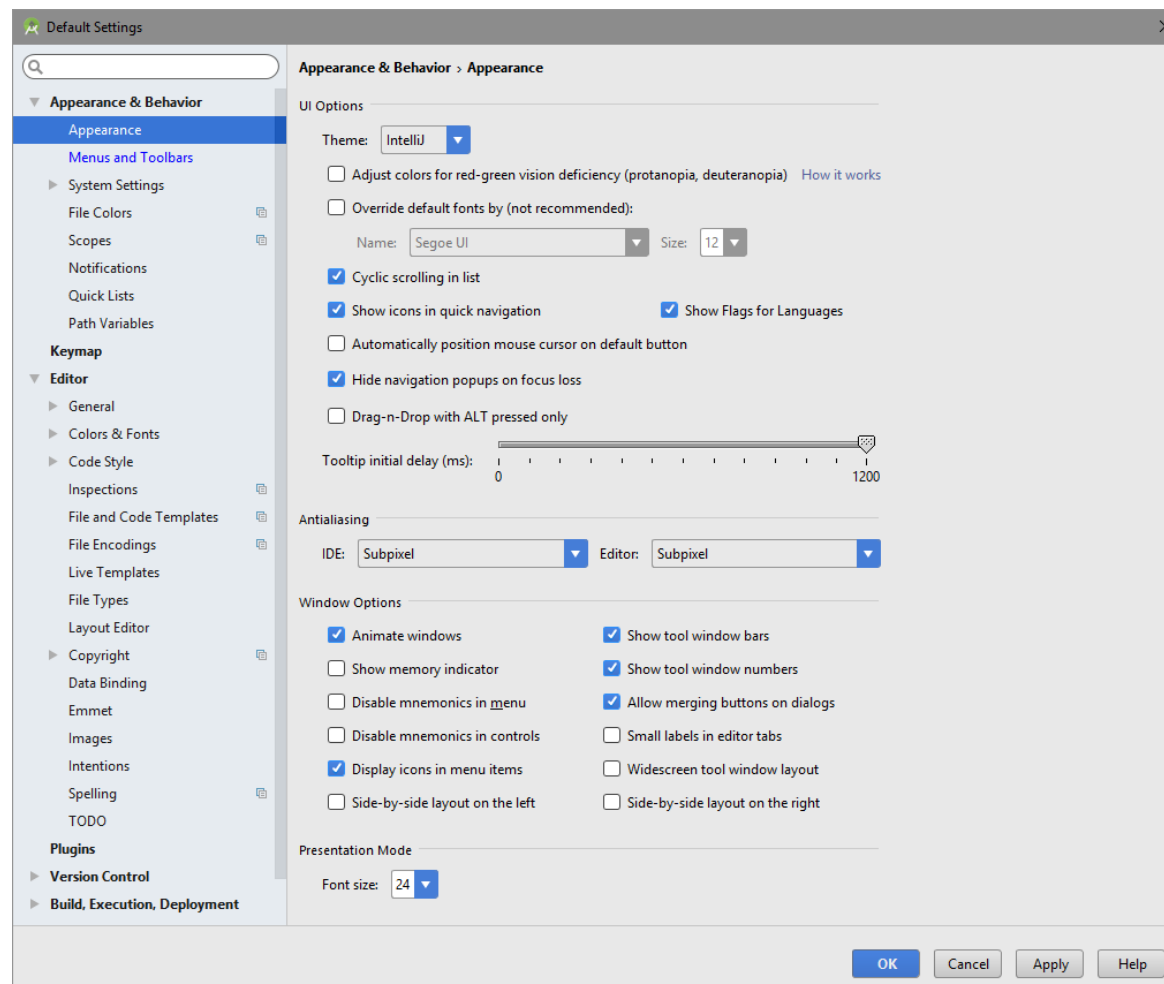


Otras características del IDE Android Studio



Configuración personalizada



Completar código



Completar código

Android Studio ofrece tres opciones para completar código, a las que puedes acceder usando combinaciones de teclas.

Tabla 2: combinaciones de teclas para completar código.

