

# Unidad 1

## Elementos de Interfaces Gráficas

---

### Elementos de una Interfaz Gráfica

Una Interfaz Gráfica de Usuario (GUI, Graphical User Interface) es el conjunto de componentes gráficos que posibilitan la interacción entre el usuario y la aplicación: ventanas, botones, listas, listas desplegables, cajas de diálogo, campos de texto, etc.

Desde el punto de vista de Java, toda interfaz gráfica tiene:

- **Componentes:** Objetos que representan un elemento gráfico (botón, cuadro de texto).
- **Contenedores:** Objetos que contienen componentes (Ventanas, Cuadros de Dialogo).
- **Administradores de diseño:** Objetos que controlan la forma en que se ubican los componentes dentro de un contenedor.
- **Contexto Gráfico:** Área de un componente gráfico donde se pueden dibujar primitivas gráficas (líneas, curvas) y colocar imágenes.
- **Eventos:** Objetos que representan un cambio en un componente, generalmente producido por el usuario al realizar alguna operación.

### Componentes gráficos

Un componente es un objeto que tiene una representación gráfica que puede ser presentada en una pantalla y puede interactuar con el usuario. La clase abstracta Control contiene los métodos comunes para todos los componentes gráficos.

Los desarrolladores cuentan con tres alternativas de herramientas para construir interfaces gráficas en Java. La más antigua, AWT; la más utilizada y con el mayor tiempo en la preferencia de los desarrolladores, SWING; y la más reciente, JavaFX, que tiene cambios importantes en el paradigma de construcción de interfaces.

A continuación se presentan componentes de las tres alternativas.

Tabla 1.1: Métodos más utilizados en un componente gráfico

Método	Descripción
setBackground(Background b)	Define fondo, color o imagen, del componente.
setCursor(Cursor c)	Define el cursor a mostrar sobre el componente.
setBorder(Border b)	Define los bordes del componente.

