

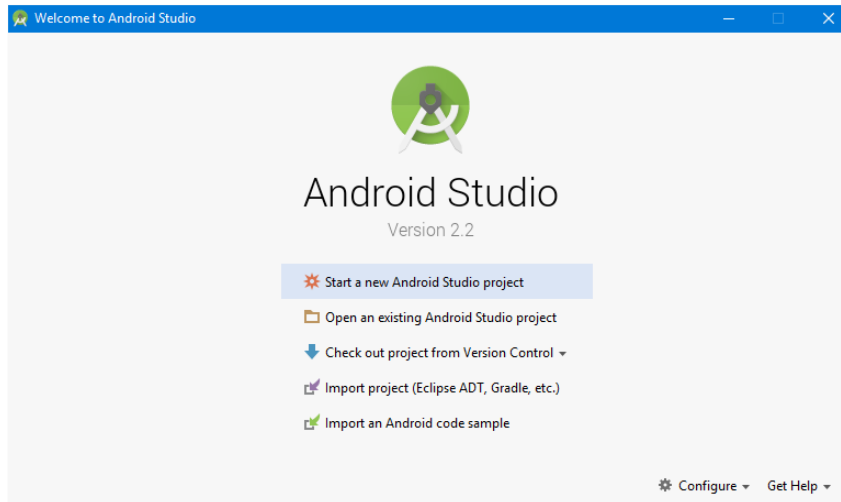


Proyecto AS

Fase del flujo de trabajo



Nuevo proyecto Android

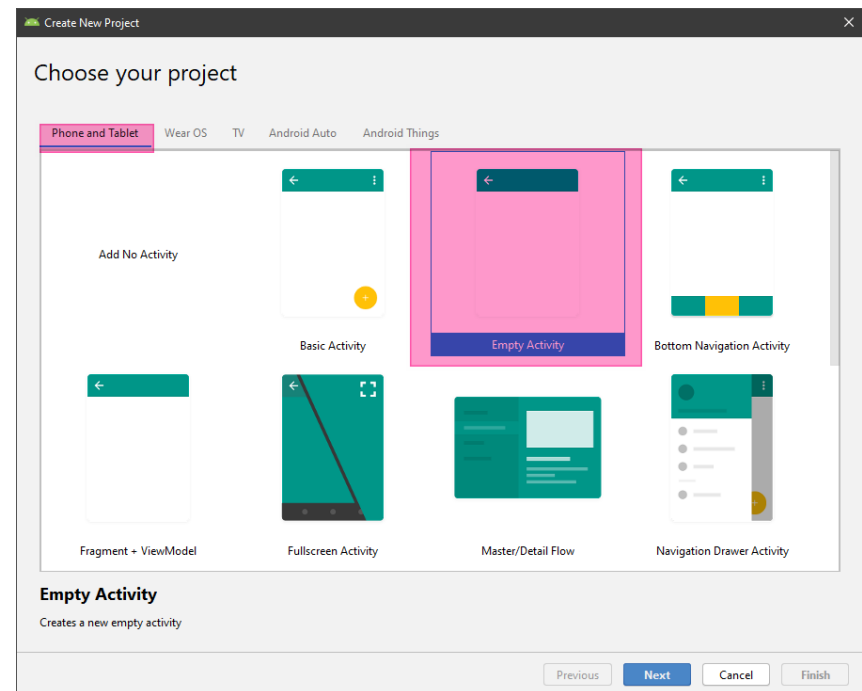


- Para crear un nuevo proyecto ejecutaremos Android Studio y desde la pantalla de bienvenida pulsaremos la opción **“Start a new Android Studio project”** para iniciar el asistente de creación de un nuevo proyecto.
- Si ya habíamos abierto anteriormente Android Studio es posible que se abra directamente la aplicación principal en vez de la pantalla de bienvenida si habíamos salido del IDE sin cerrar el proyecto. En ese caso accederemos al menú **“File / New project...”** para crear el nuevo proyecto.
- El **asistente de creación del proyecto** nos guiará por las distintas opciones de creación y configuración de un nuevo proyecto Android.