Tipo	Descripción	Windows y Linux	Mac
Compleción básica	Muestra sugerencias básicas para variables, tipos, métodos y expresiones, entre otras. Si llamas a la completación básica dos veces seguidas, verás más resultados. Entre otros, miembros privados y miembros estáticos sin importar.	Control+Espacio	Control+Espacio
Compleción inteligente	Muestra opciones relevantes en función del contexto. La completación inteligente reconoce el tipo y los flujos de datos previstos. Si llamas a la completación inteligente dos veces seguidas, verás más resultados. Por ejemplo, cadenas.	Control+Shift+Espacio	Control+Shift+Espacio
Compleción de enunciados	Completa la instrucción actual agregando elementos faltantes, como paréntesis, corchetes, llaves y formato, entre otros.	Control+Shift+Enter	Shift+Comando+Enter

También puedes realizar correcciones rápidas y mostrar acciones de intención presionando **Alt+Enter**.

Resumen de atajos de teclado

Android Studio Shortcuts You Need the Most

Navigation Shortcuts

Shortcut Description	Android Studio Shortcut
Go to class	Ctrl + N
Go to file	Ctrl + Shift + N
Navigate open tabs	ALT + Left-Arrow; ALT + Right-Arrow
Lookup recent files	CTRL + E
Go to line	CTRL + G
Navigate to last edit location	CTRL + SHIFT + BACKSPACE
Go to declaration	CTRL + B
Go to implementation	CTRL + ALT + B
Go to source	F4
Go to super Class	CTRL + U
Show Call hierarchy	Ctrl + Alt + H
Search in path/project	CTRL + SHIFT + F

Programming Shortcuts

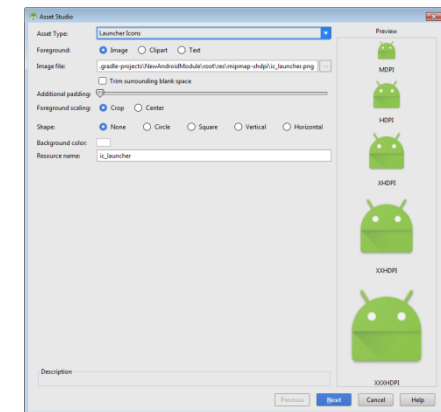
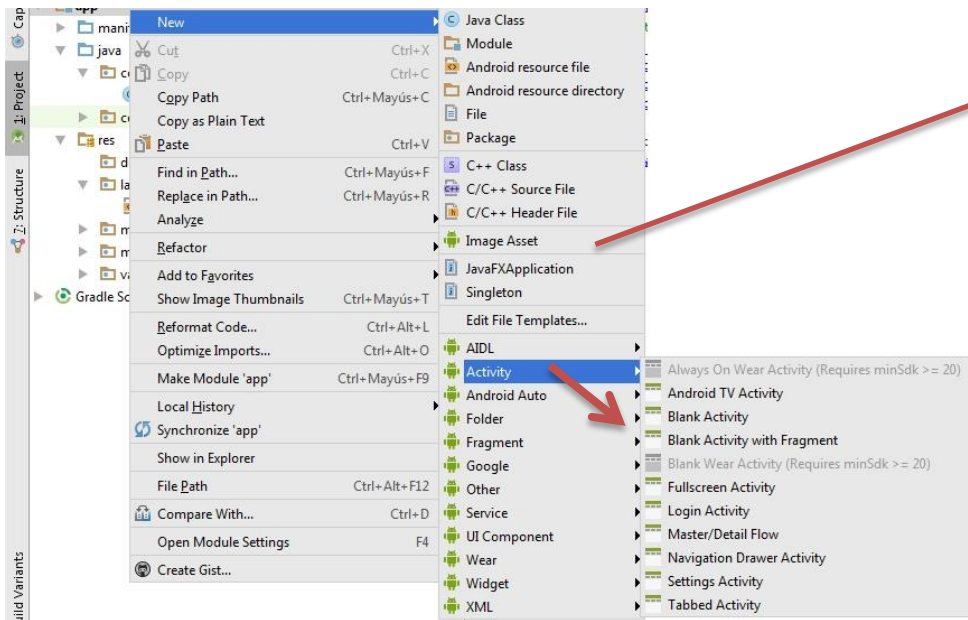
Shortcut Description	Android Studio Shortcut
Reformat code	CTRL + ALT + L
Optimize imports	CTRL + ALT + O
Code Completion	CTRL + SPACE
Issue quick fix	ALT + ENTER
Surround code block	CTRL + ALT + T
Rename and refactor	Shift + F6
Line Comment or Uncomment	CTRL + /
Block Comment or Uncomment	CTRL + SHIFT + /
Go to previous/next method	ALT + UP/DOWN
Show parameters for method	CTRL + P
Quick documentation lookup	CTRL + Q

