



Argentina programa 4.0

Clase 4

Contenido

Introducción.....	2
¿Qué es una interfaz de usuario o UI?.....	3
Importancia de la experiencia del usuario.....	5
Características de la interfaz de usuario.....	6
Swing GUI Designer.....	8
Estructura jerárquica de componentes.....	11



Introducción

Ya sea que te dediques al desarrollo web o quieras encargar el diseño de una aplicación, una página web u otro producto digital para tu negocio, existen conceptos y software para diseño de interfaces que es conveniente que conozcas.

Todo diseñador front-end debe contar con algunos recursos imprescindibles para realizar su trabajo, el primer paso es realizar una propuesta gráfica o prototipo de interfaz de usuario para que el cliente pueda imaginar cómo se verá terminado.

En el desarrollo de aplicaciones de escritorio en Java, es común utilizar la biblioteca Swing, que proporciona componentes gráficos para crear interfaces de usuario. Una forma conveniente de diseñar y crear ventanas en Java Swing es mediante el uso del Swing GUI Designer.

Existen varias herramientas disponibles para diseñar interfaces Swing, y algunos entornos de desarrollo integrados (IDE) también tienen sus propios editores visuales. Algunas opciones populares para el diseño de interfaces Swing incluyen: 1. NetBeans: Es un IDE de código abierto para Java que incluye un editor visual llamado "Matisse" para diseñar interfaces Swing. 2. IDEA de IntelliJ: Ofrece un editor visual de GUI que permite arrastrar y soltar componentes Swing en un formulario. Puede seleccionar componentes como botones, etiquetas, campos de texto y más desde una paleta y colocarlos en el formulario de manera intuitiva.

¿Qué es una interfaz de usuario o UI?

Pero primero lo primero, para quienes no saben nada del tema comencemos por la definición de interfaz de usuario.



Imagen1 -Interfaz de Usuario

Una interfaz de usuario o UI (User Interface) es el medio a través del cual el usuario se comunica con un dispositivo tecnológico, por lo que abarca todos los puntos de contacto entre la persona y el equipo o programa.

Por ejemplo, es el medio que nos permite interactuar con nuestra computadora mediante el uso del mouse y del teclado (interfaz de hardware) o cuando realizas una transferencia bancaria a través de tu celular (interfaz de software).

Para el desarrollo de todo tipo de programas es necesario que un diseñador de interfaces de usuario planifique la información que le va a mostrar al usuario y determine el mapa de navegación entre los diferentes comandos.

Entonces, ¿cómo funciona una interfaz? El usuario transmite lo que quiere hacer mediante el conjunto de controles y canales sensoriales disponibles y la interfaz responde.

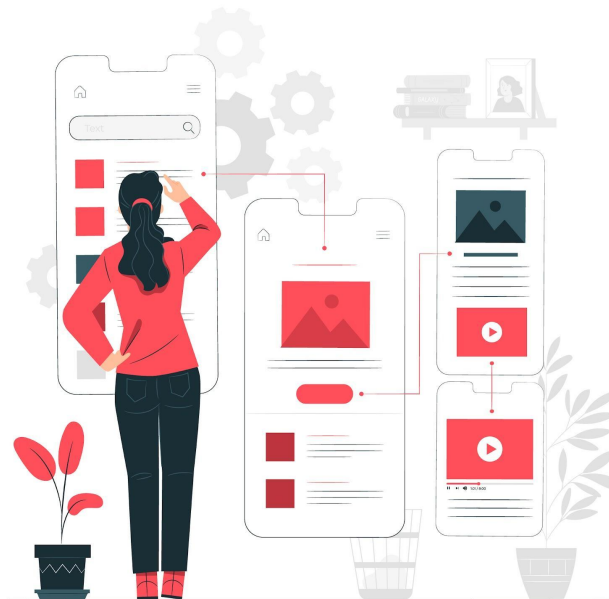


Imagen2 - Funcionamiento de la IU

Como verás, las interfaces, aunque no seamos conscientes de ello, forman parte de la vida diaria de prácticamente todas las personas. Por lo que es necesario optimizar su

funcionalidad desde que se genera el primer prototipo de la interfaz de usuario para que el diseño sea lo más intuitivo y accesible posible.

