

# Manejo de JPanel, JInternalFrame y menus.

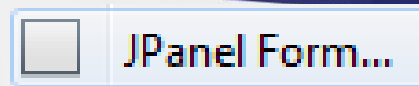
Carrera: Computación e Informática

Semestre: 2016 - I

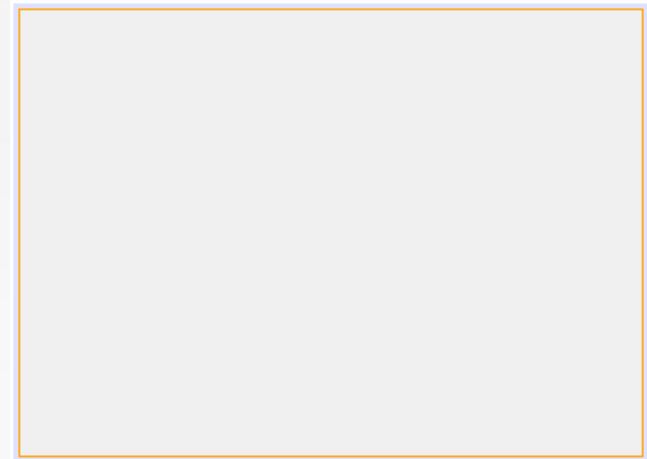
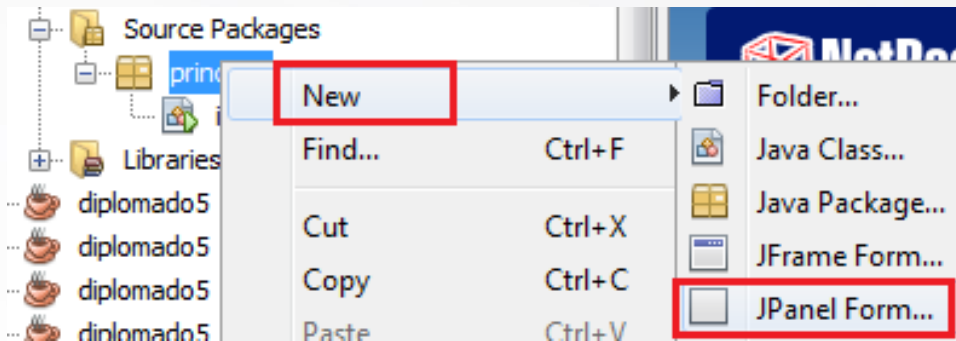
Nombre de Unidad Didáctica: Herramientas para el Desarrollo de Software



# Formulario JPanel



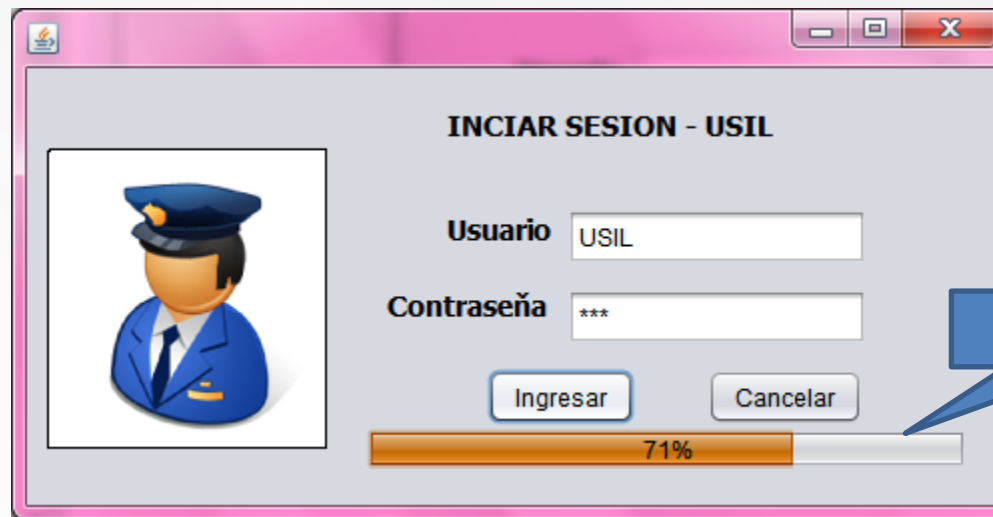
Los **JPanel en Java** son **objetos contenedores de interfaz grafica**, la finalidad de estos objetos es la agrupación de otros objetos tales como **botones, campos de texto, etiquetas, selectores, etc**; una gran ventaja de **Usar JPanel en Java** es que podemos manejar la agrupación de una mejor forma, supongamos que tenemos una serie de botones en un panel, y deseamos desactivarlos todos a la vez, en lugar de hacerlo individualmente con los botones, podemos desactivar el panel y con esto los botones.



