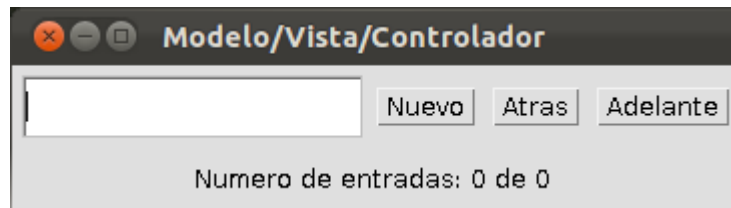


INTERFACES GRÁFICAS CON SWING

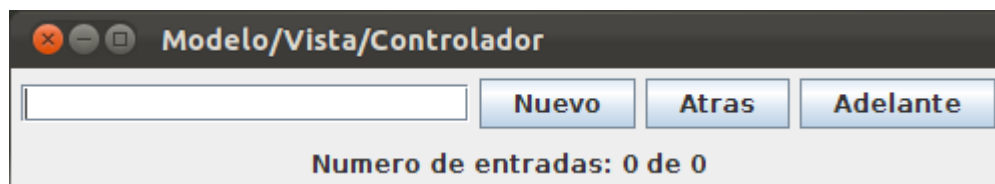
Existen clases (plantillas) propias de Java, para crear o instanciar objetos, con el mismo cometido tanto en los paquetes **AWT** como en **SWING**. Por ejemplo, en ambas bibliotecas tenemos una clase para crear ventanas **Frame** en el caso de AWT y **JFrame** en el caso de Swing. Fíjate que si es una clase del paquete Swing su nombre empieza por **J**.

Existen otras clases que son del paquete AWT pero se utilizan en Swing, por ejemplo, los eventos y escuchadores. Como no tienen representación gráfica, en Swing se reaprovechan los de AWT. Recordar que un **EVENTO** es cualquier acción que el usuario ejecute sobre la interfaz o ventana creada; por ejemplo sobre cualquier objeto de la ventana mover el mouse, hacer click, dobleclick, al presionar tecla, etc.

En el caso de Linux, esta es la versión programada con AWT:



Y programada con Swing:



Las diferencias también son evidentes, puede detectar visualmente cuales diferencias?

Swing es una biblioteca de interfaces gráficas de usuario (GUI) para Java.

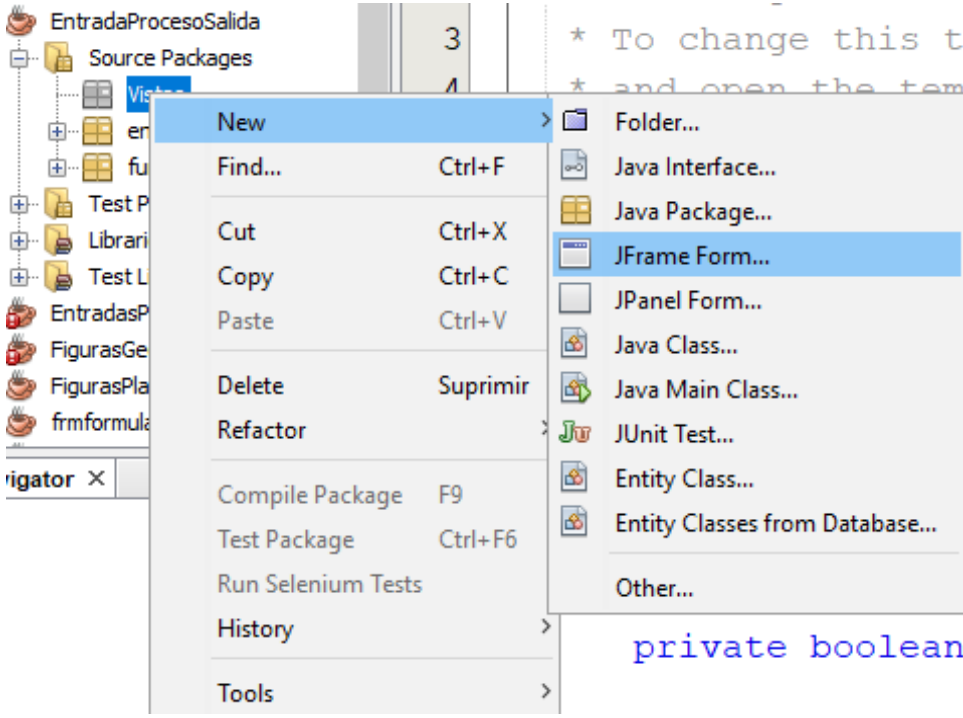
- Viene incluida con el entorno de desarrollo de Java (JDK).
- Extiende a otra librería gráfica más antigua llamada **AWT**.

