



U1

CONFECCIÓN DE INTERFACES DE USUARIO



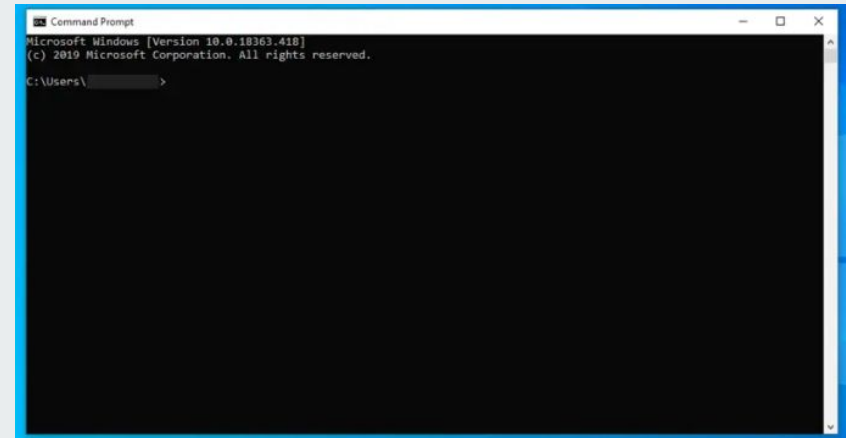
La **interfaz de usuario** es el componente de una aplicación informática que permite al usuario comunicarse con ella.

En la actualidad, el desarrollo de aplicaciones no puede concebirse sin dedicar una parte significativa del tiempo a la **planificación, el análisis, el diseño, la implementación y las pruebas de las interfaces de usuario, ya que constituyen el medio fundamental a través del cual el usuario interactúa con la aplicación** y ejecuta las operaciones para las que fue creada.

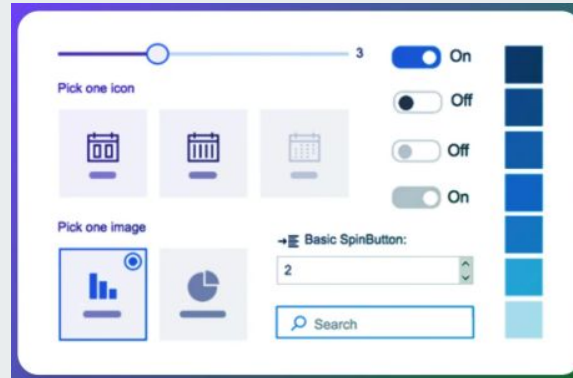
- **Textuais CLI.**
- **Gráficas GUI.**
- **Táctiles TUI.**
- **Interfaces de voz (VUI - Voice User Interface).**
- **Interfaces basadas en xestos (Gesture-Based Interface).**

CLI TEXTUAL: La comunicación se produce por medio de la inserción de ordenas escritas en un interprete de ordenes - terminal.

- **MS-DOS** (sistema operativo por comandos).
- **Terminal de Linux/Unix** (`bash`, `zsh`, `powershell`).
- **Git** (manejo de control de versiones desde consola).
- **MySQL CLI** (gestión de bases de datos mediante comandos).



Gráficas GUI. La interfaz consta de un conjunto de elementos visuales como iconos o menús con los que se interactúa, normalmente, mediante un elemento apuntador (el ratón, por ejemplo). Tienen el mérito de haber popularizado el mundo de la informática para los usuarios noveles.



Táctiles TUI. La comunicación se produce a través de un dispositivo táctil, generalmente una pantalla que puede reaccionar ante la presión táctil de elementos apuntadores o de los dedos. Es habitual combinarlo con respuestas hápticas. Tiene usos en múltiples dispositivos, desde móviles hasta terminales de punto de venta.



Interfaces de voz (VUI - Voice User Interface). Estas interfaces permiten la interacción con el sistema mediante comandos de voz. Su popularidad está en auge gracias a asistentes virtuales como Siri, Alexa y Google Assistant. Son útiles cuando las manos están ocupadas o el dispositivo no tiene pantalla, como en los altavoces inteligentes.



Interfaces basadas en xestos (Gesture-Based Interface). La interacción se realiza mediante movimientos del cuerpo o de las manos, capturados por cámaras o sensores de movimiento, como los que encontramos en consolas de videojuegos como la Xbox Kinect o en sistemas de realidad aumentada. Ejemplo: Leap Motion, Microsoft HoloLens.