Plantillas



- En la primera pantalla del asistente de creación de proyectos, elegiremos el **tipo de actividad principal** de la aplicación (por ahora una *actividad* es una "ventana" o "pantalla" de la aplicación).
- Seleccionaremos *Empty Activity*, que es el tipo más sencillo.
- También fijaremos el **tipo de aplicación** que será: para móviles o tabletas, televisión, reloj, ...



Configuración del proyecto



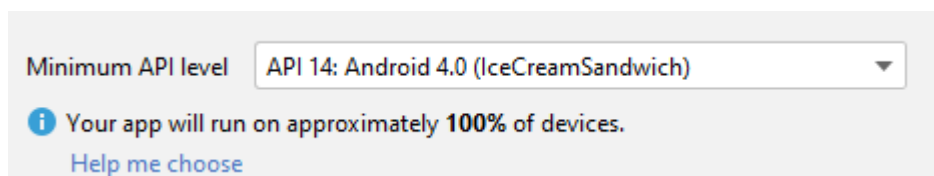
- En la siguiente pantalla indicaremos, por este orden, el **nombre de la aplicación**, el **dominio de la compañía** (sólo se utilizará como paquete de nuestras clases java), la **ruta** donde crear el proyecto, el lenguaje de programación, el objetivo y si es "instant app".

A screenshot of the 'Create New Project' dialog in Android Studio. The dialog is titled 'Configure your project'. On the left, there is a preview of an 'Empty Activity' with a back arrow. On the right, there are several input fields: 'Name' (P_00_Creacion), 'Package name' (com.pdm.p_00_creacion), 'Save location' (Proyectos_AS\Curso_2019\P_00_Creacion), 'Language' (Java), and 'Minimum API level' (API 21: Android 5.0 (Lollipop)). There is a red 'OJO!' warning icon next to the 'Save location' field. Below these fields, there is a note: 'Your app will run on approximately 85.0% of devices.' and a link 'Help me choose'. At the bottom, there are two checkboxes: 'This project will support instant apps' (unchecked) and 'Use androidx.* artifacts' (checked). At the very bottom, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.

Objetivos de la aplicación



- Se fija con el nivel de API mínimo ya que estamos definiendo la versión de sistema operativo mínima que tiene que tener el dispositivo donde lo vayamos a ejecutar. Por defecto nos indica la versión "4.0" que supone casi el 100% de los dispositivos.

