizar  
}
```



LayoutManagers

Se utiliza para definir la apariencia y el diseño de los componentes en la GUI, lo que garantiza que los componentes se muestren de forma correcta y ordenada, independientemente del tamaño de la ventana o el contenedor.





FlowLayout

Organiza los elementos en filas horizontales o verticales. Los componentes cambian de tamaño automáticamente a su contenedor y se pueden agregar múltiples paneles FlowLayout para organizar los componentes en filas o columnas.





BorderLayout

Divide el contenedor en cinco zonas: norte, sur, este, oeste y centro. Cada área puede contener un componente, y el componente se redimensiona automáticamente cuando se redimensiona la ventana.





BoxLayout

Permite colocar los componentes a lo largo del eje X o del eje Y. También posibilita que los componentes ocupen diferente espacio a lo largo del eje principal.





CardLayout

Es utilizado cuando se necesita una zona de la ventana que permita colocar distintos componentes en esa misma zona. Este suele ir asociado con botones de selección, de tal modo que cada selección determina el panel (grupo de componentes) que se presentarán.

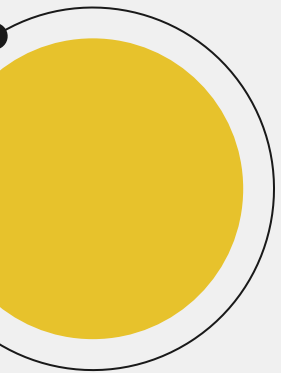




GridLayout

Tiene la función de organiza los componentes en una cuadrícula con filas y columnas del mismo tamaño. Al crear un GridLayout, especifique el número de filas y columnas, y los componentes están espaciados uniformemente en la cuadrícula.





GridBagLayout

Es igual que la composición de GridLayout, con la diferencia que los componentes no necesitan tener el mismo tamaño. Es quizá el controlador de posicionamiento más sofisticado de los que actualmente soporta AWT.



Ejercicio

En base a los parentidos, realiza un programa que haga uso del Layout GridLayout para Crear un Formulario de Inicio de Sesión.