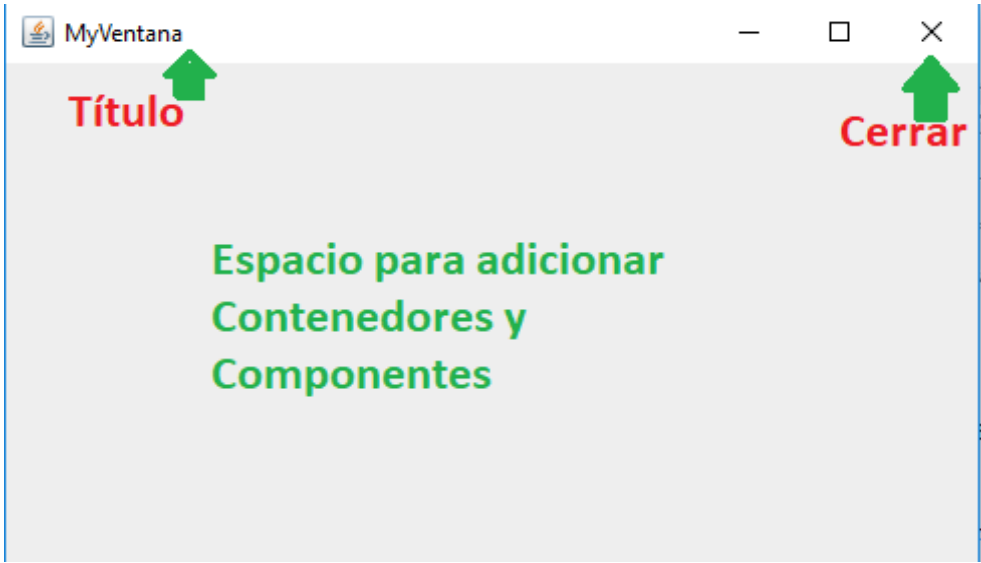
Paquetes a importar

- **javax.swing**
- **java.awt**
- **java.awt.event**

Swing tener presente los siguientes aspectos

- Ventanas
- Componentes
- Layout Managers (manejadores de aspectos)
- Manejo de eventos
- Cuadros de diálogo predefinidos
- Dibujo de gráficos
- Arquitectura **MVC**

1.	CREACIÓN DE VENTANAS (JFrame): La clase JFrame proporciona operaciones para manipular ventanas	
	CONSTRUCTORES	MÉTODOS: Acciones sobre el Objeto
	Sin título JFrame ventana = new JFrame ();	Establecer de la tamaño de la ventana: ventana.setSize(ancho, alto);
	Con título JFrame ventana = new JFrame ("Titulo ventana");	Dejar que los componentes se acomoden al tamaño la ventana ventana.pack();
		Hacer visible la ventana ventana.setVisible(true);
	Establecer la acción de cierre de la ventana. ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);	
1.	Ejemplo simple para una ventana vacía, crear una clase con el nombre de frmMyVentana , compilar y correr el programa (seguir el paso a paso).	
1.1.	 <p>The screenshot shows an IDE interface. On the left is a 'Source Packages' tree with folders like 'EntradaProcesoSalida', 'Test P', 'Librari', 'Test L', 'EntradasP', 'FigurasGe', 'FigurasPla', and 'frmformula'. A context menu is open over the 'frmformula' folder, showing options like 'New', 'Find...', 'Cut', 'Copy', 'Paste', 'Delete', 'Refactor', 'Compile Package', 'Test Package', 'Run Selenium Tests', 'History', and 'Tools'. The 'New' submenu is expanded, showing options like 'Folder...', 'Java Interface...', 'Java Package...', 'JFrame Form...' (which is highlighted), 'JPanel Form...', 'Java Class...', 'Java Main Class...', 'JUnit Test...', 'Entity Class...', 'Entity Classes from Database...', and 'Other...'. In the background, a code editor shows a snippet of Java code: 'private boolean'.</p>	