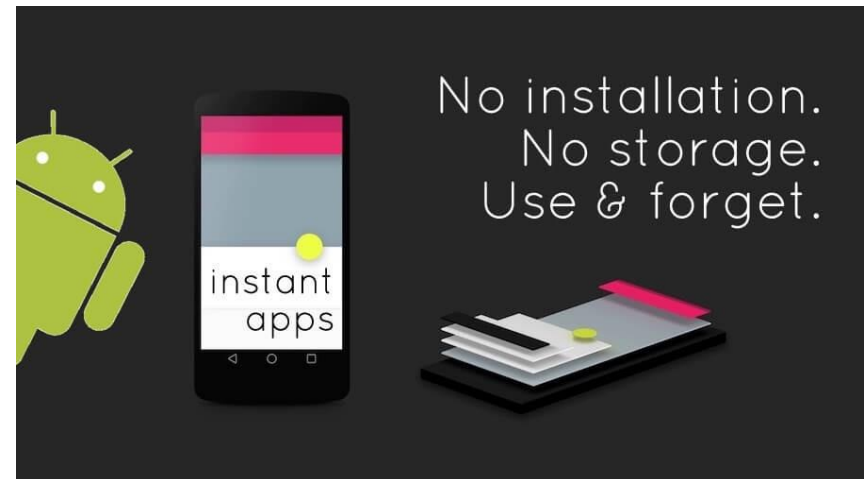


- La versión mínima que seleccionemos en esta pantalla implicará que nuestra aplicación se pueda ejecutar en más o menos dispositivos. De esta forma, cuanto menor sea ésta, a más dispositivos podrá llegar nuestra aplicación, pero más complicado será conseguir que se ejecute correctamente en todas las versiones de Android. Para hacernos una idea del número de dispositivos que cubrimos con cada versión podemos pulsar sobre el enlace "Help me choose", que mostrará el porcentaje de dispositivos que ejecutan actualmente cada versión de Android. Como información adicional, si pulsamos sobre cada versión de Android en esta pantalla podremos ver una lista de las novedades introducidas por dicha versión.
- Desarrollar para versiones anteriores a la recomendada supone no poder utilizar formatos de diseño y objetos modernos. Pero si se desea que también se pueda ejecutar en los anteriores, no hay problema, seleccionamos la API mínima (como la 2.3.3, por ejemplo) y solucionado. Ahora nuestra aplicación podrá ejecutarse en esas versiones, aunque perderemos muchas de las novedades y mejoras.
- AS se encarga de descargar paquetes de plataformas y de compatibilidad (si fueran necesarios).

"Instant app"



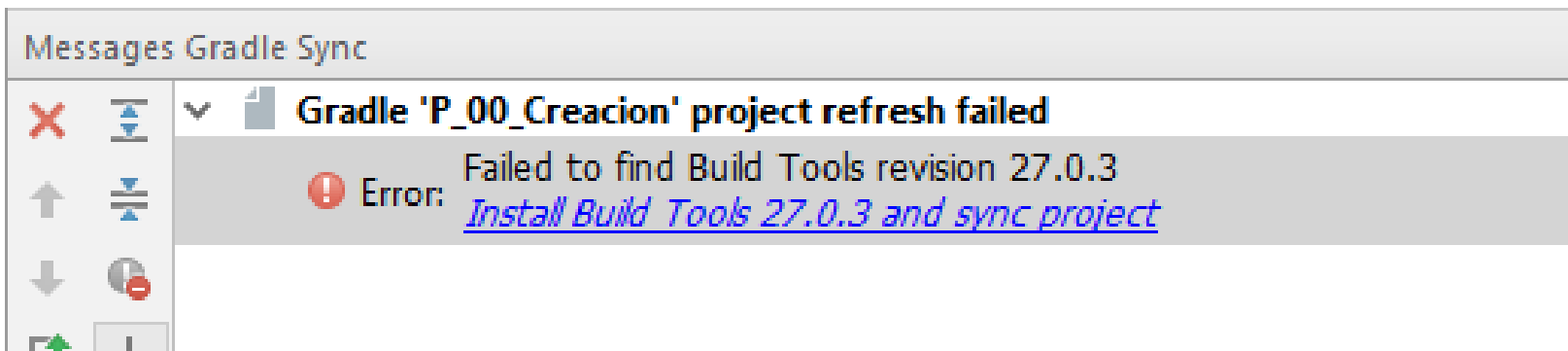
- Aplicación Android que puede utilizarse sin necesidad de que el usuario la instale.
- No necesitan ser instaladas, basta con darles a abrir desde Google Play para que se ejecuten, como si de una página web se tratase.
- [Tutorial de la plataforma](#)



