

11.A. Introducción a las GUI.

2. Librerías de Java para desarrollar GUI.

2.2. Swing.

Cuando se vio que era necesario mejorar las características que ofrecía AWT, distintas empresas empezaron a sacar sus controles propios para mejorar algunas de las características de AWT. Así, Netscape sacó una librería de clases llamada **Internet Foundation Classes** para usar con Java, y eso obligó a Sun (todavía no adquirida por Oracle) a reaccionar para adaptar el lenguaje a las nuevas necesidades.

Se desarrolló en colaboración con Netscape todo el conjunto de componentes Swing que se añadieron a la JFC.

Swing es una librería de Java para la generación del GUI en aplicaciones.

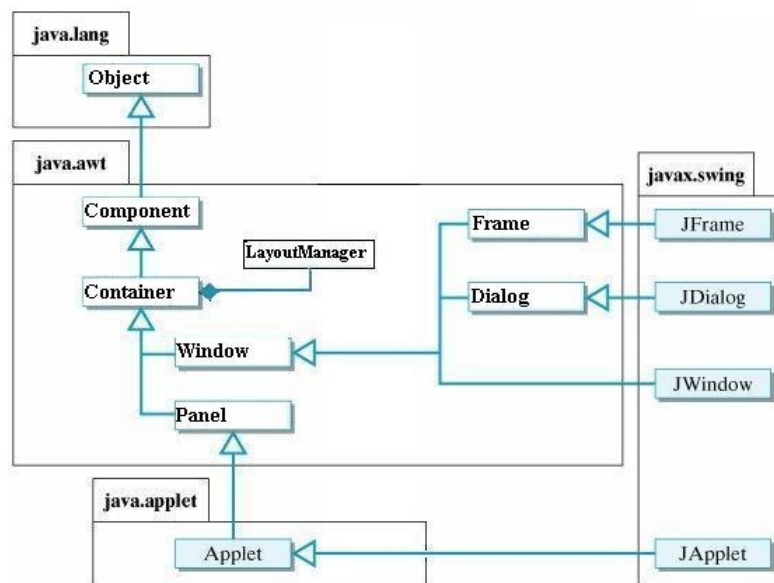


Imagen extraída de curso Programación del MECD.

Swing se apoya sobre AWT y añade **JComponents**. La arquitectura de los componentes de Swing facilita la personalización de apariencia y comportamiento, si lo comparamos con los componentes AWT.

Debes conocer

A continuación tienes un enlace con la lista de los controles Swing más habituales:

[Controles Swing](#)

Por cada componente AWT (excepto `Canvas`) existe un componente Swing equivalente, cuyo nombre empieza por `J`, que permite más funcionalidad siendo menos pesado. Así, por ejemplo, para el componente AWT `Button` existe el equivalente Swing `JButton`, que permite como funcionalidad adicional la de crear botones con distintas formas (rectangulares, circulares, etc), incluir imágenes en el botón, tener distintas representaciones para un mismo botón según esté seleccionado, o bajo el cursor, etc.

La razón por la que no existe `JCanvas` es que los paneles de la clase `JPanel` ya soportan todo lo que el componente `Canvas` de AWT soportaba. No se consideró necesario añadir un componente Swing `JCanvas` por separado.

Algunas características más de Swing, podemos mencionar:

- Es independiente de la arquitectura (metodología no nativa propia de Java)
- Proporciona todo lo necesario para la creación de entornos gráficos, tales como diseño de menús, botones, cuadros de texto, manipulación de eventos, etc.

- Los componentes Swing no necesitan una ventana propia del sistema operativo cada uno, sino que son visualizados dentro de la ventana que los contiene mediante métodos gráficos, por lo que requieren bastantes menos recursos.
- Las clases Swing están completamente escritas en Java, con lo que la portabilidad es total, a la vez que no hay obligación de restringir la funcionalidad a los mínimos comunes de todas las plataformas
- Por ello las clase Swing aportan una considerable gama de funciones que haciendo uso de la funcionalidad básica propia de AWT aumentan las posibilidades de diseño de interfaces gráficas.
- Debido a sus características, los componentes AWT se llaman componentes "de peso pesado" por la gran cantidad de recursos del sistema que usan, y los componentes Swing se llaman componentes "de peso ligero" por no necesitar su propia ventana del sistema operativo y por tanto consumir muchos menos recursos.
- Aunque todos los componentes Swing derivan de componentes AWT, y de hecho se pueden mezclar en una misma aplicación componentes de ambos tipos, se desaconseja hacerlo. Es preferible desarrollar aplicaciones enteramente Swing, que requieren menos recursos y son más portables.