Existen diferentes software para diseño de interfaces enfocados en la experiencia del usuario que podrán serte de utilidad a la hora de encarar un proyecto.

Importancia de la experiencia del usuario

Ya lo dijo Steve Jobs: El diseño no es solo lo que se ve y se siente. El diseño es cómo funciona.

Lo primero que debes detectar antes de comenzar con el primer prototipo de la interfaz de usuario es el problema o necesidad que le vas a resolver a ese usuario, no tener en cuenta la experiencia del usuario a la hora de crear tu producto digital con un software para diseño de interfaces puede costarte clientes.

Es posible que una persona que ingrese a tu web o a tu aplicación y no encuentre rápidamente lo que busca o no comprenda cómo navegar no vuelva a ingresar porque su experiencia no fue gratificante.

La interacción con una buena interfaz resulta casi de forma natural. Para ello, el diseño debe ser agradable, intuitivo y eficiente.

A continuación, conoce todas las características que no pueden faltar en el diseño de una buena interfaz de usuario.

Características de la interfaz de usuario

Para cumplir con su objetivo, ayudar al usuario a encontrar lo que busca, el diseño de interfaces de usuario (UI) debe tener en cuenta:



Imagen3 - Características de la IU

1. Claridad

Es de suma importancia en el proceso de diseño de interfaz de usuario ya que transmitir la información de manera precisa evita que el usuario cometa errores y esto se traduce en una buena experiencia de uso.

2. Concisión

Bríndale al usuario solo la información que necesita y pide, no agregues información que genere confusión y lo distraiga de lo que es más importante. Diseño y contenido tienen que acompañar y ser relevantes en cada punto.

3. Coherencia

Esta característica es fundamental ya que es la que logra que tu interfaz sea intuitiva y le permite al usuario desarrollar patrones de uso a partir de conocer la función de los botones, iconos y otros elementos de la interfaz.

4. Legibilidad

Un lenguaje simple y fácil de procesar le permite al usuario realizar una lectura rápida y acostumbrarse a las diversas funciones de la interfaz.

5. Flexibilidad

Otra característica que forma parte de una buena experiencia del usuario es permitirle deshacer acciones erróneas, restaurar elementos que haya borrado y dar opciones y alternativas para llevar a cabo un proceso. Elegir un software para diseño de interfaces que te facilite estas acciones te ayudará a que la interacción sea más sólida.

6. Atractiva visualmente

La funcionalidad es primordial, ya lo dijo Steve, pero no por eso hay que subestimar el aspecto visual durante el proceso de diseño de interfaz de usuario. Cada vez más personas están pendientes del diseño y prefieren interfaces agradables a la vista.

El software de diseño de interfaces que elijas para trabajar determinará las opciones visuales que podrás utilizar.

Swing GUI Designer

Swing GUI Designer es una herramienta visual que permite diseñar y crear interfaces gráficas de usuario de manera intuitiva, utilizando la biblioteca Swing en Java



Imagen4 - Swing GUI

Java Swing es una biblioteca gráfica para Java. Es un kit de herramientas de widgets de interfaz gráfica de usuario (GUI) de Java que incluye un amplio conjunto de widgets.



Swing incluye controles integrados como árboles, botones de imagen, paneles con pestañas, controles deslizantes, barras de herramientas, selectores de color, tablas y áreas de texto para mostrar HTTP o formato de texto enriquecido (RTF). Los componentes Swing están escritos completamente en Java y, por lo tanto, son independientes de la plataforma.

Swing ofrece personalización de la apariencia de cada componente de una aplicación sin realizar cambios significativos en el código de la aplicación. También incluye una característica de aspecto, que le permite emular la apariencia de los componentes nativos sin dejar de tener la ventaja de la independencia de la plataforma. Esta característica particular hace que escribir aplicaciones en Swing sea fácil y lo distingue de otros programas nativos.

