



# ***Conceptos útiles sobre Java y Android***

# Conceptos útiles sobre Java y Android

Cuando desarrollamos aplicaciones Android encontramos frecuentemente algunos conceptos mientras desarrollamos. Es recomendable tenerlos claros para no detener nuestro avance durante el curso y que estemos familiarizados con ellos.

## JAVA

Es el lenguaje principal que se utiliza para desarrollar aplicaciones Android es Java. Durante el desarrollo haremos uso en múltiples ocasiones de los siguientes conceptos:

**POJO (Plain Old Java Object):** Es una instancia de una clase que no extiende ni implementa nada en especial. Normalmente posee un constructor, las propiedades definidas y métodos getter y setter para acceder a dichas propiedades.

**Herencia:** Mientras desarrollamos en Android encontraremos que muchas clases extienden de otras que nos brinda el SDK, por ejemplo: 'MainActivity **extends** AppCompatActivity'. La Herencia es un concepto de la programación orientada a objetos que nos ayuda a compartir y reutilizar propiedades y comportamientos entre distintos objetos. Te recomendamos leer la página de wikipedia y buscar más información sobre este concepto: [https://es.wikipedia.org/wiki/Herencia\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Herencia_(inform%C3%A1tica))

**Interfaz (Interface):** En Java una interfaz es un conjunto de métodos y propiedades abstractas. Dentro de la interfaz se define lo que se debe realizar pero no la implementación. Se puede ver una interfaz como un contrato que debe cumplir la clase que la implementa.

## Android Layouts

Al desarrollar nuestras interfaces de UI en Android utilizaremos continuamente distintos Layouts para agrupar y organizar nuestros elementos visuales de la manera que sea necesitamos.

**FrameLayout:** Un FrameLayout está diseñado para llenar un área de la pantalla y desplegar un único elemento dentro de él, ya que puede llegar a ser complicado organizar vistas dentro de él sin que se cubran unas con otras. Sin embargo, se puede lograr a través del atributo `android:layout_gravity` pero te recomendamos que únicamente agreges un elemento a un FrameLayout.

**RelativeLayout:** Es un contenedor de vistas que nos permite organizar las posiciones de nuestros elementos en relación a otros elementos o al padre de estos elementos.

**LinearLayout:** Es un contenedor de vistas que nos permite ordenar los elementos de la vista de manera horizontal o vertical uno tras otro según se lo indiquemos utilizando el atributo `android:orientation`

**ScrollView:** Lo utilizamos para agregar la funcionalidad de scrolling a nuestras vistas en el caso que estas superen, o puedan superar, el tamaño físico en la pantalla. ScrollView es una clase que extiende de FrameLayout, por lo que debería ser utilizada únicamente con un solo elemento hijo. Comúnmente se utiliza un LinearLayout vertical dentro de un ScrollView para permitir hacer Scroll a los elementos del LinearLayout.

### **Archivos importantes en Android:**

Hay archivos dentro de nuestras aplicaciones que, aunque no sean tan recurrentes las veces que los modificaremos, es importante sepamos utilizarlos de la manera correcta y comprendamos su importancia.

**AndroidManifest.xml:** Es el archivo en donde debemos declarar explícitamente las actividades de nuestra aplicación y los permisos que nuestra aplicación requiere. **Tip:** Aquí se configura la imagen que deseamos utilizar como ícono para nuestra aplicación.

**Archivos build.gradle:** Cuando instalamos Android Studio, este viene integrado con un plugin de Gradle que nos permite automatizar la construcción de nuestros proyectos. Dentro de estos archivos configuraremos dependencias a librerías, variables que deseamos exportar, tareas automatizadas, importar plugins, entre otros.

**strings.xml:** Dentro de este archivo debemos colocar todas nuestras cadenas de texto que en algún momento podamos llegar a querer traducir a otros idiomas. Básicamente la importancia de este archivo es para poder localizar nuestras aplicaciones de una manera más sencilla.