General Shortcuts

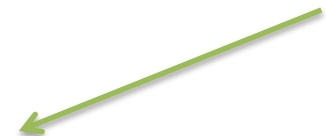
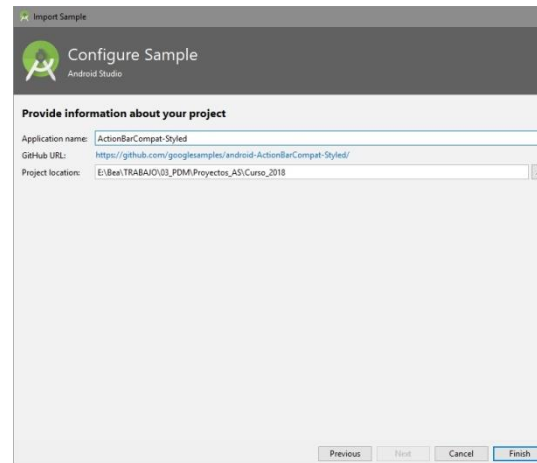
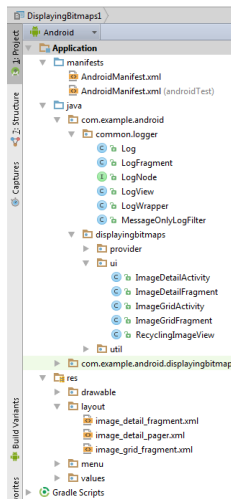
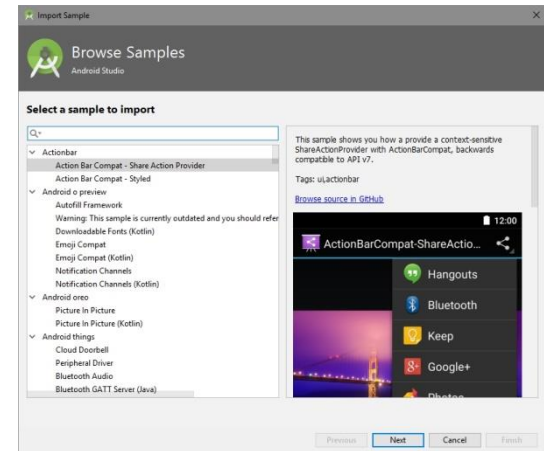
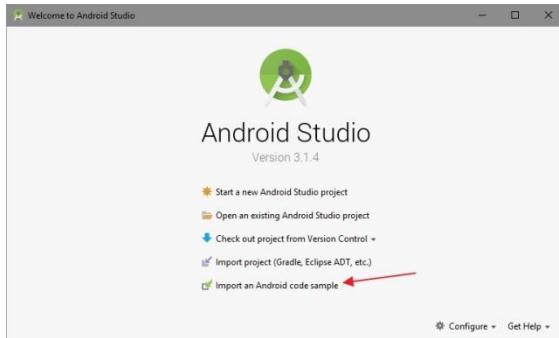
Shortcut Description	Android Studio Shortcut
Delete line	CTRL + Y
Safe Delete	Alt + DELETE
Close Active Tab	CTRL + F4
Build and run	SHIFT + F10
Build	CTRL + F9
All purpose (Meta)Shortcut	CTRL + SHIFT + A

Plantillas y asistentes

- Android Studio proporciona a los programadores plantillas y asistentes para acelerar el desarrollo. Bien!!!



Ejemplos de código en Github



Desgraciadamente,
a veces fallan!