Swing se distribuyó como una biblioteca descargable y se ha incluido como parte de Java Standard Edition 1.2. Originalmente, la biblioteca de gráficos para Java, desarrollada por Netscape Communication Corporation, se llamaba Internet Foundation Classes (IFC). El primer lanzamiento de IFC fue el 16 de diciembre de 1996. La evolución de JFC se remonta a 1997, cuando a Sun Microsystems y Netscape Communication Corporation se le ocurrió la idea de fusionar IFC con otras tecnologías.



Imagen5 - Swing Logo

Swing es una herramienta muy utilizada por los desarrolladores de software para la creación de interfaces de usuario. Es una de las mejores opciones para desarrollar aplicaciones de escritorio en Java y ofrece una amplia variedad de widgets personalizables y fáciles de usar. Con Swing, es posible crear aplicaciones atractivas y con una gran experiencia de usuario.

Además, gracias a su independencia de la plataforma, los componentes Swing son fáciles de utilizar en cualquier sistema operativo y no requieren de muchas modificaciones en el código de la aplicación.

En definitiva, Java Swing es una herramienta esencial para cualquier desarrollador de software que busque una solución rápida y eficiente para la creación de interfaces gráficas de usuario. Gracias a su gran variedad de widgets personalizables y su facilidad de uso, es la elección perfecta para la creación de aplicaciones de escritorio de alta calidad en Java.

Estructura jerárquica de componentes

Los componentes básicos proporcionados por **Java Swing** son:

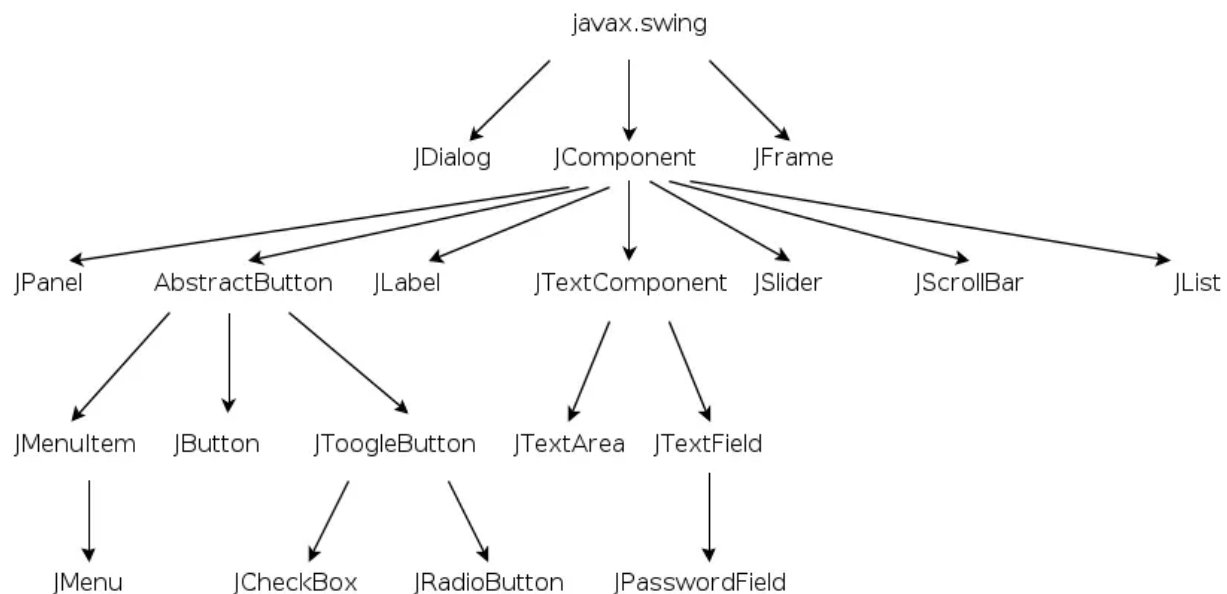


Imagen6 - Estructura jerárquica de componentes Swing

Este árbol trabaja con tres tipos de elementos en una aplicación:

1. **Contenedor de nivel superior:** Es un elemento que generalmente se usa como base, es decir, proporcionar un lugar para usar otros elementos. Ejemplos de este tipo serían: `JFrame`, `JDialog` y `JComponent`.
2. **Componente de nivel intermedio:** Es un elemento que se usa solo para manejar la ubicación de los elementos de botones y etiquetas. Ejemplos de este tipo serían: `JPanel`, `JScrollBar`, etc.
3. **Contenedor Atómico:** Es un elemento que no se usa para almacenar otros elementos, es una entidad autosuficiente, que sirve para presentar

información al usuario o para recibir información proporcionada por el usuario. Algunos de los ejemplos de este tipo son: `JButton`, `JLabel`, etc.

