

Instalación

En primer lugar, debemos descargar la aplicación de Android Studio para nuestro sistema operativo del siguiente enlace:

<https://developer.android.com/studio/index.html?hl=es-419>

Configuración

En el siguiente enlace podemos seguir los pasos para la configuración de Android Studio en la documentación oficial de Google

<https://developer.android.com/studio/install.html?hl=es-419>

En el caso de instalarlo en el sistema operativo Windows, en algunos sistemas de Windows, la secuencia de comandos de inicio no encuentra el destino de instalación del JDK. Si se produce este problema, debes configurar una variable de entorno que indique la ubicación correcta. Para ello selecciona **Inicio > Equipo >** (botón derecho) **Propiedades > Propiedades avanzadas del sistema**. Luego abre la pestaña **Avanzado > Variables de entorno** y agrega una nueva variable de sistema **JAVA_HOME** que apunte a tu carpeta de JDK. Por ejemplo, C:\Program Files\Java\jdk1.8.0_77.

Enlaces

Algunos enlaces de la documentación oficial de Android sobre la gestión de dispositivos virtuales:

[Crear y administrar dispositivos virtuales:](#)

Hola Mundo

En el siguiente enlace de la documentación oficial de Android se describen los diferentes directorios de un proyecto Android:

[Crear un Proyecto en Android](#)

Build.gradle (Module:app)

```
android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "holamundo.tutorial.com.helloworld"
        minSdkVersion 15
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
            'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'com.google.android.material:material:1.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'
}
```

Commented [EZ1]: Es el nombre del Id con el que se encontrara en googlePlay

Commented [EZ2]: Aqui se pondre las versiones que se suba a googlePlay

Commented [EZ3]: Dependencias o librerías que sean externas

Estructura de ficheros

- **Manifests**

Es Solo un xml de manifiesto en donde, exist configuraciones del modulo móvil

Nombre paquete principal

Icono de aplicación

Label nombre

Temas

Activity principal que actuaria en el aplicativo.

- **java**

Paquetes como principal y paquetes de tests

- **res**

Van todos los recursos

Drawable: parte decorativo iconos, imágenes

Layout: son interfaces de usuarios en la aplicación

Mipmap: iconos de la aplicación

Values: variable de colores

Strings: puede ayudar a dar formato de idiomas ejemplo

Styles: definir estilo que están predefinidos, pero puedes modificarlos.

Activity

Como hemos comentado en el vídeo los Activities son los componentes principales de una app Android, que nos permiten controlar una pantalla de nuestra aplicación.

Ciclo de vida

El estado de los Activities va cambiando en función de las interacciones que el usuario va realizando con el dispositivo en relación con el Activity que se está ejecutando en ese momento.

Enlaces

[Android Developer: Activities y ciclo de vida](#)

Ciclo de vida

OnCreate

OnStart

OnResume

OnPause funciona cuando salimos de la aplicación

Cuando retomamos la aplicación ya no se lanza OnCreate se inicia desde:

OnStart

OnResume

Enlaces

[Android Developer: Depurar tu app](#)

[Escribir y ver registros en logcat](#)

Enlaces

[Iniciar otra Activity](#)

Enlaces

Algunos enlaces de la documentación oficial de Android sobre eventos de entrada:

[Responder a eventos click \(ejemplo en un Botón\)](#)

[Eventos de entrada](#)

Enlaces

Para iniciar un nuevo Activity que controle una pantalla de nuestra app necesitamos lanzar un Intent explícito. En el siguiente enlace de la documentación oficial de Android lo podemos consultar:

[Iniciar otra Activity](#)

Enlaces

[Android Developer: Intents y filtros de intents](#)

[Android Developer: Intent comunes](#)

Aclaraciones

Hay que decir que en Octubre de 2017 Adobe lanzó la versión estable de Adobe XD que hasta ese momento era una versión beta gratuita. Siempre existe la posibilidad de descargar la versión de prueba para probarlo por un período de días gratuitamente.

El hecho de mostrar la aplicación es un ejemplo de que para diseñar una app Android es muy recomendable por no decir imprescindible diseñar un mockup o un prototipo (sea en digital o en papel) antes de empezar a programar.

Esta práctica nos ayuda a definir con detalle qué es lo que queremos programar. En el caso de estar tratando directamente con un cliente para el que vamos a desarrollar una app el prototipo que diseñemos nos ayudará a que el cliente de conformidad y sepa que lo que vamos a desarrollar se ajusta a sus necesidades.

Enlaces

Un enlace a la documentación oficial de Android sobre el componente LinearLayout:

[Diseño lineal \(LinearLayout\)](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre controles de entrada:

[Controles de entrada](#)

[Botones](#)

[Campos de texto](#)

[Casillas de verificación \(checkboxes\)](#)

Enlaces

Enlace la documentación oficial de Android sobre uso de imágenes vectoriales:

[Agregar gráficos vectoriales de varias densidades](#)

Recursos

[Material design \(icons\)](#)

[Flaticons](#)

En res en la carpeta:

Drawable click derecho selecciona vector asset

Gestion de Imágenes!..