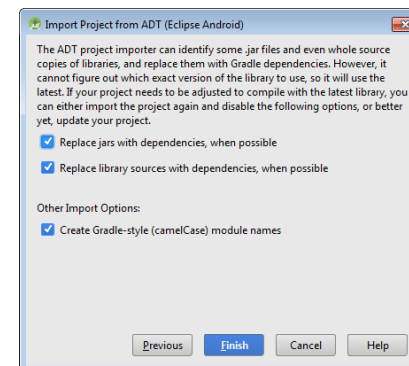
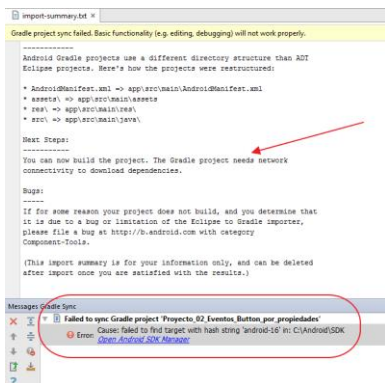
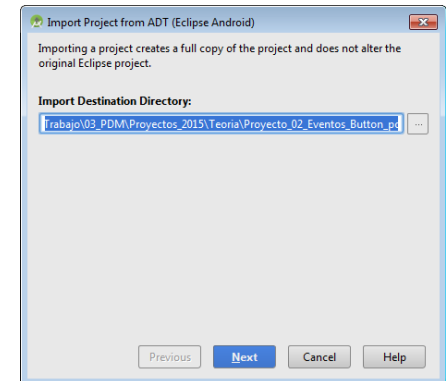
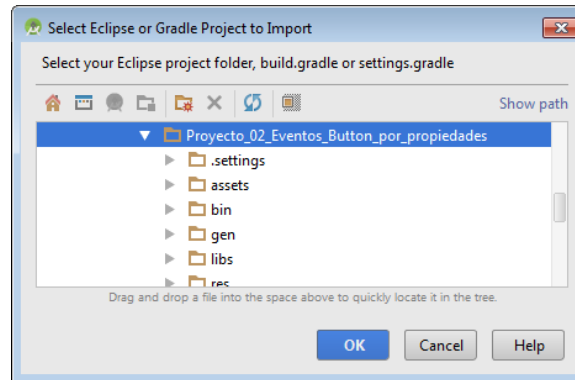
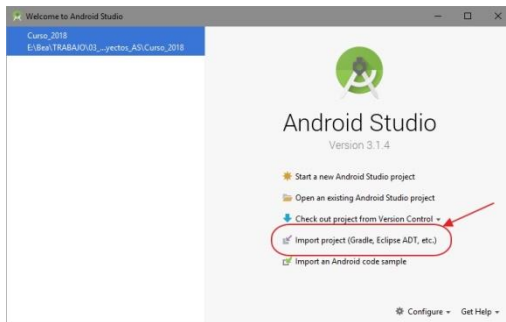
Proyectos Eclipse



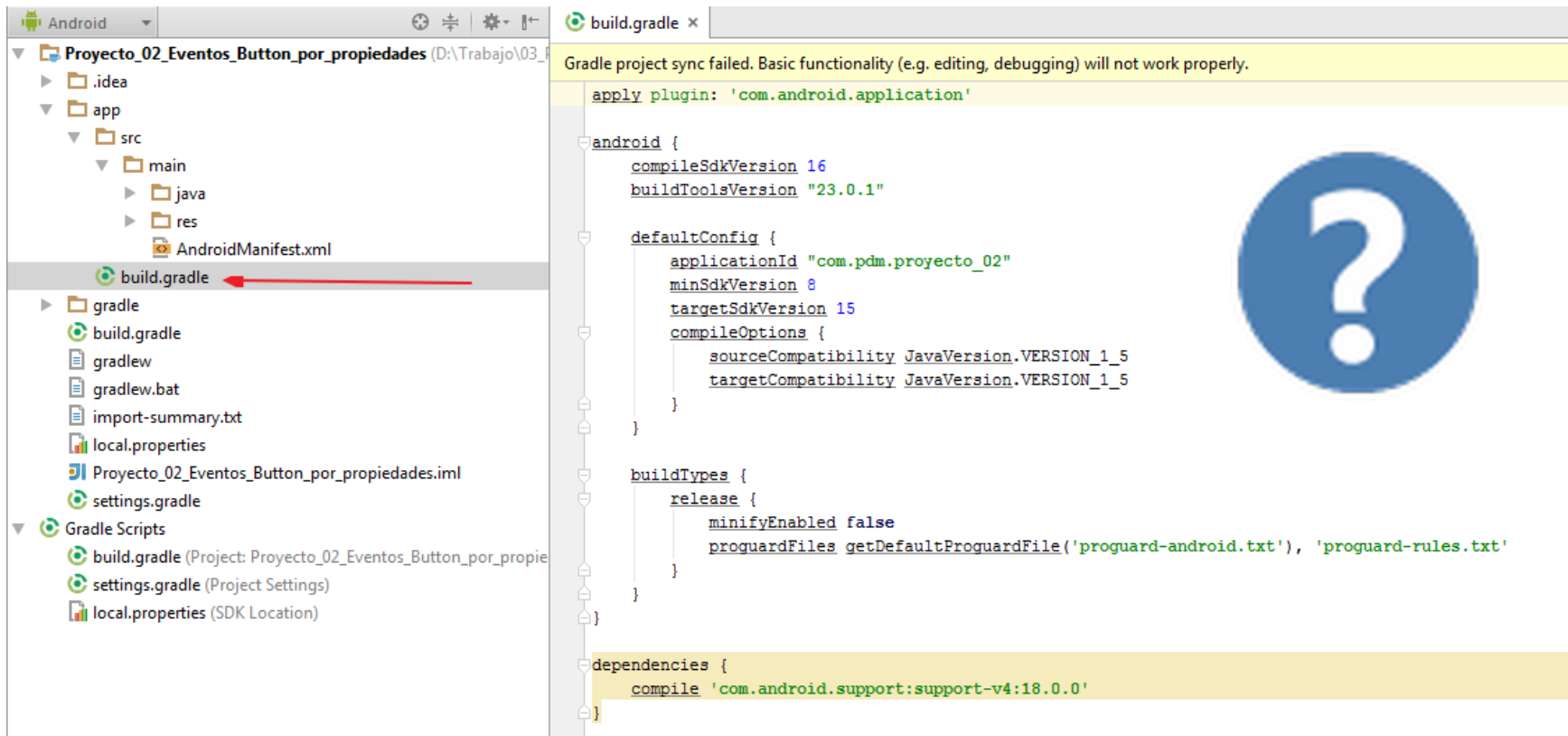
- **Android Studio** es ya el IDE Oficial de Android, al ser más potente, versátil y actual que Eclipse.
- Pero hay muchos proyectos para Android desarrollados con Eclipse. En las empresas, en Internet,...
- Qué ocurre si queremos trabajar con ellos, ya sea para modificarlos, estudiarlos, etc.?