Los componentes de Java Swing en Java son una parte importante del desarrollo de interfaces de usuario. Permiten que las aplicaciones tengan elementos visuales para que el usuario interactúe y son una parte crucial del marco Swing. Describiremos algunos de ellos:

JFrame y JDialog:

`JFrame` es una clase utilizada en Swing para generar ventanas sobre las cuales añadir distintos objetos con los que podrá interactuar o no el usuario. A diferencia de `JPanel`, `JFrame` posee algunas nociones típicas de una ventana como minimizar, cerrar, maximizar y poder moverla.

`JDialog`: Diálogos modales. Cuando nuestra aplicación necesita más de una ventana, debemos utilizar ventanas secundarias mediante el componente `JDialog`.

Un `JFrame` no admite a ninguna otra ventana como padre. El `JDialog` sí admite como padres un `JFrame` u otros `JDialog`. Esto es importante porque una ventana hija siempre quedará por encima de su ventana padre. Si al crear los `JDialog` les pasamos el `JFrame` como padre o parent, los `JDialog` siempre estarán visibles por encima del `JFrame`. No se podrán ir "detrás" de él.

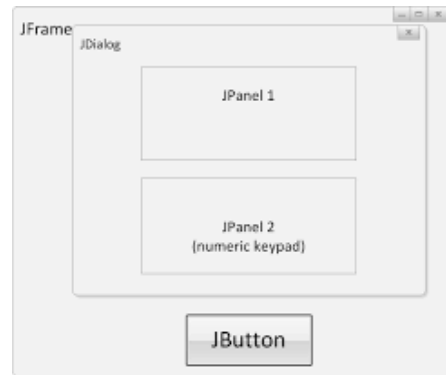


Imagen7 -JFrame y JDialog

JLabel

Estos componentes pueden ser usados para mostrar textos simples, imágenes con textos. También pueden ser usados para mostrar los resultados de un proceso. Los métodos más usados son:

- `setText()` - Permite colocar un valor de texto, puede ser usado para mostrar un resultado en este componente.
- `setFont()` - Permite definir el tipo y el estilo de la fuente (negrita y/o cursiva) y el tamaño de la fuente.
- `setIcon()` - Permite que un ícono o una imagen sea usada y mostrada en la pantalla.