Componentes

Toda interfaz por compleja que sea, se compone de una serie de elementos gráficos atómicos que tienen sus propias características y funciones y que se combinan para formar la interfaz. A estos elementos se les llama **componentes** o **controles**. Algunos de los componentes más típicos son:

- **Etiquetas:** Permiten situar un texto en la interfaz. No son interactivos.
- **Campos de texto:** Cuadros de una sola línea en los que podemos escribir algún dato.
- **Áreas de texto:** Cuadros de varias líneas permite escribir párrafos.
- **Botones:** Áreas que se pueden pulsar para llevar a cabo alguna acción.
- **Botones de radio:** Botones circulares que se presentan agrupados para realizar una selección de un único elemento entre ellos. Se usan para preguntar un dato dentro de un conjunto. El botón marcado se representa mediante un círculo.
- **Casillas de verificación:** botones en forma de rectángulo. Se usan para marcar una opción. Cuando está marcada aparece con una marca de verificación dentro de ella. Se pueden seleccionar varias en un conjunto.
- **Imágenes:** se usan para añadir información gráfica a la interfaz.
- **Password:** es un cuadro de texto en el que los caracteres aparecen ocultos. Se utiliza para escribir contraseñas que no deben ser vistas por otros usuarios.
- **Listas:** conjunto de datos que se presentan en un cuadro entre los que es posible elegir uno o varios.
- **Listas desplegables:** combinación de cuadro de texto y lista, permiten escribir un dato o seleccionarlo de la lista que aparece oculta y puede desplegarse.

Evento

Fecha Hora

7/06/15 18:13

Tipo



Banquete



Jornada



Congreso

Asistentes previstos

10

Tipo de Cocina

Buffé

Buffé vegetariano

Carta

Máximo asistentes: Banquete 100, Jornada y Congreso 50 asistentes.

Congreso

Jornadas del congreso

1



Se requiere habitación

Número de habitaciones

1

Tipo de habitación

Individual

Biblioteca de componentes



Bibliotecas de componentes.

Los componentes de una interfaz gráfica suelen agruparse en **bibliotecas**, que permiten usar clases ya definidas o crear nuevas. La elección de la biblioteca depende principalmente del **lenguaje de programación** y del **entorno de desarrollo** utilizado:

- **JAVA Foundation Classes (JFC):** Incluyen las bibliotecas necesarias para crear interfaces gráficas en aplicaciones y applets de Java.
- **AWT:** Primera biblioteca de Java para interfaces gráficas. Es común a todas las plataformas, pero cada una tiene sus propios componentes escritos en código nativo. Actualmente está prácticamente en desuso.
- **Swing:** Biblioteca posterior, totalmente multiplataforma porque no usa código nativo. Muchos de sus componentes derivan de AWT (por ejemplo, *Button* en AWT corresponde a *JButton* en Swing). Incluye gran cantidad de nuevos componentes y es el estándar actual para el desarrollo de interfaces gráficas en Java.
- **JavaFX:** Es la biblioteca más moderna para crear **interfaces gráficas ricas, dinámicas y modernas** en Java. Sustituye progresivamente a **Swing** como la tecnología recomendada. Permite trabajar con **gráficos 2D y 3D, animaciones, efectos, controles multimedia, hojas de estilo (CSS) y un lenguaje declarativo propio (FXML)**. Orientado a aplicaciones de escritorio modernas.

Herramientas para diseñar la interfaz. ¿Wireframe?

Para Swing

- En Eclipse puedes añadir el **WindowBuilder** (plugin oficial de Google).
 - Permite:
 - i. Arrastrar y soltar botones, menús, campos de texto, etc.
 - ii. Genera el código Java de forma automática.