Proyecto en Android Studio




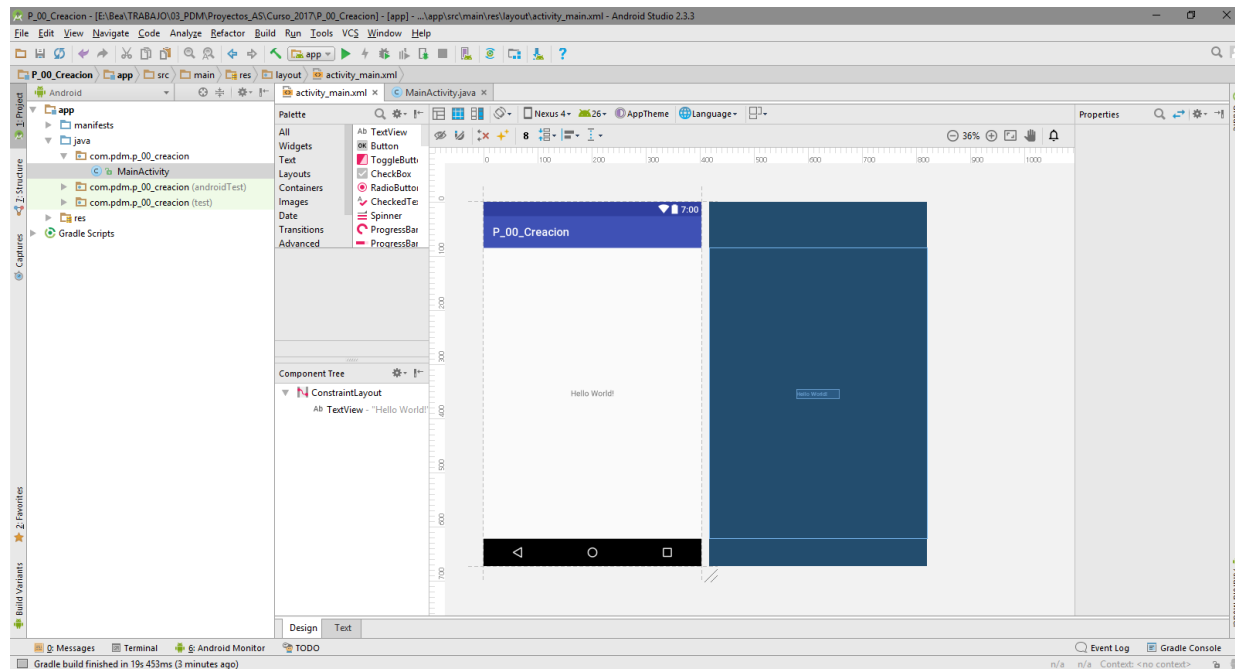
- Android Studio creará por nosotros toda la estructura del proyecto y los elementos indispensables que debe contener.
- Si todo va bien aparecerá la pantalla principal de Android Studio con el nuevo proyecto creado. (Paciencia, sobre todo con el primer proyecto, luego parece que aprende!)
- Si falta algo, AS se encarga de plantear la solución:



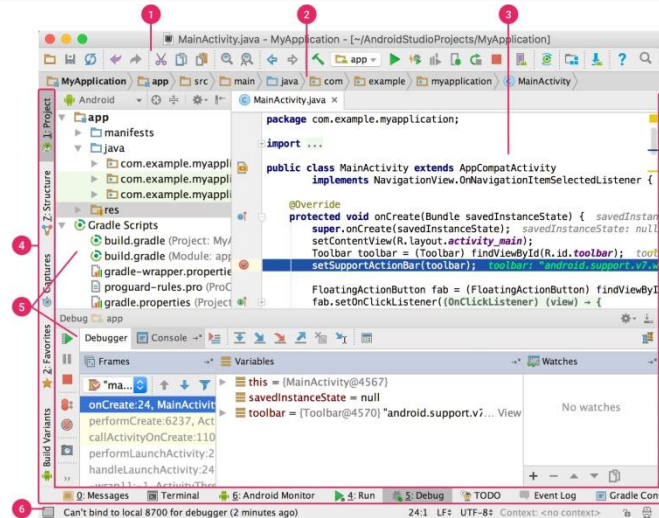
Problemas al cargar el proyecto nuevo



- En ocasiones Android Studio no realiza correctamente esta primera carga del proyecto y es posible que os encontréis con un error del tipo "*Rendering Problems...*" si queremos ver la UI. Para solucionarlo no tenéis más que cerrar la ventana del editor gráfico y volverla a abrir pulsando sobre el fichero "*activity_main.xml*" que podéis ver en el explorador de la parte izquierda o  Tip: Try to refresh the layout.