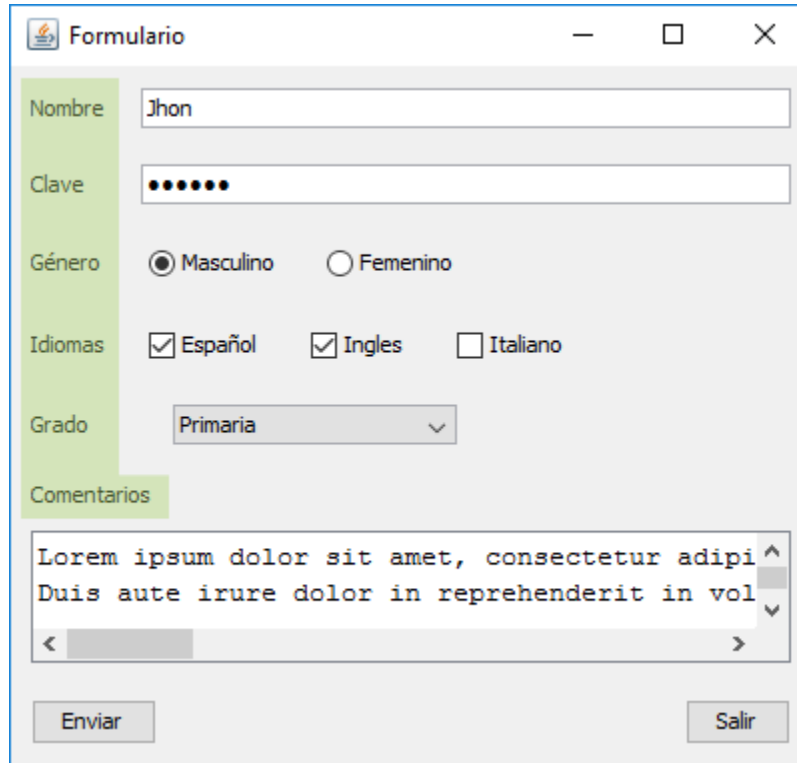


Imagen8 -JLabel

JTextField, JPasswordField y JTextArea

Estos componentes son usados para que el usuario introduzca un valor textural y capturarlo. JTextField permite introducir una línea de texto, JPasswordField permite introducir un valor pero lo muestra camuflado para no poder ver el valor introducido y JTextArea permite introducir varias líneas de texto. Los métodos más usados son:

- `setText()` - Permite colocar un valor de texto, puede ser usado para mostrar un resultado en este componente.
- `getText()` - Permite recuperar el valor de una caja de texto.
- `setFont()` - Permite definir el tipo y el estilo de la fuente (negrita y/o cursiva) y el tamaño de la fuente.

- `setEnabled()` - Permite que la escritura sea deshabilitada y puede ser usado para mostrar un resultado como en un JLabel.

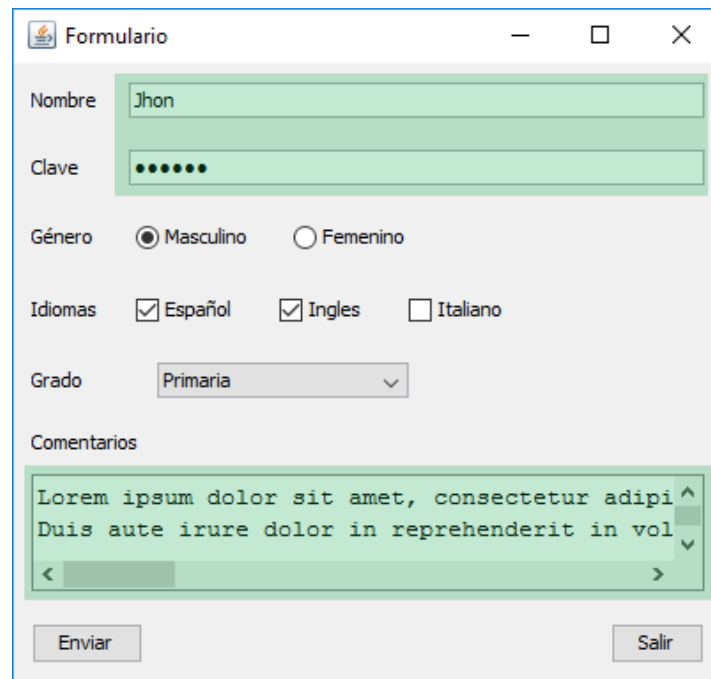


Imagen9 - JTextField

Botones (JRadioButton, JCheckBox y JButton)

Estos componentes permiten varias formas gráficas con las cuales el usuario puede interactuar, el JRadioButton permite elegir una opción dentro de un grupo de elecciones, en cuanto al JCheckBox permite seleccionar más de una de estas opciones. Finalmente los JButton son usados para ejecutar un evento cuando son presionados.

