



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA



# Programación Gráfica

---

Agustín J. González  
ELO329/ELO330



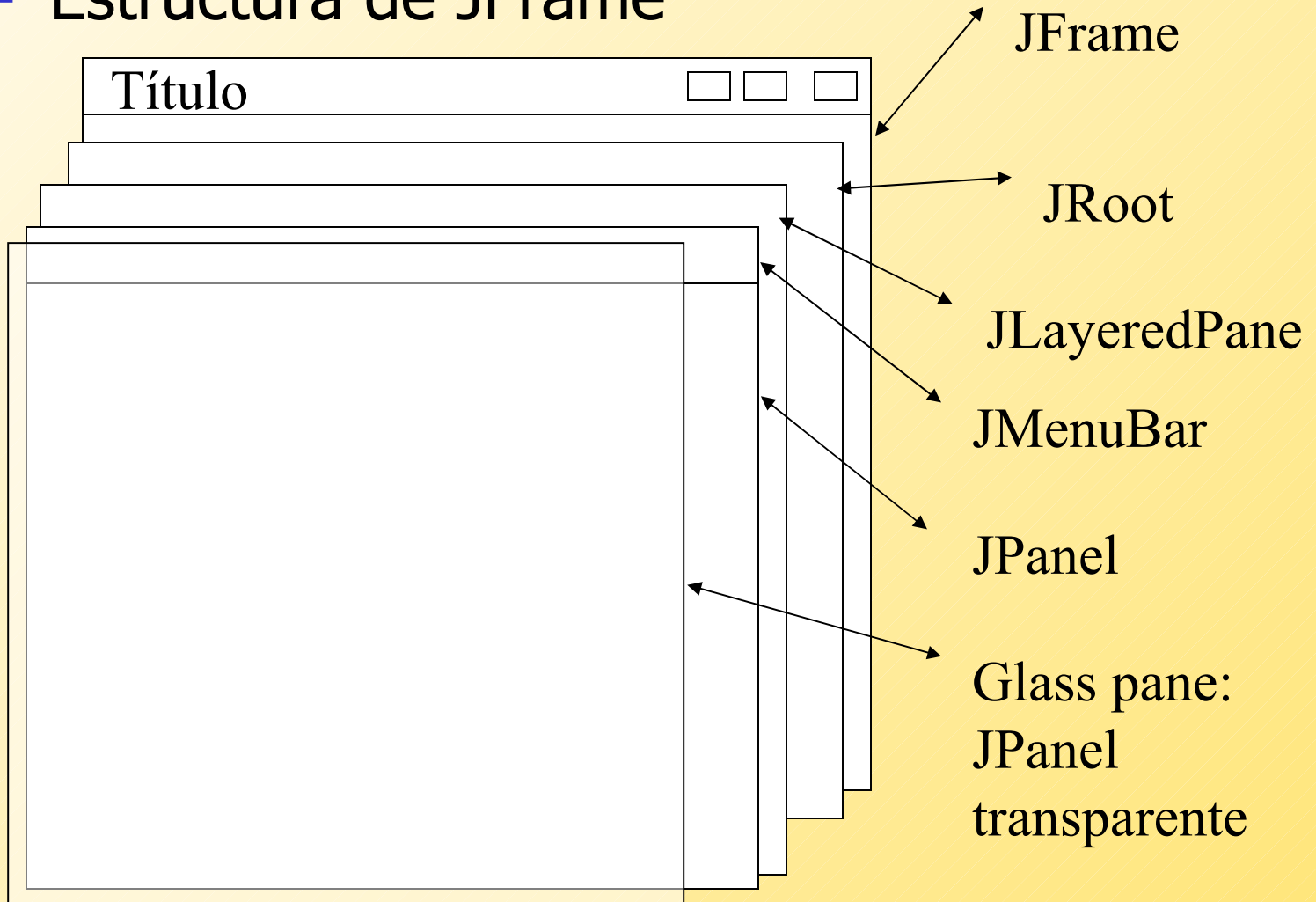
# AWT y Swing

---

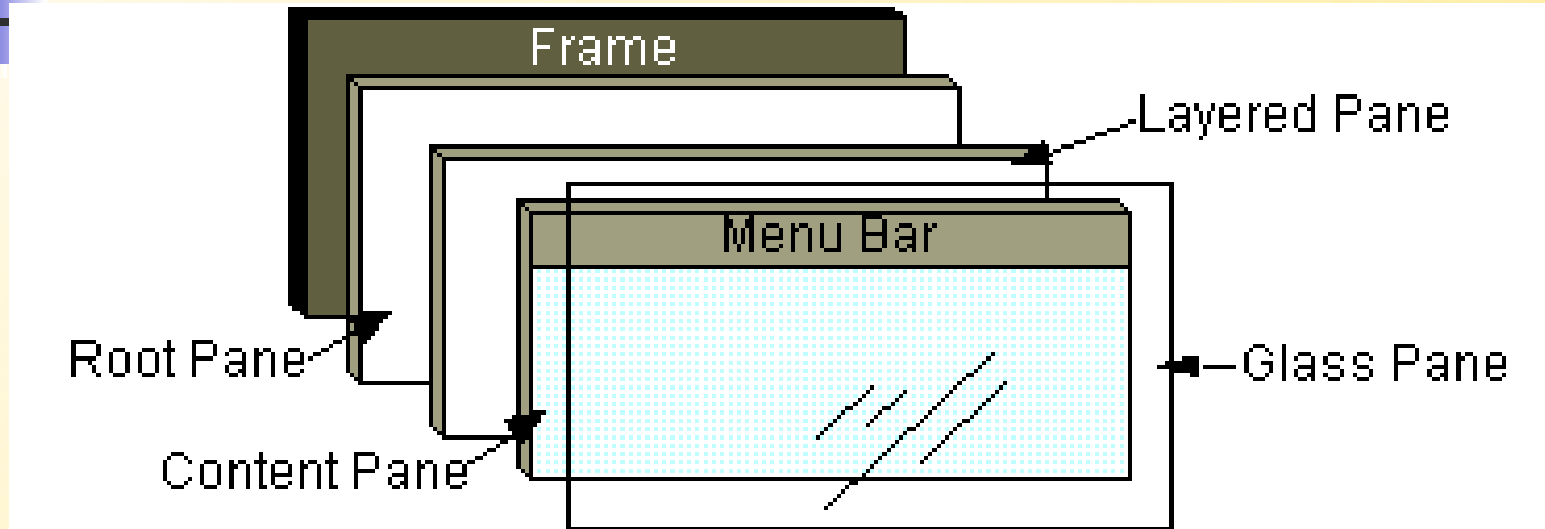
- En sus orígenes Java introdujo la AWT (Abstract Window Toolkit). Ésta “creaba” los objetos delegando su creación y comportamiento a herramientas nativas de la plataforma que corresponda.
- Este esquema condujo a problemas por diferencia en distintas plataformas y S.O.
- La solución fue desarrollar todos los objetos de la GUI basados sólo en elementos muy básicos y comunes en todas las plataformas. Así surge Swing. (Ver demo de la JFC en `/usr/local/jdk/demo/jfc/SwingSet2` en aragorn o en su versión de Java)

# Desplegando información

- Estructura de JFrame



# Estructura de un JFrame



- El RootPane viene con el JFrame. También lo traen los JInternalFrame y los otros contenedores de ventanas superiores (autónomas): JDialog, JApplet, JFrame.
- El root pane tiene 4 partes: vidrio, panel de capas, panel de contenido, y una barra de menú opcional.



# Panel de vidrio

---

- Oculto por omisión (defecto).
- Si se hace visible, es como una hoja de vidrio sobre todas las partes del panel raíz.
- Es transparente, a menos que se implemente un método para pintarlo.
- Intercepta los eventos del panel raíz.



## Layered Pane (panel de capas múltiples)

---

- Contiene una barra de menú opcional y el panel para poner contenidos.
- Puede también contener otras componentes en orden especificado por eje Z (profundidad).
- Ver más detalles en curso tutorial de Swing
- Ver liga desde página del ramo.

# Menús

## ■ Ejemplo

- Crear un frame
- Crear un menubar
- Crear a un menu
- Crear algunos itemes del menu
- Capturar eventos
- Agregar item al menu
- Agregar el menu al menubar
