

Introducción

Los componentes swing utilizan la infraestructura de AWT, incluyendo el modelo de eventos AWT, el cual gestiona eventos como la pulsación del teclado, ratón, etc. Los programas swing deben importar los paquetes: `javax.swing.*` y AWT: `java.awt.*` y `java.awt.event.*`.

La estructura básica de una interfaz gráfica utilizando swing tiene los siguientes elementos:

- Contenedores de alto nivel: representan las ventanas de la interfaz gráfica de usuario (clase `JFrame` y cuadros de diálogo `JDialog`). Estos tienen contenedores intermedios.
- Contenedores intermedios: agrupan componentes de acuerdo con algún criterio lógico (clase `JPanel` y clase `JScrollPane`). A su vez, los contenedores intermedios contienen componentes.
- Componentes: elementos de bajo nivel que representan diferentes componentes gráficos, estos interactúan con los usuarios a través de eventos.

Los pasos básicos para la construcción de una interfaz gráfica de usuario son:

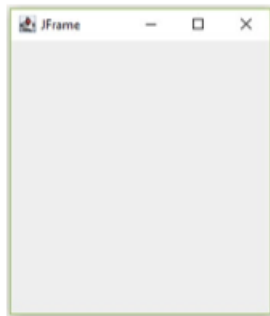
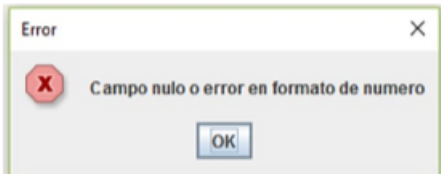
1. Crear una nueva clase para una ventana (o directamente instanciar un objeto `JFrame`).
2. Crear los componentes de la interfaz gráfica de usuario.
3. Crear uno o varios contenedores intermedios.
4. Asociar los componentes al contenedor.
5. Asociar el contenedor a la ventana.
6. Hacer visible la ventana.

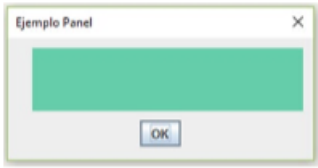

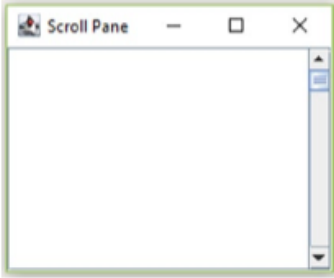
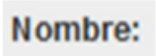
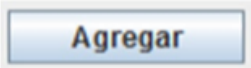

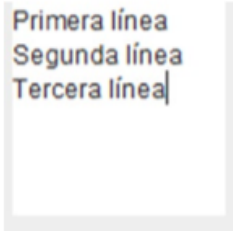
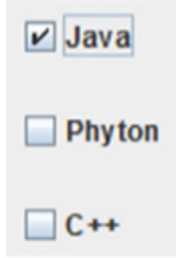
Un componente es el objeto fundamental de la interfaz de usuario en Java.


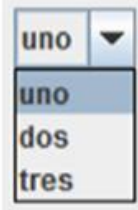

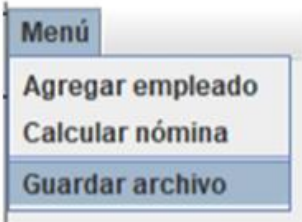
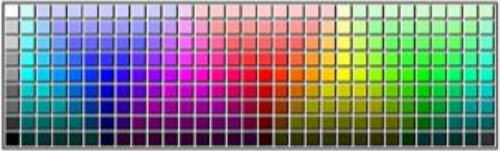


Todo lo que ve en la interfaz en una aplicación Java es un componente. Para que este pueda ser utilizado debe colocarse en un contenedor. Todos los componentes swing se derivan de la clase abstracta `JComponent`.

A continuación, se muestra un listado abreviado de diferentes contenedores y componentes swing.