1.2.	Cada proyecto deberá quedar en una carpeta totalmente nueva, ya que por cada proyecto se generan otros directorios y archivos; el nombre de la clase se recomienda anteponer el prefijo frm frmMyVentana
1.3.	Los IDE por defecto generan una plantilla para la ventana, sobre la cual se trabajará para incluir los nuevos componentes. Compilamos y Ejecutamos la clase, para descartar errores.
1.4.	<p>El resultado es el siguiente, incluye los botones para cerrar, minimizar y maximizar la ventana, para el título con el mouse derecho aplicar propiedades TITLE “My Ventana</p>  <p>Detalles a tener en cuenta:</p> <ul style="list-style-type: none"> • La ventana tiene barra de título y botones. • Le hemos dado unas dimensiones iniciales. • La hemos hecho visible. <p>Si pruebas a cerrar la ventana pulsando en el botón del marco comprobarás que esta se cierra. Ya que en el constructor se aplican los métodos dispose() y System.exit(0); igualmente se obtiene el mismo resultado si a la ventana se aplica el método JFrame.EXIT_ON_CLOSE</p>
1.5.	<p>Si se utiliza un IDE que genera la plantilla del formulario, como mínimo le genera el código respectivo a la instanciación de objetos a partir de los componentes que se van creando en el diseño; el siguiente código estándar para cualquier ventana:</p> <pre>package Vistas; public class frmMyVentana extends javax.swing.JFrame {</pre>

	<pre> public frmMyVentana() { initComponents(); } @SuppressWarnings("unchecked") // <editor-fold defaultstate="collapsed" desc="Generated Code"> private void initComponents() { setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE); setTitle("MyVentana"); javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane()); getContentPane().setLayout(layout); layout.setHorizontalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING) .addGap(0, 400, Short.MAX_VALUE)); layout.setVerticalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING) .addGap(0, 300, Short.MAX_VALUE)); pack(); }// </editor-fold> public static void main(String args[]) { /* Create and display the form */ java.awt.EventQueue.invokeLater(new Runnable() { public void run() { new frmMyVentana().setVisible(true); } }); } // Variables declaration - do not modify // End of variables declaration }//fin clase frmMyVentana </pre>
1.6.	<p>Otras acciones de cierre de la clase JFrame:</p> <ul style="list-style-type: none"> ● JFrame.EXIT_ON_CLOSE : Cierra la ventana y abandona aplicación. ● JFrame.DISPOSE_ON_CLOSE : Libera los recursos asociados a la ventana. ● JFrame.DO_NOTHING_ON_CLOSE : No hace nada. ● JFrame.HIDE_ON_CLOSE : Cierra la ventana, sin liberar sus recursos.