Recursos

[Librería Picasso para Android](#)

[Glide](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre ConstraintLayout:

[Build a Responsive UI with ConstraintLayout \(inglés\)](#)

[Comprendiendo los beneficios de utilizar ConstraintLayout \(en inglés\)](#)

[Compilar una IU con el editor de diseño](#)

Recursos

[Material Design: Style](#)

[Adobe Kuler](#)

Manejo de imágenes

<https://bumptech.github.io/glide/>

Enlaces

Algunos enlaces a la documentación oficial de Android sobre Fragments:

[Fragments](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre Fragments:

[Fragments](#)

[ListFragment \(referencia\)](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre comunicación entre Fragments:

[Comunicación con otros Fragmentos](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre diseño multiscreen con Fragments:

[Building dynamic UI con Fragments \(inglés\)](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre el menú Navigation Drawer:

[Creación de un panel de Navegación lateral](#)

La recomendación es cuando se llegue al contenido principal usar Fragmentos y agregar un id de contenedor del fragmento.

Enlaces

Algunos enlaces a la documentación oficial de Android sobre el componente ListView:

[Vista de lista \(ListView\)](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre el componente ListView:

[Vista de lista \(ListView\)](#)

[Crear diseños con un Adaptador](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre el componente GridView:

[Vista de cuadrícula \(GridView\)](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre el componente RecyclerView:

[RecyclerView](#)

Enlaces

Algunos enlaces a la documentación oficial de Android sobre el componente ViewHolder:

[View Holder en RecyclerView \(reference\)](#)

Guardar datos en preferencias

Si deseas guardar una colección relativamente pequeña de pares clave-valor, debes usar las SharedPreferences API. Un objeto SharedPreferences tiene asociado un archivo que contiene pares clave-valor y proporciona métodos simples para leer y escribir dichos pares.

Para más información consultar el siguiente enlace de la documentación oficial de Android Developer:

[Cómo guardar conjuntos clave-valor](#)

Instalación

Podemos tomar las líneas de código que debemos usar para la instalación de Realm en nuestro proyecto Android del siguiente enlace:

[Realm para Android: instalación](#)

Solución errores durante la instalación

Si durante la instalación tenemos algún conflicto con el Gradle y al sincronizar nos aparece un error como éste:

Error:Conflict with dependency 'com.google.code.findbugs:jsr305' in project ':app'. Resolved versions for app (3.0.2) and test app (2.0.1) differ. See <http://g.co/androidstudio/app-test-app-conflict> for details.

Podemos resolverlo aplicando sobre el fichero build.gradle (app) la siguiente línea de código:

```
android { configurations.all { resolutionStrategy.force  
'com.google.code.findbugs:jsr305:x.x.x' } }
```

donde x.x.x debemos completarlo con la versión más actual que aparezca en el mensaje de error. En el ejemplo que os puse la versión que pondría en x.x.x sería 3.0.2, quedando la línea de código de la siguiente manera:

```
android { configurations.all { resolutionStrategy.force  
'com.google.code.findbugs:jsr305:3.0.2' } }
```

BASE DE DATOS INTERNA

Instalación

Podemos tomar las líneas de código que debemos usar para la instalación de Realm en nuestro proyecto Android del siguiente enlace:

[Realm para Android: instalación](#)

Solución errores durante la instalación

Si durante la instalación tenemos algún conflicto con el Gradle y al sincronizar nos aparece un error como éste:

```
Error:Conflict with dependency 'com.google.code.findbugs:jsr305' in project ':app'.  
Resolved versions for app (3.0.2) and test app (2.0.1) differ.  
See http://g.co/androidstudio/app-test-app-conflict for details.
```

Podemos resolverlo aplicando sobre el fichero build.gradle (app) la siguiente línea de código:

```
android { configurations.all { resolutionStrategy.force  
'com.google.code.findbugs:jsr305:x.x.x' } }
```

donde x.x.x debemos completarlo con la versión más actual que aparezca en el mensaje de error. En el ejemplo que os puse la versión que pondría en x.x.x sería 3.0.2, quedando la línea de código de la siguiente manera:

```
android { configurations.all { resolutionStrategy.force  
'com.google.code.findbugs:jsr305:3.0.2' } }
```

Mapas de Google.

Enlaces

Algunos enlaces a la documentación oficial de Android sobre la API de Google Maps para Android:

[Google Maps Android API](#)

[Google Maps: Primeros pasos en Android \(configuración\)](#)

[Obtener una clave API](#)

Iconos click derecho iconosassent