Lista de contenedores

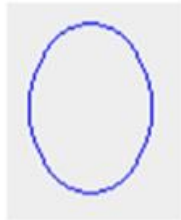
Contenedores de alto nivel	Descripción	Figura
<i>JFrame</i>	Clase para generar ventanas en las cuales se agregarán distintos objetos con los que podrá interactuar el usuario.	
<i>JDialog</i>	Clase para crear diálogos personalizados.	

Contenedores intermedios	Descripción	Figura
<i>JPanel</i>	Clase que representa un contenedor genérico.	
<i>Container</i>	Contenedor genérico AWT.	
<i>JScrollPane</i>	Gestiona una ventana gráfica con barras de desplazamiento vertical y horizontal opcionales.	
Componentes	Descripción	Figura
<i>JLabel</i>	Un componente utilizado para mostrar una cadena corta de caracteres o un icono de imagen.	
<i>JButton</i>	Un componente para crear un botón etiquetado. La aplicación realiza algún tipo de acción cuando se pulsa el botón.	
<i>TextField</i>	Un componente que implementa un texto editable de una sola línea.	
<i>TextArea</i>	Un componente que implementa un texto editable multilínea.	
<i>JCheckBox</i>	Un componente que implementa una casilla de verificación: un elemento que se puede seleccionar o deseleccionar y que muestra su estado al usuario.	

JRadioButton	Un componente que implementa un botón de radio: un elemento que se puede seleccionar o deseleccionar y que muestra su estado al usuario.	
JComboBox	Un componente que combina un campo editable y una lista desplegable. Se puede seleccionar un valor de la lista.	
JList	Un componente que muestra una lista de objetos y permite al usuario seleccionar uno o más elementos.	
JMenu	Implementación de un menú: una ventana emergente que contiene JMenuItems, estos se muestran cuando se selecciona un elemento en JMenuBar.	
JColorChooser	Proporciona un panel de control que permite a un usuario manipular y seleccionar un color.	
JOptionPane	Facilita la aparición de un cuadro de diálogo estándar que solicita a los usuarios un valor o informa algo.	
JToolBar	Proporciona un componente que es útil para mostrar controles de uso común.	
JSlider	Permite seleccionar un valor deslizando un indicador dentro de un rango.	
JSpinner	Un campo de entrada que permite seleccionar un número o un valor de una secuencia ordenada.	

<i>JPasswordField</i>	Un componente que permite editar una sola línea de texto se muestra que hay algo escrito, pero no muestra los caracteres originales.	Contraseña: <input type="password" value="*****"/>						
<i>JTable</i>	Un componente utilizado para mostrar y editar tablas de celdas bidimensionales regulares.	<table border="1"> <tr> <td>Pepito</td><td>Pérez</td><td>3300000.0</td></tr> <tr> <td>Sutanito</td><td>Martínez</td><td>2.43E7</td></tr> </table>	Pepito	Pérez	3300000.0	Sutanito	Martínez	2.43E7
Pepito	Pérez	3300000.0						
Sutanito	Martínez	2.43E7						

Cada componente posee una colección de atributos y métodos bastante extensa. Por cuestiones de espacio no se incluyen en este texto. Se recomienda referirse a la documentación API de Java API Java.

Clases de soporte	Descripción	Figura
<i>Graphics</i>	Clase abstracta para los contextos gráficos que permiten que se dibuje imágenes en componentes fuera de la pantalla.	
<i>Color</i>	Clase para encapsular colores RGB o colores arbitrarios identificados por un <i>ColorSpace</i> .	Hola mundo!!!
<i>Font</i>	Clase que representa las fuentes que se utilizan para representar el texto de forma visible.	Hola mundo!!

Objetivos de aprendizaje

Al finalizar este ejercicio, tendrás la capacidad para:

- Conocer y aplicar diversos componentes swing para desarrollar interfaces gráficas de usuario.
- Realizar una gestión básica de los eventos generados mediante la interacción con los diferentes componentes swing.