2.	COMPONENTES GUI EN UNA VENTANA (ENTRADAS - SALIDAS): Son Objetos gráficos que se crean para que el usuario ingrese o visualice la información, interactuar con la interfaz; de momento tomaremos los más básicos: <div><div><div>Componentes de una ventana</div><div><div><div></div><div></div><div></div><div></div><div></div></div><div><div>JButton</div><div>JLabel</div><div>TextField</div><div>JCheckBox</div><div>JRadioButton</div></div></div></div></div>		
	Componente	Prefijo	Descripción
2.1.	JLabel	lbl	Etiqueta o mensaje es solamente visual, que normalmente acompaña a otro objeto, para describirlo, también se pueden mostrar valores de salida.
2.2.	TextField	txt	Caja de Texto, donde el usuario ingresará valores de entrada, acuerdo al requerimiento, también se pueden mostrar valores de salida, bloqueando para no modificar su resultado
2.3.	JButton	btn	Un botón se requiere para que el usuario al utilizarlo dispare un evento o acción que ejecutará un cálculo o proceso.
2.4.	JCheckBox	chk	Casilla de verificación, el usuario puede tildar o des-tildar una o varias opciones de una lista; por ejemplo: Hobbies (cine, deporte, tv)
2.5.	JRadioButton	rad	El usuario puede seleccionar una y solamente una opción de una lista; por ejemplo para seleccionar el género de una persona (Hombre - Mujer)

2.6.	Constructores, para crear los componentes JLabel lblEtiqueta = new JLabel (); TextField txtCajaTexto = new TextField (); Button btnBoton = new Button (); En el diseño para cambiar el nombre del objeto creado con el nombre y el prefijo correspondiente es con el mouse derecho sobre el mismo objeto.	
2.7.	<pre>= new JLabel (void String text, Icon icon, int horizontalAlignment String text, int horizontalAlignment String text Icon image, int horizontalAlignment Icon image</pre>	void String text
2.8.	<pre>= new JTextField (void String text int columns String text, int columns Document doc, String text, int columns</pre>	void Int columns
2.9.	<pre>= new JButton (void Icon icon Action a String text, Icon icon</pre>	void String text
2.10.	CONSTRUCTORES	MÉTODOS: Acciones sobre el Objeto
	Sin título JLabel lblEtiqueta = new JLabel ();	Establecer de la tamaño de la ventana: lblEtiqueta.setSize(ancho, alto); lblEtiqueta.setLocation(posX, posY);
	Sin título TextField txtCajaTexto = new TextField ();	Establecer de la tamaño de la ventana: txtCajaTexto.setSize(ancho, alto); Posicionar el Objeto en la coordenada (x, y) de la ventana txtCajaTexto.setLocation (posX, posY);
	Sin título Button btnBoton = new Button ();	Establecer de la tamaño de la ventana: btnBoton.setSize(ancho, alto);

		Posicionar el Objeto en la coordenada (x, y) de la ventana btnBoton.setLocation(posX, posY);
	<p>Tener presente que el único tipo de datos que manejan los componentes son de tipo String y el valor a enviar debe ir entre comillas dobles, para ello se debe tener presente para efectos de procesos los métodos de biblioteca para sus respectivas conversiones:</p> <p>Para extraer el valor existente dentro del objeto</p> <ul style="list-style-type: none"> double numero = Double.parseDouble (nombreObjeto.getText()); int numero = Integer.parseInt (nombreObjeto.getText()); float numero = Float.parseFloat (nombreObjeto.getText()); <p>Para asignar una cadena al objeto diseñado:</p> <ul style="list-style-type: none"> objeto.setText(String.valueOf(valorNumerico)); 	

TALLER DE APLICACIÓN

1.	<p>Crear un programa en Java, que permita dibujar una siguiente GUI, que permita al usuario hallar el área y perímetro de las figuras planas (Rectángulos – Triángulos – Circulo):</p> <ul style="list-style-type: none"> • Aplicar todos los conceptos vistos • Crear un formulario FigurasFormularios • Crear un paquete Formularios • Crear dentro del paquete un JFrame-Form • Crear un Tabbed Pane (Tabs o pestañas) – con el título Cuadrados • Crear un Panel dentro del primer Tab • Crear otro Panel pero fuera del primer Tab – para Triangulos • Crear otro Panel pero fuera del primer Tab y segundo Tab – para Circulos • Dentro de cada Tab y Panel respectivo, crear los componentes para las entradas, salidas y procesos (JLabel, JTextField, Button), ver gráfica de ejmplo para el rectangulo • Nombres de los objetos creados claros y con prefijos • Probar el Diseño
----	---