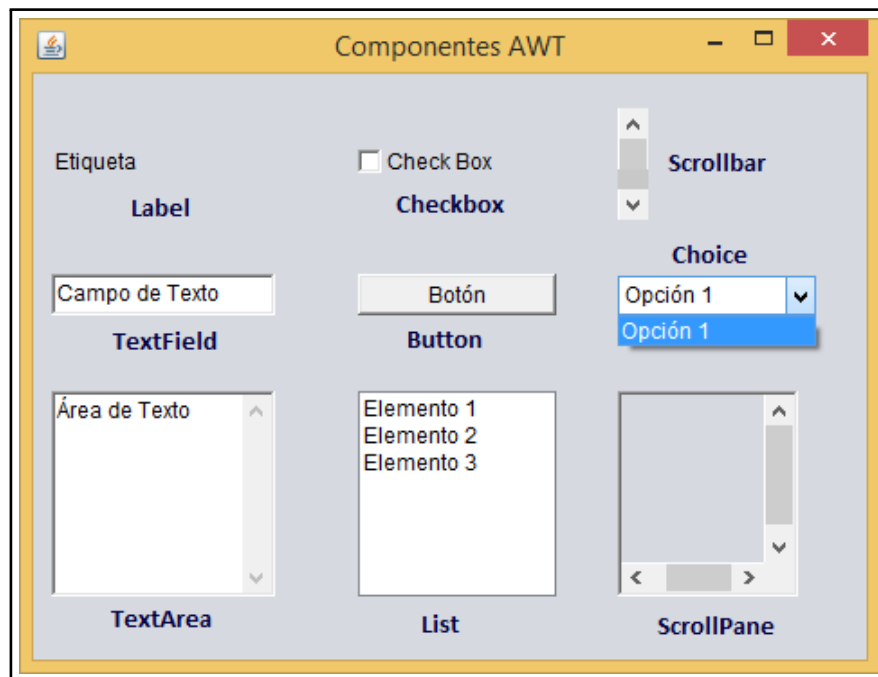


Figura 1.1: Algunos componentes de AWT

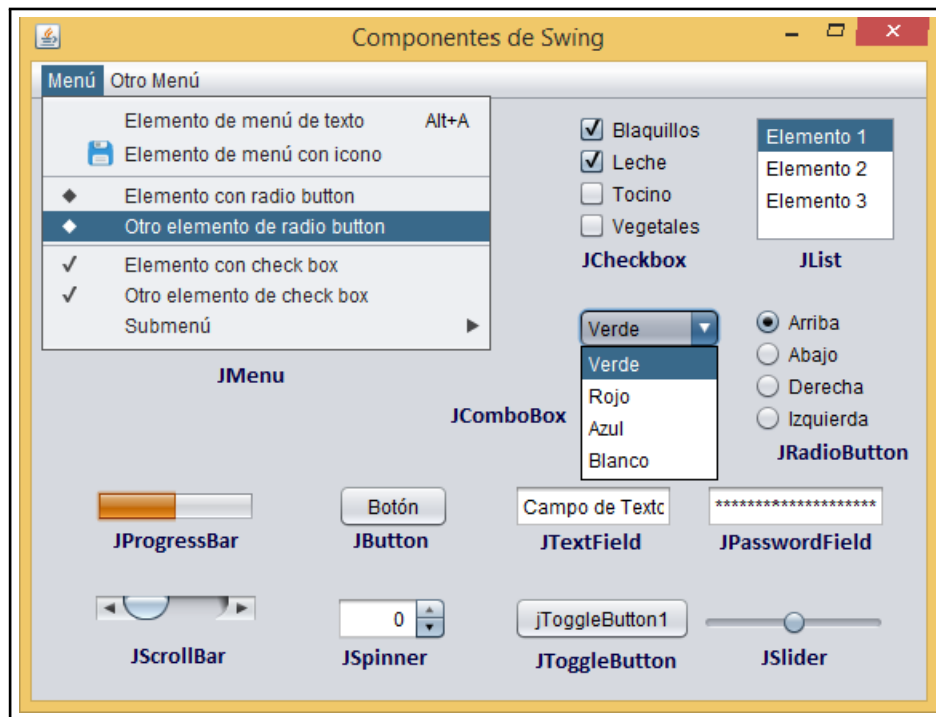


Figura 1.2: Algunos componentes de Swing

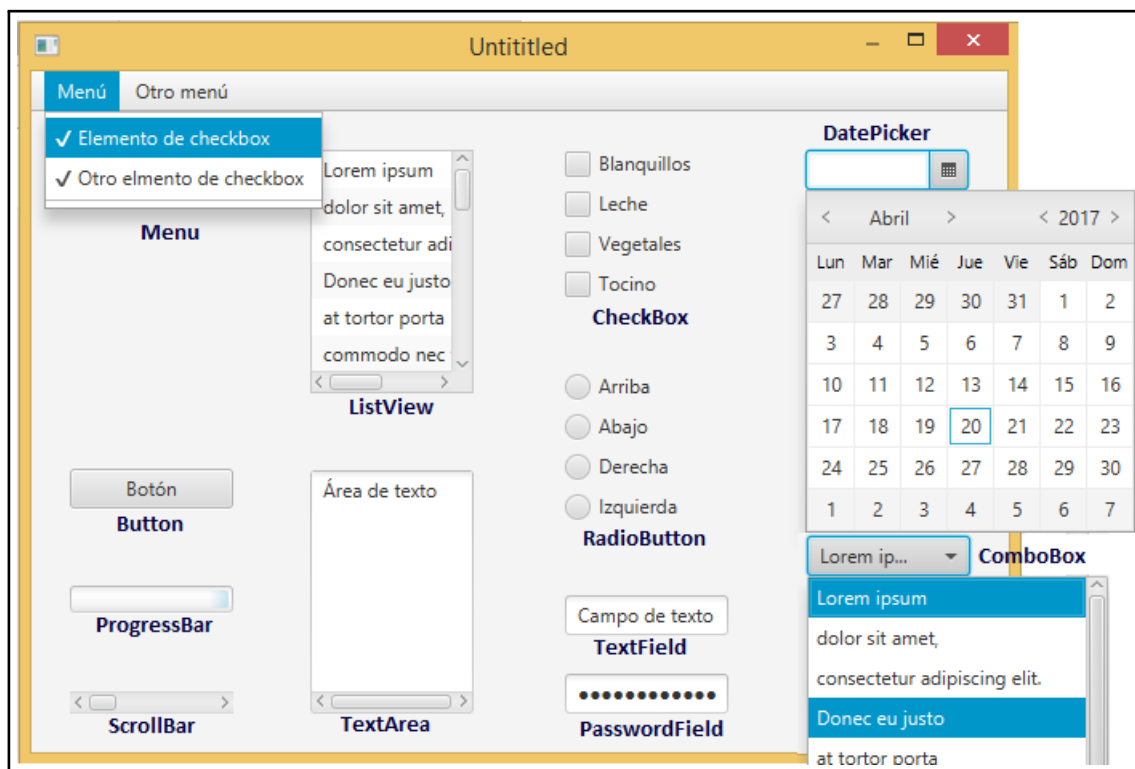


Figura 1.3: Algunos componentes de JavaFX

## Contenedores gráficos

Un contenedor es un componente gráfico que tiene la capacidad de almacenar otros componentes gráficos.

Tabla 1.2: Métodos más utilizados en contenedores

Método	Descripción
setScene(Scene value)	Define la escena que se mostrará en el contenedor.
show()	Hace visible el contenedor.
sizeToScene()	Redimensiona el contenedor al tamaño de su contenido.

## Administradores de diseño

Un administrador de diseño (layout) es un objeto que se encarga de controlar la ubicación y distribución de los componentes dentro de un contenedor. Un layout se considera otro nodo dentro del grafo de escena de una interfaz gráfica, y más que modificar el comportamiento de los contenedores principales son contenedores en sí.

## Diseño de una interfaz gráfica

Una interfaz gráfica común tiene los siguientes elementos:



Figura 1.4: Elementos comunes de una interfaz gráfica de usuario

El diseño de una interfaz gráfica debe ser jerárquico. La ventana estará construida por los siguientes elementos:

- Una barra de menú (opcional).
- Un contenedor gráfico para los componentes que definen la interfaz gráfica.
- Un objeto que maneje los eventos de la interfaz.

### Barra de Menú

Es un área de la interfaz de usuario que indica y presenta las opciones o herramientas de una aplicación dispuestas en menús desplegables. En la mayoría de entornos de escritorio los diferentes menús presentes en estas barras pueden ser desplegados por medio de atajos de teclado, al mantener presionada la tecla ALT y la tecla correspondiente a la letra subrayada en la barra de menú.

### Contenedor gráfico

Define el área donde se construirá la interfaz gráfica. Este contenedor sirve como organizador de los componentes gráficos.

### Manejador de Eventos

Cuando un usuario utiliza una interfaz gráfica, realiza cambios sobre los componentes (pulsar un botón, capturar sobre un campo, etc). Estos cambios se reflejan en objetos denominados eventos.

Los eventos deben ser atendidos para que la interfaz tenga el comportamiento esperado por el usuario, por lo cual toda interfaz gráfica debe utilizar un elemento manejador de eventos.