Conociendo AS

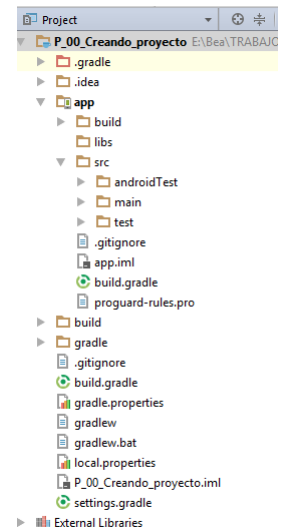
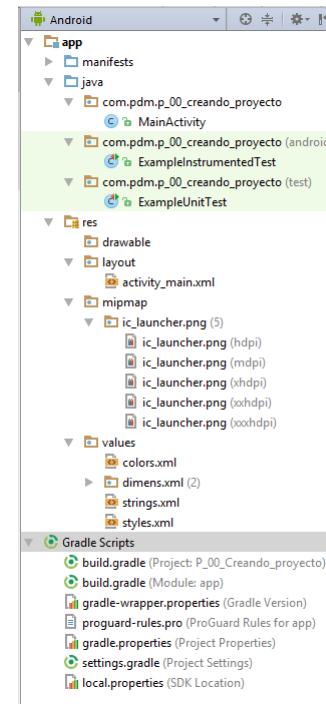


- 1 La **barra de herramientas** te permite realizar una gran variedad de acciones, como la ejecución de tu app y el inicio de herramientas de Android.
- 2 La **barra de navegación** te ayuda a explorar tu proyecto y abrir archivos para editar. Proporciona una vista más compacta de la estructura visible en la ventana **Project**.
- 3 La **ventana del editor** es el área donde puedes crear y modificar código. Según el tipo de archivo actual, el editor puede cambiar. Por ejemplo, cuando se visualiza un archivo de diseño, el editor muestra el editor de diseño.
- 4 La **barra de la ventana de herramientas** se extiende alrededor de la parte externa de la ventana del IDE y contiene los botones que te permiten expandir o contraer ventanas de herramientas individuales.
- 5 Las **ventanas de herramientas** te permiten acceder a tareas específicas, como la administración de proyectos, las búsquedas, los controles de versión, etc. Puedes expandirlas y contraerlas.
- 6 En la **barra de estado**, se muestra el estado de tu proyecto y del IDE en sí, como también cualquier advertencia o mensaje.

Vistas del proyecto



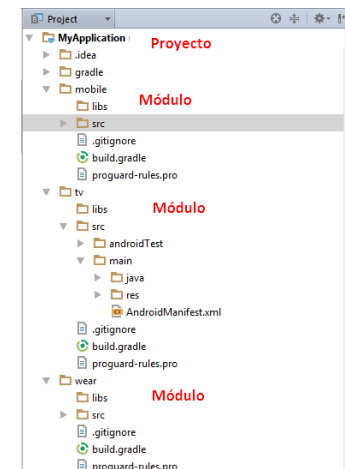
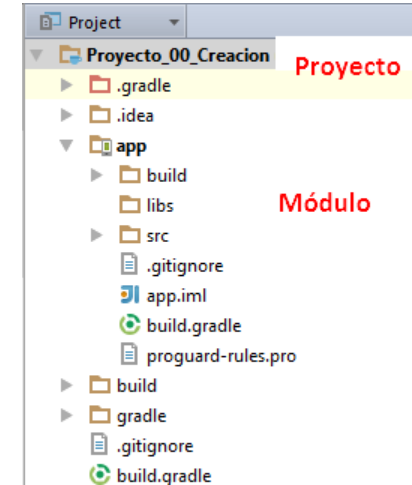
- Un proyecto en Android está formado por un gran número de ficheros almacenados en diferentes carpetas.
- Android Studio proporciona diferentes opciones para visualizar la estructura del proyecto.
- Las opciones principales serían:
 - **Android:** proporciona una vista que facilita el desarrollo de proyectos Android, permitiendo un acceso rápido a los archivos de código fuente clave del proyecto, además de ayudar en el trabajo con el sistema de construcción basado en **Gradle**. **RECOMENDADA**.
 - **Project:** muestra todos los archivos y carpetas correspondientes al proyecto, que es posible localizar dentro de la carpeta **AndroidStudioProjects** de nuestro usuario,
 - **Package:** se muestran los diferentes paquetes incluidos en el proyecto.
 - **Problems:** permite mostrar sólo aquellos archivos que contienen errores.
 - ...