# Control ProgressBar

A veces, una **tarea en ejecución** dentro de un programa toma un tiempo para completar. Un programa fácil de usar proporciona alguna indicación al usuario de que se está produciendo la tarea, la duración de la tarea podría tomar, y la cantidad de trabajo que ya se ha hecho. Una forma de indicar el trabajo, y tal vez la cantidad de progreso, es utilizar una imagen animada.

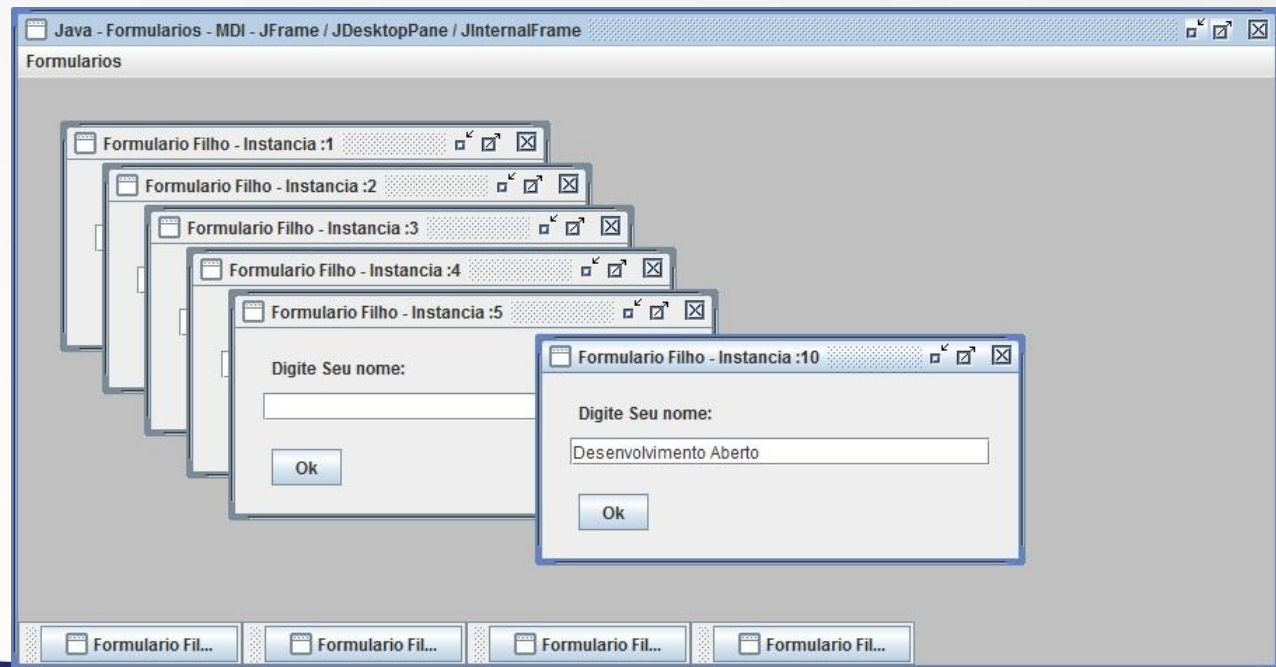
<https://docs.oracle.com/javase/tutorial/uiswing/components/progress.html>



# Control JDesktop Pane

Dentro del uso de los **JFrame** tenemos las aplicaciones **MDI (Multiple Document Interface)** el cual nos permite ingresar diferentes formularios hijos a nuestra aplicación ya sean del tipo **JInterFrame**, **JDialog**, **JOptionPane**, etc.