♦ Para JavaFX

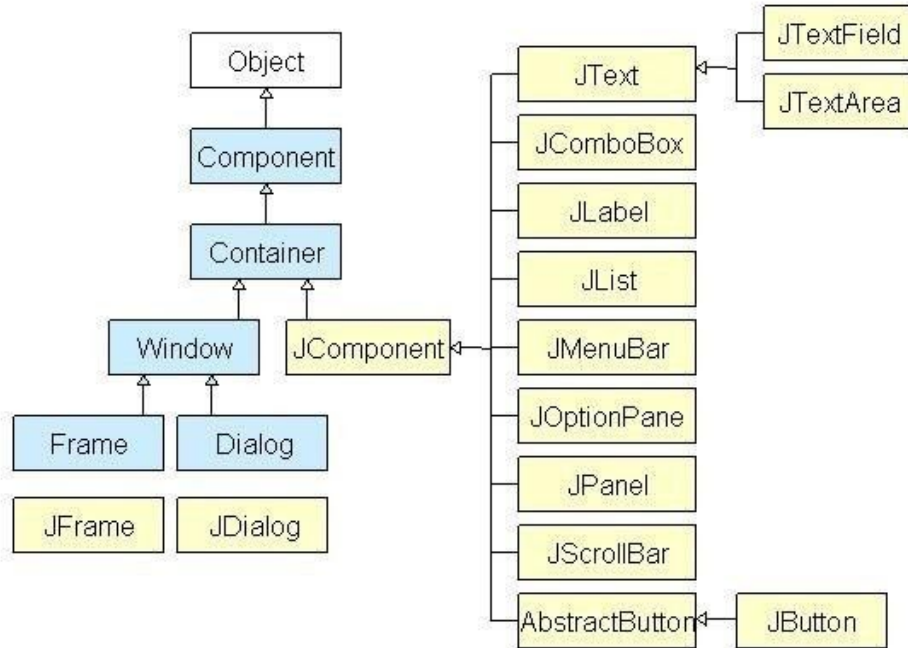
- Lo ideal es usar **JavaFX Scene Builder** (herramienta externa).
 - Diseñas la interfaz de forma visual (drag & drop).
 - Guardas el diseño en un archivo **FXML**.
 - Ese archivo se importa en tu proyecto Eclipse y lo vinculas al código controlador.

👉 Flujo típico en Eclipse con JavaFX:

1. Diseñas en **Scene Builder**.
2. Guardas el `.fxml`.
3. En Eclipse lo llamas desde tu aplicación JavaFX.

De esta forma tienes **wireframe + prototipo funcional** sin programar todos los layouts a mano.

SWING JERARQUIA DE COMPONENTES



Colectores (I)



¿Qué es un formulario?

- Ventana con:
 - Botones para minimizar, maximizar y cerrar
 - Barra de título
 - Bordes delimitadores
- Base para crear aplicaciones de escritorio
- Contiene controles y componentes:
 - Botones, cajas de texto (simples)
 - Barras de menú, rejillas de datos (complejos)

Contenedores:

- Un formulario usa **contenedores** para organizar componentes
- Contenedores permiten incluir otros componentes y más contenedores
- Forman una **estructura jerárquica**

Colectores (II)



Contenedor de nivel superior

- Contenedor especial que incluye un **panel de contenido (content pane)**
- Permite añadir componentes y distribuirlos fácilmente
- Tipos principales:
 - **JFrame** (ventana)
 - **JDialog** (diálogo)
 - **JApplet** (applet)

Tipos de contenedores de nivel superior

- **JFrame**
 - Ventana con título, botones y bordes
 - Ícono por defecto: taza de café (modificable)
 - Puede contener barra de menú
- **JDialog**
 - Ventanas para solicitar información al usuario
 - Pueden ser modales (bloquean otras ventanas) o no modales
- **JApplet**
 - Ejecuta aplicaciones Java en una página web

Uso común en aplicaciones de escritorio

- Ventana principal: **JFrame**
- Ventanas secundarias: otras ventanas o diálogos (**JDialog**)

Colectores (III)



1.3.1 Colectores secundarios

También se interpretan como **diálogos** los siguientes componentes:

- **Panel de opciones (JOptionPane):**
Genera ventanas con botones para responder preguntas con opciones tipo: sí-no, aceptar-cancelar, aceptar, etc.
- **Selector de archivos (JFileChooser):**
Permite seleccionar un archivo del sistema de archivos del equipo donde se ejecuta la aplicación.
- **Selector de colores (JColorChooser):**
Permite seleccionar entre un conjunto de colores y devuelve la selección usando el código adecuado.