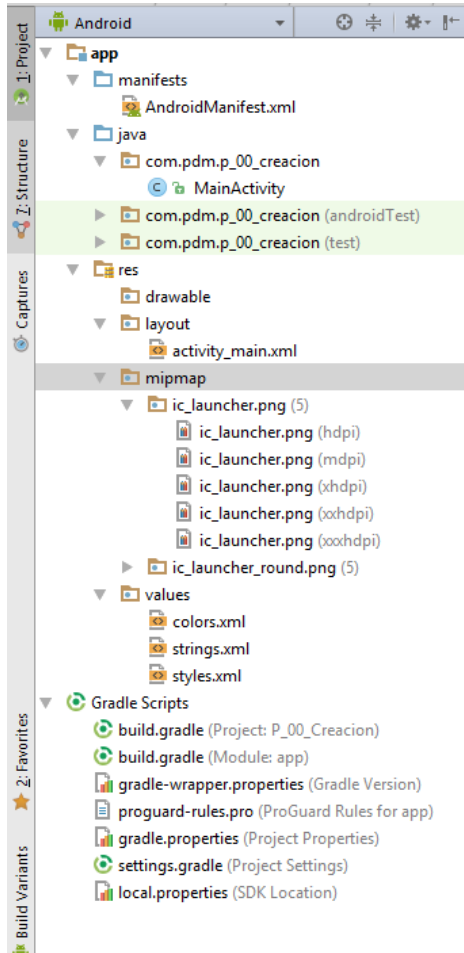
Conceptos



- Lo primero que debemos distinguir son los conceptos de proyecto y módulo.
- La entidad *proyecto* es única, y engloba a todos los demás elementos.
- Dentro de un proyecto podemos incluir varios *módulos*, que pueden representar aplicaciones distintas, versiones diferentes de una misma aplicación (para móvil, para reloj, etc.).



Estructura del proyecto



- **AndroidManifest.xml:** Este fichero describe la aplicación Android.
- **java:** Contiene el código fuente de la aplicación. Como puedes observar los ficheros Java se almacenan en carpetas según el nombre de su paquete.
 - **MainActivity:** Clase Java con el código de la actividad inicial.
 - **ExampleUnitTest y ExampleInstrumentedTest:** Clases Java pensada para insertar código de [testeo](#) de la aplicación.
- **res:** Contiene los recursos usados por la aplicación.
- **Gradle Scripts:** Ficheros Gradle que permiten construir la aplicación.

Fichero AndroidManifest.xml



- Este archivo, el [manifiesto](#) de la aplicación, contiene la información que el sistema operativo necesita conocer antes de poder ejecutar la aplicación.
- Contiene la definición en XML de muchos de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, icono,...), sus componentes (actividades, servicios, etc.) o los permisos necesarios para su ejecución, etc.
- Los asistentes de creación de AS ayudan a su construcción y pocas veces deberemos editarlo manualmente.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest>

  <uses-permission />
  <permission />
  <permission-tree />
  <permission-group />
  <instrumentation />
  <uses-sdk />
  <uses-configuration />
  <uses-feature />
  <supports-screens />
  <compatible-screens />
  <supports-gl-texture />

  <application>

    <activity>
      <intent-filter>
        <action />
        <category />
        <data />
      </intent-filter>
      <meta-data />
    </activity>

    <activity-alias>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </activity-alias>

    <service>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </service>

    <receiver>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </receiver>

    <provider>
      <grant-uri-permission />
      <meta-data />
      <path-permission />
    </provider>