- Incorporar el menubar

```
JFrame f = new JFrame("MenuT");
JMenuBar mb = new JMenuBar();
JMenu menu = new
JMenu("Choose");
JMenuItem item1, item2;
item1 = new JMenuItem("Data 1");
item2 = new JMenuItem("Data 2");
// Action listeners!!
menu.add(item1);
menu.add(item2);
mb.add( menu );
f.setJMenuBar( mb );
```



# Pintado de Componentes

---

- En general hay que tratar de usar componentes estándares de Swing. Ellas se encargan de hacer su (re)pintado en pantalla cuando corresponda.
- Es el caso de Labels, buttons, componentes de texto, icons, borders.
- Si luego de hacer visible una componente, ésta se modifica considerar el llamado a `validate()`.
- Cuando la interfaz posee objetos “dibujados” por la aplicación, considere redefinir el método `protected void paintComponent(Graphics)`. Éste es invocado cada vez que una componente gráfica requiere ser re-pintada.





# repaint

---

- Cuando una componente cambia alguno de sus atributos, por ejemplo un label cambia su texto, este método se encarga de itinerar el repintado de la componente.
- Si por el contrario, la componente no se auto refresca, debemos llamar a `repaint()` para solicitar el llamado a `paintComponent` en forma explícita. Ver ejemplo `IconDisplayer.java`
- Ver ejemplo: [CoordinatesDemo.java](#)



# Java 2D

---

- Java 2D provee gráficos, texto, e imágenes de dos dimensiones a través de extensiones de Abstract Windowing Toolkit (AWT)
- Incluye clases para Rectángulos, Líneas, Elipses.
- La clase `Graphics2D`, a través de su método `draw`, permite dibujar estos objetos debido a que todos ellos implementan la interfaz `shape`.
- Ver demo: `ShapesDemo2D.java`
- Ver ejemplos: `Sketch`, `MouseTest`
- Ejercicio: Hacer diagrama de secuencia cuando se presiona dos veces el botón de creación y luego el de cierre.