Para el uso de estas aplicaciones tenemos el uso de un **JDesktopPane** que sirve como contenedor de los diferentes formularios hijos dentro del **JFrame** padre.



# JInternalFrame



JInternalFrame Form...

El **JInternalFrame** es una ventana especial que ni es ventana ni es nada. De hecho, no hereda de **Window**. En realidad es un componente java que se dibuja dentro de otro componente, pero adornado con la barra de título de una ventana y sus botones de maximizar, minimizar y cerrar. Puede incluso arrastrarse y modificarse de tamaño siempre que se mete dentro del componente adecuado.

El sitio bueno para meter los **JInternalFrame** es el **JDesktopPane**. Este panel admite **JInternalFrame** y sabe manejarlos.

```
JFrame ventana = new JFrame();  
JDesktopPane desktop = new JDesktopPane();  
ventana.getContentPane().add(desktop);  
JInternalFrame internal = new JInternalFrame();  
desktop.add(internal);
```

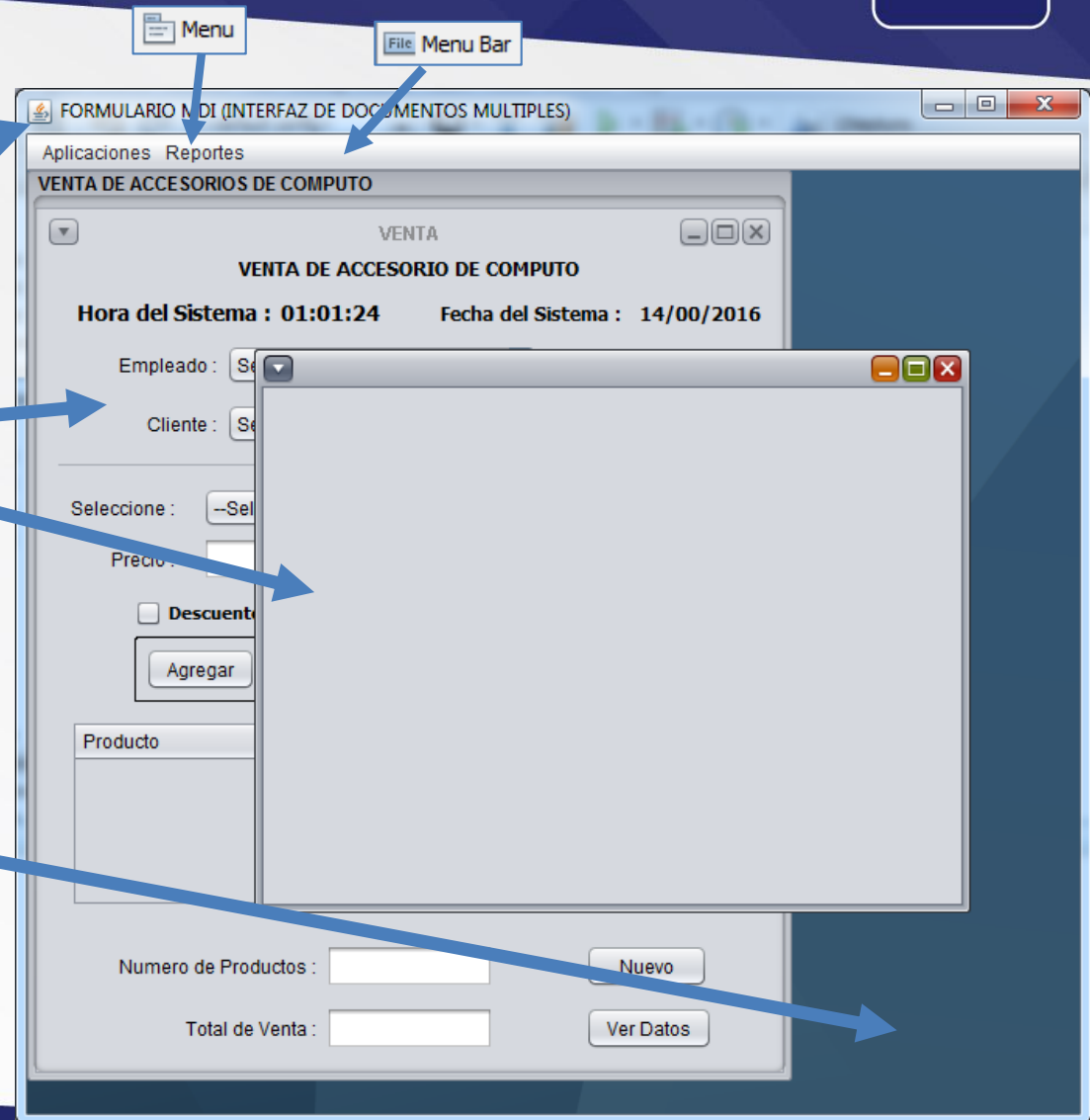
En este código el **JInternalFrame** se comportará como una ventana, pero que no puede salirse del **JDesktop** que la contiene.