Importación directa



Causas del error



Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 16
    buildToolsVersion "23.0.1"

    defaultConfig {
        applicationId "com.pdm.proyecto_02"
        minSdkVersion 8
        targetSdkVersion 15
        compileOptions {
            sourceCompatibility JavaVersion.VERSION_1_5
            targetCompatibility JavaVersion.VERSION_1_5
        }
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.txt'
        }
    }
}

dependencies {
    compile 'com.android.support:support-v4:18.0.0'
}
```

Solución

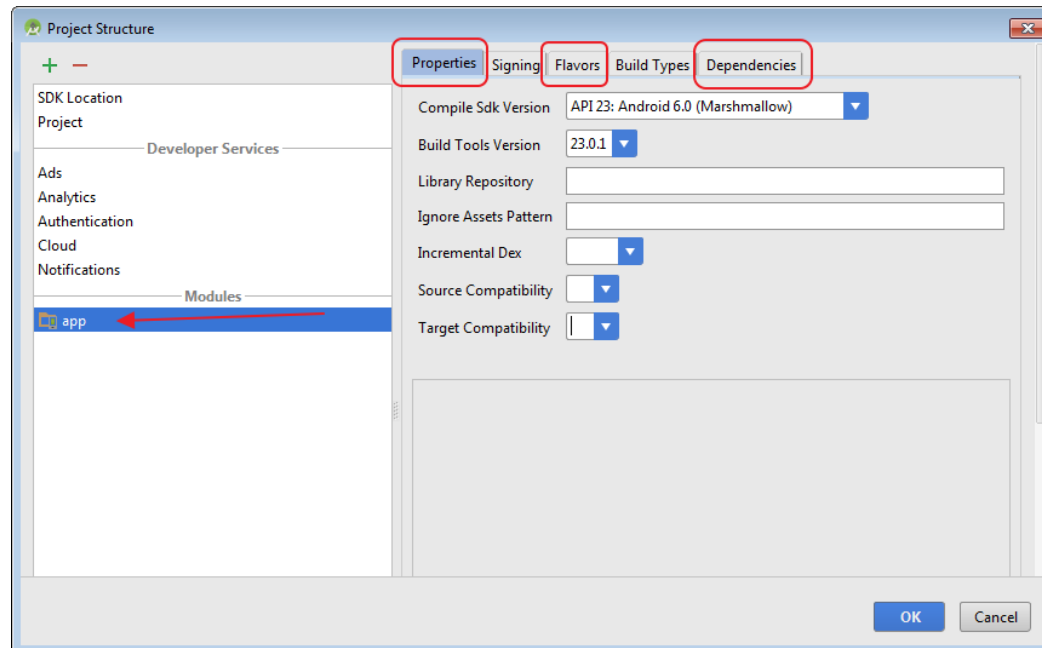


- Hay que configurar convenientemente el fichero build.gradle de nuestra app:
apply plugin: 'com.android.application'