Otros tipos de colectores para distribuir controles en la ventana principal

Entre los más habituales están:

- **Paneles (JPanel):** Contenedor intermedio cuya función principal es colocar controles.
- **Barra de menús (JMenu):** Permite la creación de menús complejos de opciones.
- **Barra de herramientas (JToolBar):** Utilizada para contener iconos de acceso rápido a opciones de la aplicación.
- **Pestañas (JTabbedPane):** Tipo particular de panel que permite distribuir elementos en pestañas o fichas.
- **Paneles deslizables (JScrollPane):** Panel especial que permite desplazar automáticamente su contenido.
- **Ventanas internas (JInternalFrame):** Ventanas “hijas” que no pueden salir de los límites de la ventana “padre” donde se crearon. Usadas en aplicaciones con varios documentos abiertos simultáneamente.
- **Paneles divididos (JSplitPane):** Permite visualizar dos componentes, uno a cada lado, asignando espacio dinámicamente a cada uno.

Componentes de la interfaz



¿Qué son los componentes?

- Elementos gráficos que se anidan en los contenedores para formar aplicaciones.
- Se usan para:
 - Mostrar información: etiquetas, imágenes, listas (componentes pasivos).
 - Solicitar información al usuario: cuadros de texto, botones, listas de selección (componentes activos).
- Cada componente se identifica por su **clase** (definición visual y funcional) y su **nombre** (identificador único en la aplicación).
- Poseen **propiedades** que se pueden modificar: texto mostrado, color, tamaño, etc.

Disposición de componentes: Layout

- Define la posición y orden en el formulario.
- Tipos comunes:
 - Bordes (Norte, Sur, Este, Oeste)
 - Rejilla (GridLayout)
 - Fluido (FlowLayout)
- Interacción con el usuario gestionada por **eventos** y **listeners**

Componentes de la interfaz



Añadir y eliminar componentes en WindowBuilder (Eclipse)

- WindowBuilder incluye una **paleta de componentes** a la derecha.
- Categorías:
 - Contenedores: JPanel, JFrame, JDialog, etc.
 - Controles: JButton, JTextField, JLabel, etc.
 - Menús: JMenuBar, JMenu, JMenuItem.
- Para añadir un componente:
 - Seleccionar en la paleta y hacer clic en el diseño visual.
 - Se puede mover y redimensionar con el ratón.
- Guías visuales ayudan a alinear componentes.
- Para eliminar:
 - Seleccionar componente y pulsar Supr o botón derecho > Delete.

Componentes de la interfaz



Modificar propiedades en WindowBuilder

- Selecciona el componente en el editor visual.
- Modifica sus propiedades en el panel **Properties**.
- Propiedades importantes:
 - **variable name**: para referenciarlo en el código.
 - **text**: texto visible en botones, etiquetas, etc.
 - **tooltipText**: ayuda que aparece al pasar el ratón.

Añadir funcionalidad en Eclipse

- WindowBuilder genera código básico automáticamente.
- Para añadir lógica:
 - Ve al archivo Java correspondiente.
 - Añade código en los métodos de eventos (listeners).
 - Puedes añadir listeners directamente desde el editor visual (clic derecho > Add Event Handler).

Componentes de la interfaz



Layouts disponibles y uso en WindowBuilder

Layout	Descripción
BorderLayout	Componentes en Norte, Sur, Este, Oeste
GridLayout	Rejilla con filas y columnas
GridBagLayout	Rejilla flexible, componentes pueden ocupar varias celdas
FlowLayout	Componentes uno tras otro, saltando fila si no hay espacio
GroupLayout	Layout avanzado, usado por defecto en WindowBuilder

WindowBuilder permite seleccionar y cambiar el layout fácilmente desde el editor visual. Puedes personalizar el posicionamiento sin escribir código manualmente.

Componentes de la interfaz



Asociación de acciones a eventos en Eclipse

- En GUI, la ejecución depende de la interacción del usuario.
- Define:
 - Qué acciones debe realizar la aplicación.
 - Cuándo realizarlas (qué eventos las disparan).
- En Java:
 - Objetos que generan eventos (**event sources**) registran listeners.
 - Listeners implementan métodos que se ejecutan cuando ocurre un evento.
- En WindowBuilder:
 - Añade listeners fácilmente desde el editor visual.
 - Implementa la lógica dentro de los métodos generados.