# JInternalFrame

JFrame Form...

JInternalFrame Form...

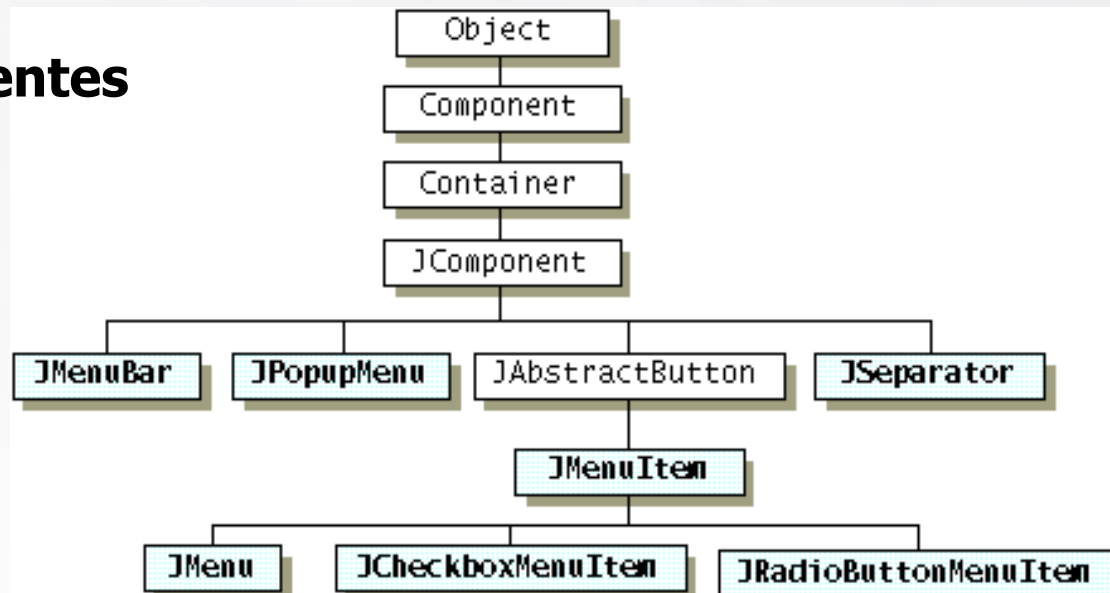
Desktop Pane



# MenuBar , Menu y MenuItem

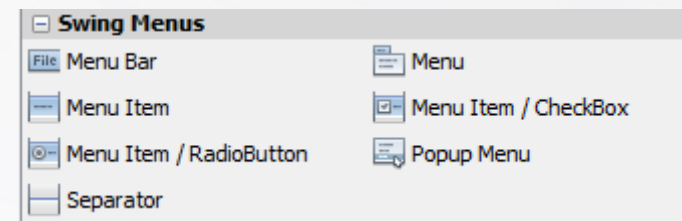
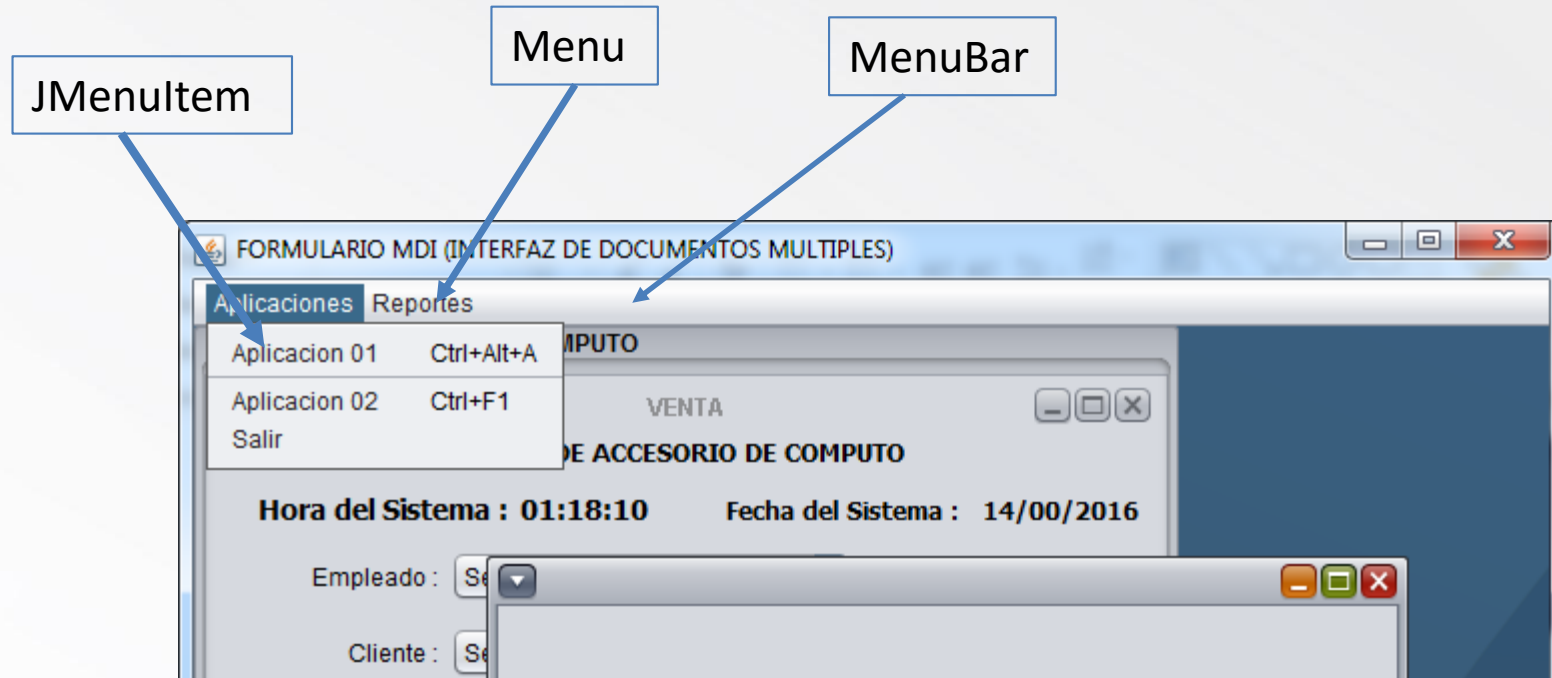
Un menú ofrece una forma de ahorro de espacio para permitir al usuario elegir una de varias opciones. Otros componentes con los que el usuario puede hacer una elección única en muchos incluyen cuadros combinados, listas, botones de radio, y barras de herramientas. Si alguno de sus elementos de menú realiza una acción que se duplica por otro elemento de menú o un botón de herramienta-bar, entonces, además de esta sección debe leer Cómo utilizar las acciones.

## Jerarquía de los componentes del Menu





# MenuBar , Menu y MenuItem





# MUCHAS GRACIAS