```
android {  
    compileSdkVersion 23  
    buildToolsVersion "23.0.1"  
  
    defaultConfig {  
        applicationId "com.pdm.proyecto_02"  
        minSdkVersion 14  
        targetSdkVersion 23  
    }  
  
    buildTypes {  
        release {  
            minifyEnabled false  
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.txt'  
        }  
    }  
}  
  
dependencies {  
    compile 'com.android.support:support-v4:23.0.1'  
}
```

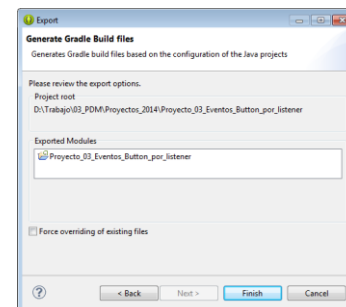
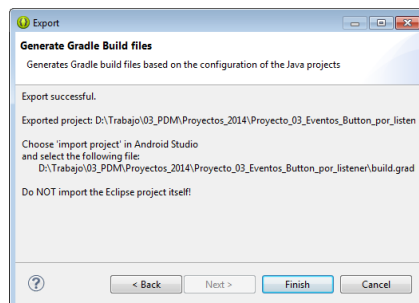
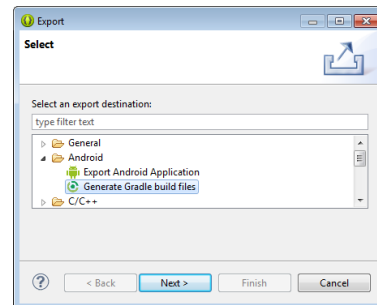
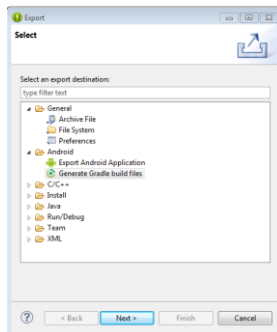
Solución 2

- El fichero también puede configurarse haciendo uso de la herramienta gráfica (método RECOMENDADO):
- File -> Project Structure:



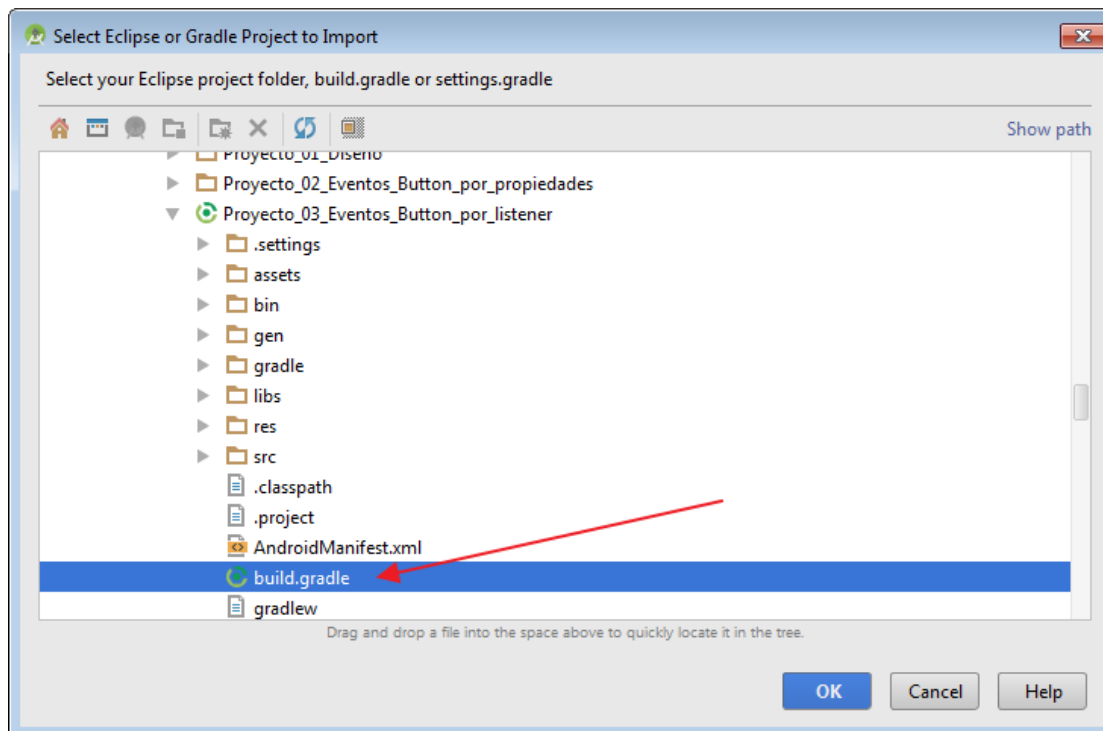
Preparando en Eclipse