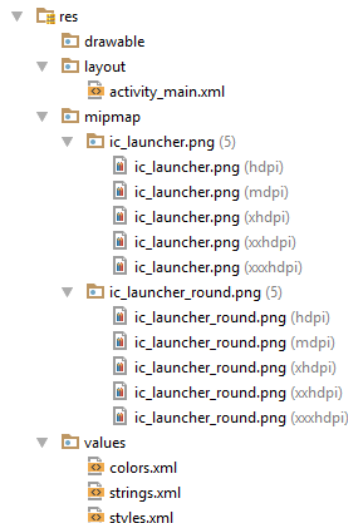
    <uses-library />

  </application>
</manifest>
```

Carpeta res



- Contiene todos los ficheros de [recursos](#) necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc. Los diferentes tipos de recursos de deberán distribuir entre las siguientes carpetas (no todas estas carpetas tienen por qué aparecer en cada proyecto Android, tan sólo las que se necesiten):

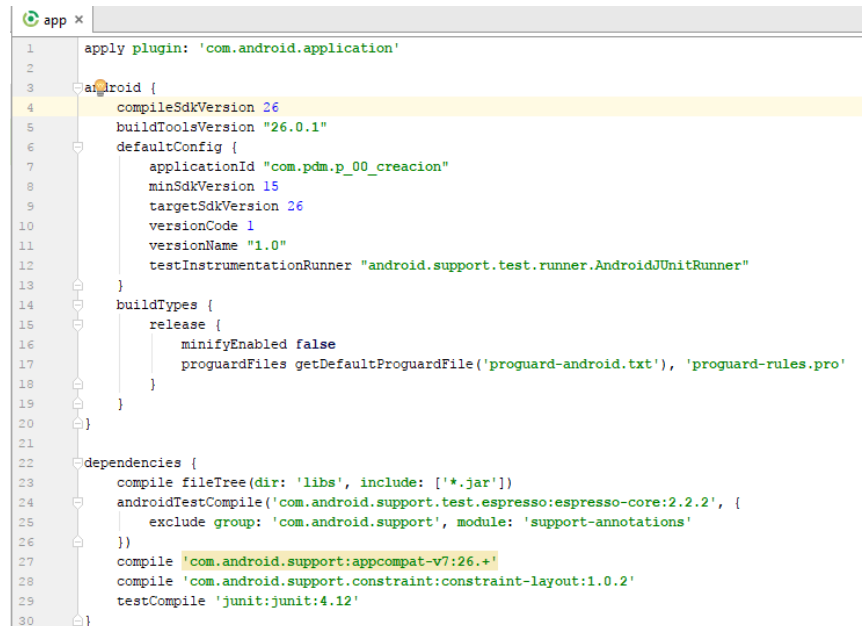
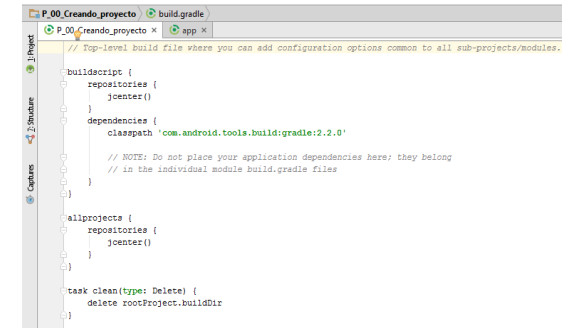


Carpeta	Descripción
/res/drawable/	Contiene las imágenes y otros elementos gráficos usados por la aplicación. Para poder definir diferentes recursos dependiendo de la resolución y densidad de la pantalla del dispositivo se suele dividir en varias subcarpetas: /drawable (recursos independientes de la densidad) <ul style="list-style-type: none">• /drawable-ldpi (densidad baja)• /drawable-mdpi (densidad media)• /drawable-hdpi (densidad alta)• /drawable-xhdpi (densidad muy alta)• /drawable-xxhdpi (densidad muy muy alta :)
/res/mipmap/	Contiene los iconos de lanzamiento de la aplicación (el icono que aparecerá en el menú de aplicaciones del dispositivo) para las distintas densidades de pantalla existentes. Al igual que en el caso de las carpetas /drawable, se dividirá en varias subcarpetas dependiendo de la densidad de pantalla: <ul style="list-style-type: none">• /mipmap-mdpi• /mipmap-hdpi• /mipmap-xhdpi• ...
/res/layout/	Contiene los ficheros de definición XML de las diferentes pantallas de la interfaz gráfica. Para definir distintos layouts dependiendo de la orientación del dispositivo se puede dividir también en subcarpetas: <ul style="list-style-type: none">• /layout (vertical)• /layout-land (horizontal)
/res/anim/ /res/animator/	Contienen la definición de las animaciones utilizadas por la aplicación.
/res/color/	Contiene ficheros XML de definición de listas de colores según estado.
/res/menu/	Contiene la definición XML de los menús de la aplicación.
/res/xml/	Contiene otros ficheros XML de datos utilizados por la aplicación.
/res/raw/	Contiene recursos adicionales, normalmente en formato distinto a XML, que no se incluyan en el resto de carpetas de recursos.
