

# ¿Qué es Swing?

Swing es un kit de herramientas GUI Widget para Java. Es una API que proporciona una interfaz gráfica de usuario a los programas Java. A diferencia de AWT, el kit de herramientas Swing ofrece funciones más avanzadas, como animaciones y gestión de eventos.

Inicialmente, Java Swing ofrecía muy pocas opciones de colores y otros ajustes, lo que hacía que toda la aplicación resultara aburrida y monótona. Con el crecimiento del marco Java, se introdujeron nuevos cambios para mejorar la interfaz de usuario y permitir así a los desarrolladores mejorar el aspecto de las aplicaciones Java Swing.

# ¿Qué es Look and Feel?

El «Look» se refiere a la apariencia visual de los widgets GUI, y la «Feel» se refiere al comportamiento de los widgets. El JRE de Sun proporciona los siguientes (L&F):

- CrossPlatformLookAndFeel: es el «Java L&F», también conocido como «Metal», que tiene el mismo aspecto en todas las plataformas. Forma parte de la API de Java (javax.swing.plaf.metal) y es el predeterminado.
- SystemLookAndFeel: aquí, la aplicación utiliza el L&F predeterminado del sistema en el que se ejecuta. El L&F del sistema se determina en tiempo de ejecución, cuando la aplicación solicita al sistema que devuelva el nombre del L&F adecuado. Para Linux y Solaris, los L&F del sistema son «GTK+» si está instalado GTK+ 2.2 o posterior, y «Motif» en caso contrario. Para Windows, el L&F del sistema es «Windows».
- Synth: la base para crear nuestra apariencia con un archivo XML.
- Multiplexing: una forma de hacer que los métodos de la interfaz de usuario deleguen en varias implementaciones de apariencia diferentes al mismo tiempo.

Podemos usar UIManager para cargar la clase L&F directamente desde la ruta de clases. Para ello, escribe el siguiente código en tu método main:

```
UIManager.setLookAndFeel("nombredelaclase");
```

Para ver que L&F tienes instalados corre el siguiente código en tu método main:

```
UIManager.LookAndFeelInfo[] looks = UIManager.getInstalledLookAndFeels();
for (UIManager.LookAndFeelInfo look : looks) {
    System.out.println(look.getClassName());
}
```