- Escribir el código respectivo que responda a los eventos de los botones creados

FigurasFormularios - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Services Files

Start Page Design History

To add a component multiple times, select it via click in palette and then Shift-click

Triangulos Circulos Rectangulos

[1]. ENTRADAS [3]. SALIDAS

LADO 1: [4] AREA: 00 [5]

LADO 2: PERIMETRO: 00

TIPO: cuadrado o Rectangulo

Calcular [6] Limpiar

Other Components - Navigator

Form frmFormularios

Other Components

[JFrame]

JTabbedPane1 [JTabbedPane]

tabTriangulos [JPanel]

tabCirculos [JPanel]

tabRectangulos [JPanel]

txtLado1 [JTextField]

txtLado2 [JTextField]

txtArea [JTextField]

lblPerimetro [JLabel]

lblTipoRectangulo [JLabel]

cmdCalcular [JButton]

cmdLimpiar [JButton]

lbl1 [JLabel]

lbl2 [JLabel]

lbl8 [JLabel]

lbl9 [JLabel]

lbl10 [JLabel]

lbl11 [JLabel]

lbl12 [JLabel]

lbl13 [JLabel]

Palette

Swing Containers

Panel

Tabbed Pane

Split Pane

Scroll Pane

Tool Bar

Desktop Pane

Internal Frame

Layered Pane

Swing Controls

Label

Button

Toggle Button

Check Box

Radio Button

Button Group

Combo Box

List

Text Area

Scroll Bar

Slider

Progress Bar

Formatted Field

Password Field

Spinner

Separator

Text Pane

Editor Pane

Tree

Table

Swing Menus

Menu Bar

Menu

Menu Item

Menu Item / CheckBox

Menu Item / RadioButton

Popup Menu

Separator

Swing Windows

Swing Fillers

AWT

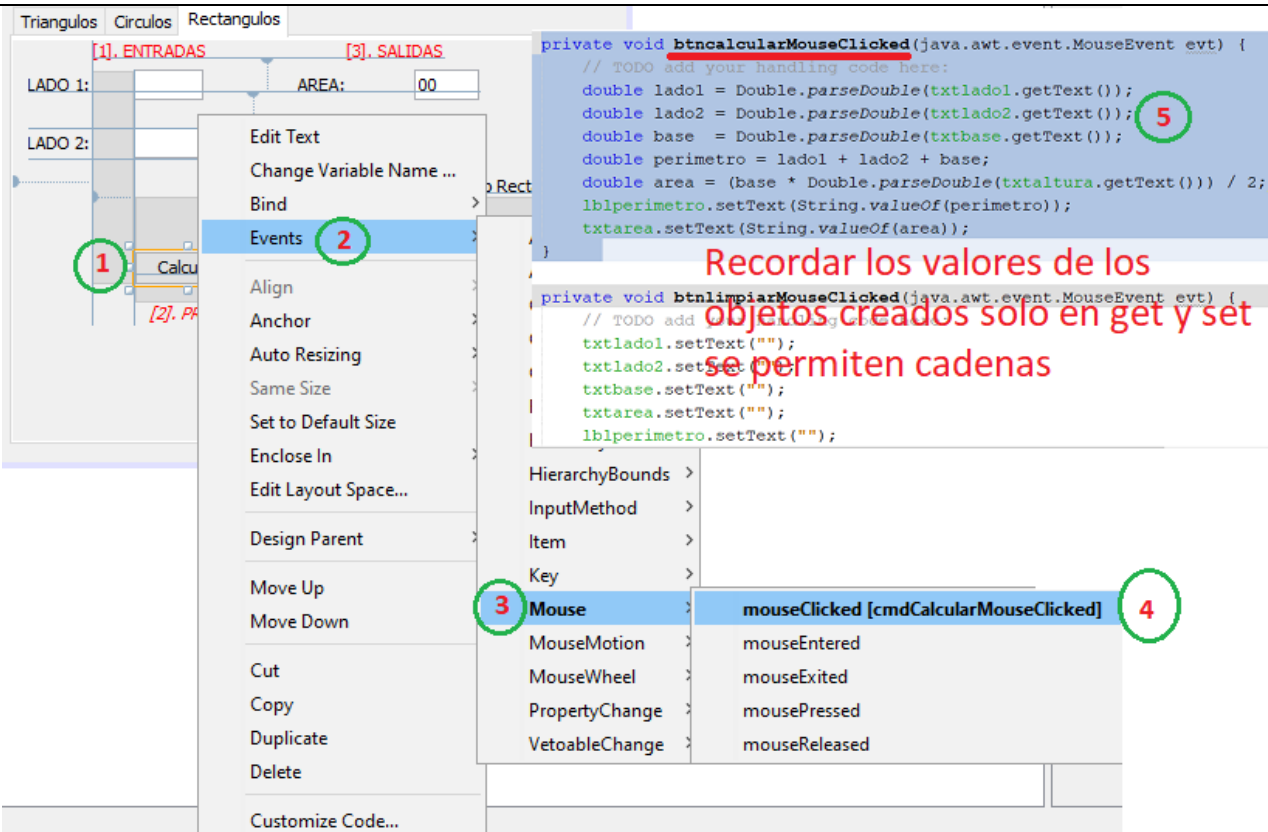
Beans

Java Persistence

Text Field - Cajas Texto
Para entradas y salidas [2]. PROCESOS

Button - Botón
Eventos que ejecutan el código correspondiente a procesos