/res/values/	Contiene otros ficheros XML de recursos de la aplicación, como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml), arrays de valores (arrays.xml), tamaños (dimens.xml), etc.

Fichero /app/build.gradle



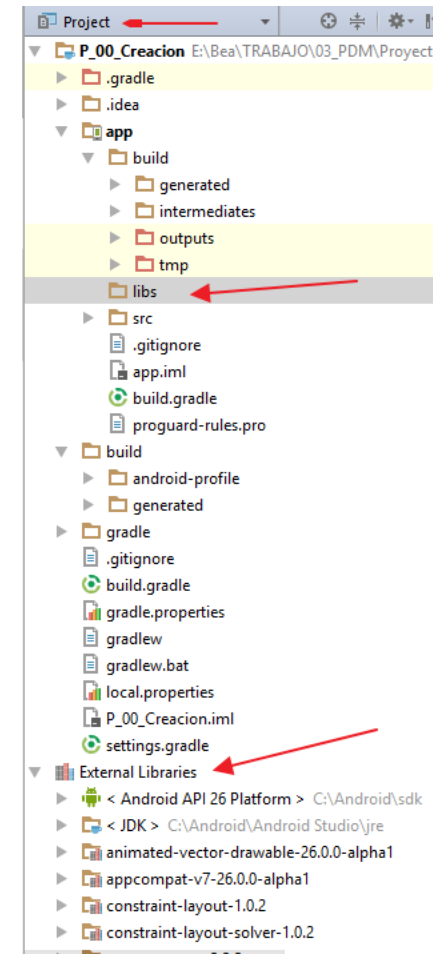
- Contiene información necesaria para la compilación del proyecto, por ejemplo la versión del SDK de Android utilizada para compilar, la mínima versión de Android que soportará la aplicación, referencias a las librerías externas utilizadas, etc.
- En un proyecto pueden existir varios ficheros *build.gradle*, para definir determinados parámetros a distintos niveles. Por ejemplo, en nuestro proyecto podemos ver que existe un fichero *build.gradle* a nivel de proyecto y otro a nivel de módulo dentro de la carpeta /app. El primero de ellos definirá parámetros globales a todos los módulos del proyecto y el segundo sólo tendrá efecto para cada módulo en particular.



Carpeta /app/libs



- Puede contener las librerías java externas (ficheros *.jar*) que utilice nuestra aplicación.
- Normalmente no se incluye directamente aquí ninguna librería, sino que haremos referencia a ellas en el fichero *build.gradle* descrito en el punto anterior, de forma que entren en el proceso de compilación de nuestra aplicación.



Carpeta /app/build/



- Contiene una serie de elementos de código **generados automáticamente** al compilar el proyecto.
- **Importante:** dado que estos ficheros se generan automáticamente tras cada compilación del proyecto es importante que no se modifiquen manualmente bajo ninguna circunstancia.
- A destacar sobre todo el fichero que aparece desplegado en la imagen anterior, llamado **"R.java"**, donde se define la clase R.
- Esta clase R contendrá en todo momento una serie de constantes con los identificadores (ID) de todos los recursos de la aplicación incluidos en la carpeta /app/src/main/res/, de forma que podamos acceder fácilmente a estos recursos desde nuestro código java a través de dicho dato.
- Así, R.layout.activity_main es la constante que contendrá el ID del layout **"activity_main.xml"** contenido en la carpeta /app/src/main/res/layout/.