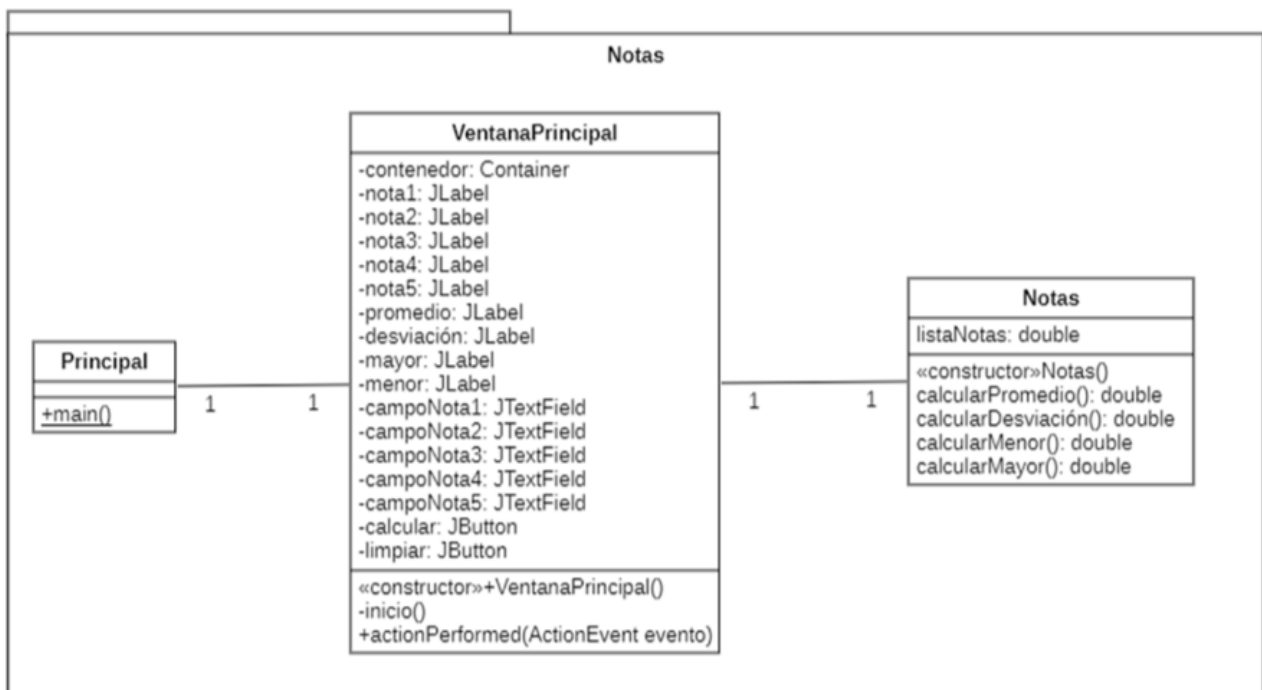
Enunciado: Notas

Se requiere desarrollar un programa con interfaz gráfica de usuario que genere una ventana donde se solicite el ingreso de cinco notas obtenidas por un estudiante.

El programa debe calcular y mostrar en la parte inferior de la ventana los siguientes datos:

- El promedio de notas ingresadas.
- La desviación estándar de las notas ingresadas.
- La mayor nota obtenida.
- La menor nota obtenida.

Diagrama de clases



Explicación del diagrama de clases

Se ha definido un paquete denominado “Notas” que incluye un conjunto de clases. El punto de entrada al programa es la clase Principal que cuenta con el método main. La clase Principal está relacionada mediante una relación de asociación con la clase VentanaPrincipal que posee atributos privados para identificar los diferentes componentes gráficos de la ventana: un contenedor de componentes gráficos (Container), etiquetas (JLabel), campos de texto para ingreso de las notas (JTextField) y botones (JButton) para calcular diferentes métricas y borrar las notas. La clase VentanaPrincipal cuenta con un constructor y unos métodos para generar la ventana gráfica con sus componentes (inicio) y para gestionar los diferentes eventos surgidos al interactuar con esta ventana (actionPerformed).