Label - Etiquetas
que acompañan los componentes
si no los requiere en el código para procesos no requiere renombrar su nombre

	 <p>Recordar los valores de los objetos creados solo en get y set se permiten cadenas</p>
	<p>Validar los campos obligados, por ejemplo:</p> <pre>if (txtlado1.getText().equals("")){</pre> <p>JOptionPane.showMessageDialog (null, "FAVOR COMPLETAR EL LADO 1");</p>
	<p>Validar longitud de del texto y valores solo numéricos:</p> <pre>private void txtlado1KeyTyped(java.awt.event.KeyEvent evt) { char tipoTecla = evt.getKeyChar(); if (! Character.isDigit(tipoTecla)) evt.consume(); int maximo = 3; if (txtlado1.getText().length() > maximo-1) evt.consume(); }</pre>

Tildar al frente de la caja de texto con una imagen de visto bueno, cuando se pase a otro componente (Focus - FocusLost):

- Crear un nuevo paquete de iconos
- Copiar una imagen en el paquete creado de iconos, para el ejemplo **tilde.gif**
- Crear la etiqueta lblTildeLado1
- En la caja de texto respectivo, aplicar Evento - Focus – FocusLost
- Crear al iniciar la clase un objeto global con la figura descargada
- `javax.swing.ImageIcon iconoTilde = new`
`javax.swing.ImageIcon(getClass().getResource("/iconos/tilde.gif"));`
-

```
private void txtlado1FocusLost(java.awt.event.FocusEvent evt) {
    int maximo = 3;
    if (txtlado1.getText().length() >= maximo-1)
        lblTildeLado1.setIcon(iconoTilde);
    else
        lblTildeLado1.setIcon(null);
}
```

TIPOS DE TRIANGULOS - FORMULAS

TODOS LOS COMPONENTES CREADOS, SOLAMENTE TRABAJAN CON TIPO CAD...

LADO 1: 123 ✓ BASE:

LADO 2: | ALTURA:

PERIMETRO: 00

AREA:

LIMPIAR CALCULAR SALIR