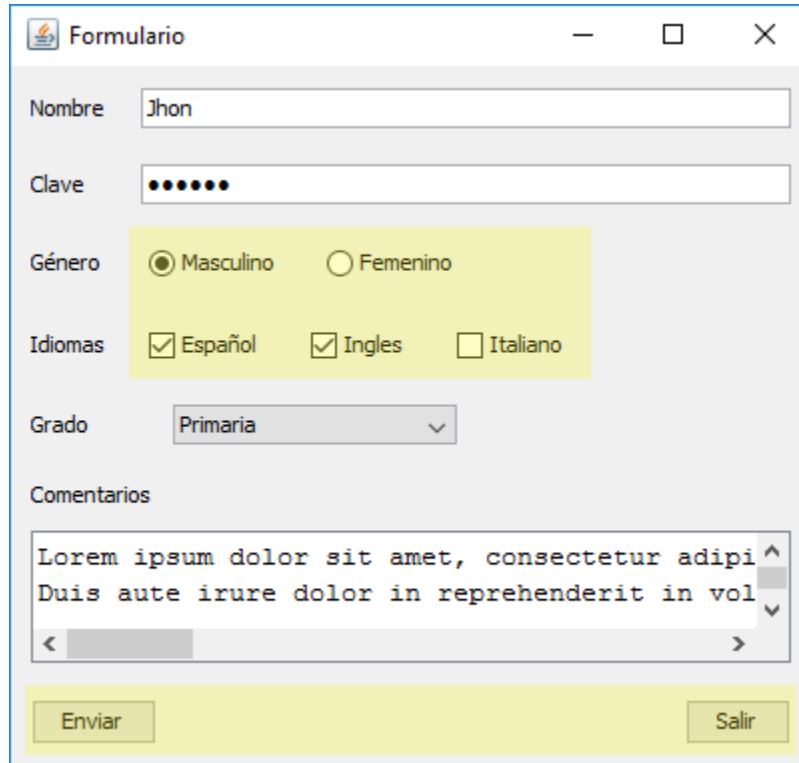


Imagen10 - Botones

Listas desplegables (JcomboBox)

Parecido al JRadioButton, permite seleccionar una opción de un grupo de alternativas pero sin ocupar espacio en la pantalla.

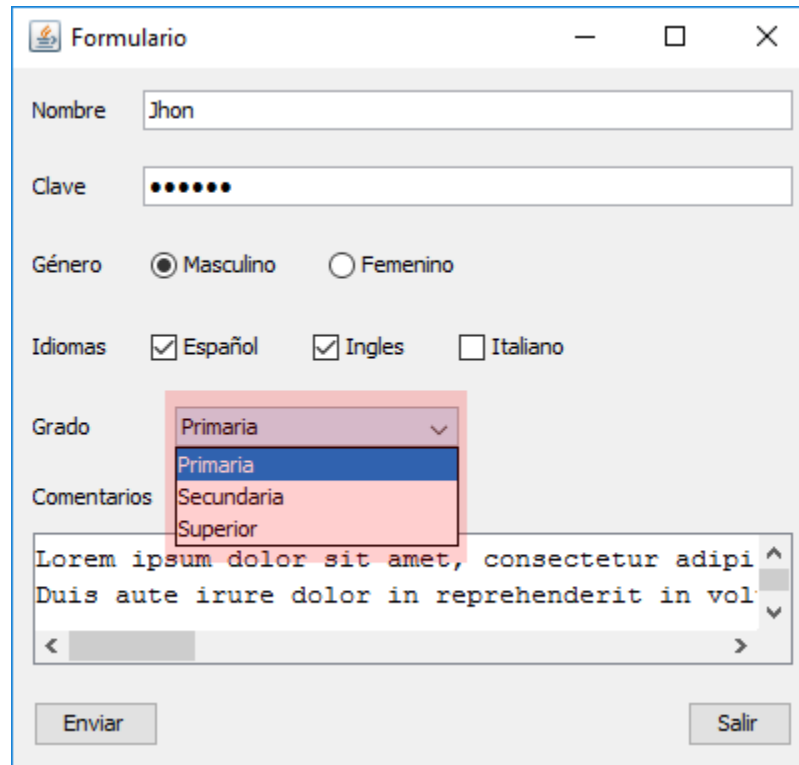


Imagen11 - Listas Desplegables

Ahora que ya sabes la existencia de los componentes Swing es super recomendable buscar y explorar más información sobre ellos. Son innumerables las posibilidades con Java Swing. Sus componentes pueden ayudarte a construir aplicaciones geniales.

Video explicativo : GUI - Parte 1 <https://youtu.be/Fc4uFeMXBS8>

GUI - Parte2 <https://youtu.be/DAU0rDqcArs>