La clase VentanaPrincipal se conecta por medio de una relación de asociación con la clase Notas. La relación de asociación tiene una multiplicidad de uno en cada extremo, lo cual indica que la VentanaPrincipal se relaciona con una única instancia de Notas y, a su vez, Notas se relaciona con una sola VentanaPrincipal.

La clase Notas tiene como un único atributo un array de notas de tipo double. Además, cuenta con un constructor y métodos para calcular la media, desviación estándar, la mayor nota y la menor nota.

Diagrama de objetos



Ejecución del programa

Notas

Nota 1: 5

Nota 2: 4.5

Nota 3: 4

Nota 4: 3.5

Nota 5: 3

Calcular Limpiar

Promedio = 3,00

Desviación estándar = 1,22

Valor mayor = 5.0

Valor menor = 3.0

Ejercicios propuestos

- Modificar el programa del ejercicio para generar mensajes de alerta cuando se ingresen datos que no sean numéricos.
- Es obligatorio ingresar las cinco notas, se genera un mensaje de alerta cuando una nota no ha sido ingresada.

Instrucciones Java del ejercicio

Clase	Método	Descripción
Math	<i>double pow(double a, double b)</i>	Retorna el valor del primer argumento elevado a la potencia del segundo argumento.
	<i>double sqrt(double a)</i>	Retorna la raíz cuadrada positiva del valor a.
JFrame	<i>JFrame()</i>	Constructor de la clase <i>JFrame</i> .
	<i>void setTitle(String título)</i>	Establece el título de la ventana con el <i>String</i> especificado.
	<i>void setSize(int x, int y)</i>	Cambia el tamaño del componente para que tenga una anchura x y una altura y.
	<i>void setLocationRelativeTo(Component c)</i>	Establece la ubicación de la ventana en relación con el componente especificado.
	<i>void setDefaultCloseOperation(opciones)</i>	Usado para especificar una de las siguientes opciones del botón de cierre: EXIT_ON_CLOSE, HIDE_ON_CLOSE, DISPOSE_ON_CLOSE o DO_NOTHING_ON_CLOSE.
	<i>void setResizable(boolean resizable)</i>	Para evitar que se cambie el tamaño de la ventana.
	<i>void setVisible(boolean b)</i>	Muestra u oculta la ventana según el valor del parámetro b.
ActionListener	<i>void actionPerformed(ActionEvent e)</i>	Se invoca cuando ocurre un evento.
Container	<i>Container getContentPane()</i>	Retorna el objeto <i>ContentPane</i> de la ventana.
	<i>void setLayout(LayoutManager mgr)</i>	Establece el <i>layout</i> de la ventana.
	<i>Component add(Component comp)</i>	Añade el componente especificado al final del contenedor.
Component	<i>void addActionListener(this)</i>	Añade un oyente de eventos al componente actual.
	<i>void setBounds(int x, int y, int ancho, int alto)</i>	Mueve y cambia el tamaño del componente.
JLabel	<i>JLabel()</i>	Constructor de la clase <i>JLabel</i> .
	<i>void setText(String text)</i>	Define una línea de texto que mostrará este componente.
TextField	<i>TextField()</i>	Constructor de la clase <i>TextField</i> .
	<i>String getText()</i>	Retorna el texto contenido en el componente de texto.
JButton	<i>JButton()</i>	Constructor de la clase <i>JButton</i> .
	<i>void setText(String text)</i>	Define una línea de texto que mostrará este componente.
Event	<i>Object getSource()</i>	El objeto sobre el cual el evento inicialmente ha ocurrido.