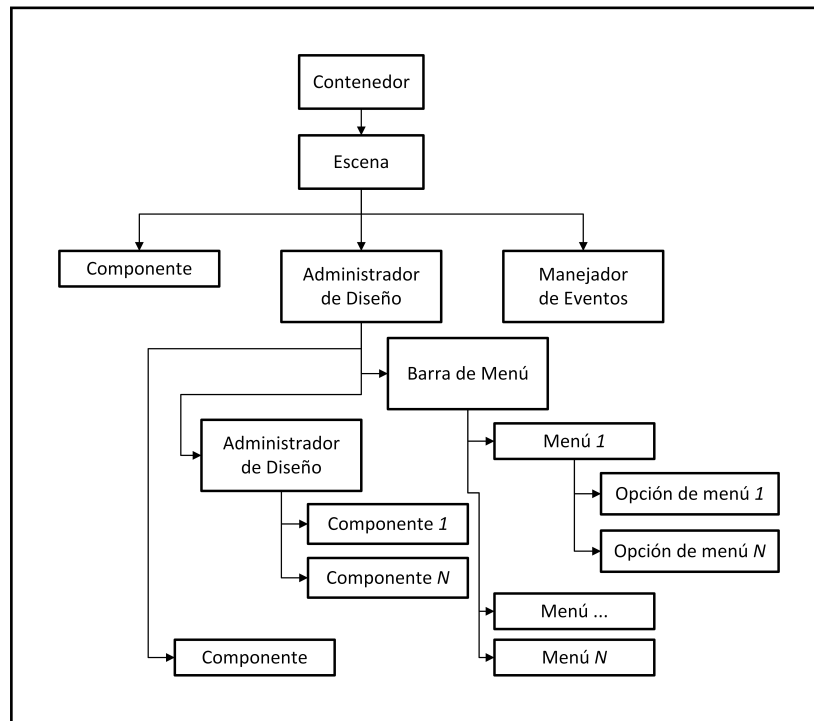


Figura 1.5: Estructura general de una interfaz gráfica de usuario

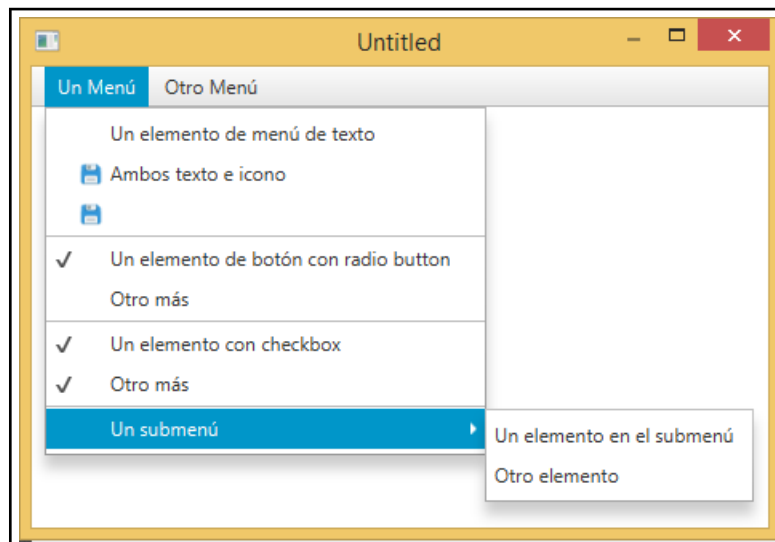


Figura 1.6: Ejemplo de una barra de menú y varias opciones de menú



Figura 1.7: Ejemplo de un contenedor con varios componentes gráficos

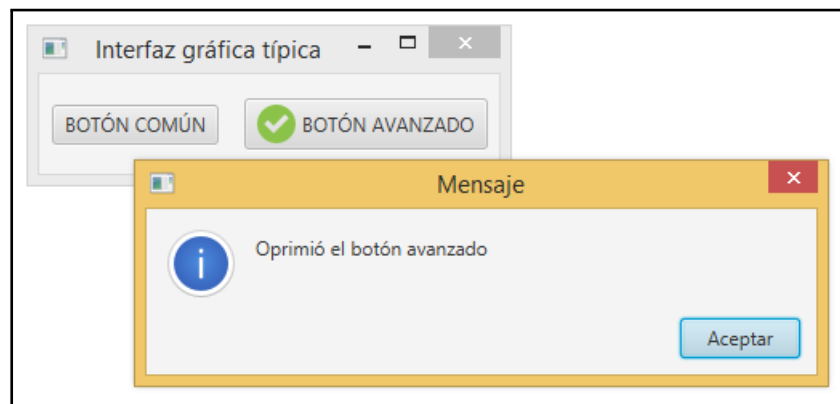


Figura 1.8: Ejemplo de un manejo de evento. Al oprimir el botón, se despliega una ventana de diálogo

## Estructura de clases para una interfaz gráfica

En este curso, cada aplicación gráfica estará diseñada dentro de un paquete, y tendrá una clase aplicación (donde se define el método main), y una o varias clases instanciables que definen los menús, los contenedores y los demás elementos necesarios para la interfaz.

La clase aplicación creará una instancia del **contenedor principal** (la ventana), definirá su configuración (tamaño, ubicación), le asignará la barra de menú correspondiente (si esta existe) y el contenedor para los componentes gráficos. Esta clase también creará los objetos que atenderán los eventos producidos en la aplicación para el comportamiento adecuado de esta.

Código 1.1: Creación de una ventana simple

```
import javafx.application.Application;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Ventana extends Application {

    @Override
    public void start(Stage primaryStage) {
        Parent root = new Pane();
        Scene scene = new Scene(root);

        primaryStage.setTitle("Ejemplo_de_Ventana");
        primaryStage.setScene(scene);
        primaryStage.setWidth(400);
        primaryStage.setHeight(300);
        primaryStage.show();
    }

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