- Quizás el desarrollador ha tenido la gentileza de preparar desde Eclipse la migración:



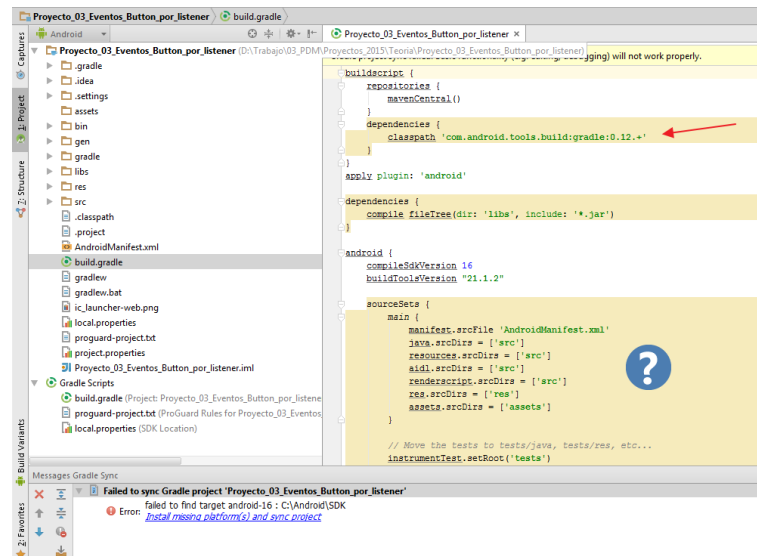
Importando proyecto preparado

- Importamos eligiendo el fichero build.gradle generado en Eclipse:



Corregir errores

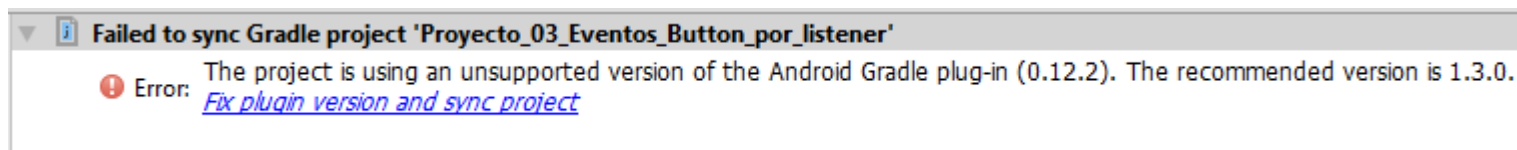
- Seguimos teniendo que corregir errores y además la estructura del proyecto es un poco "eclipsiana": ha desaparecido el build.gradle del módulo y solo hay uno relativo a la aplicación (cuando sabemos que no es la manera recomendada por Android)



Fijar versión de gradle



- Después de corregir los errores aunque la estructura del proyecto ya es algo más correcta, tendremos que fijar la versión de gradle a utilizar porque seguramente la que utilice Eclipse esté desfasada:

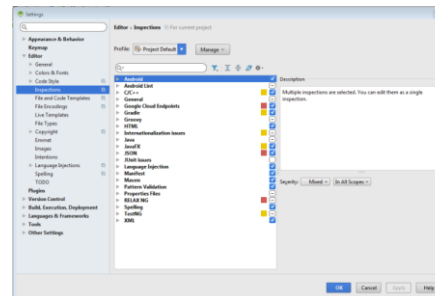


- Basta con clickar en "Fix Plugin and"

Consumo de batería



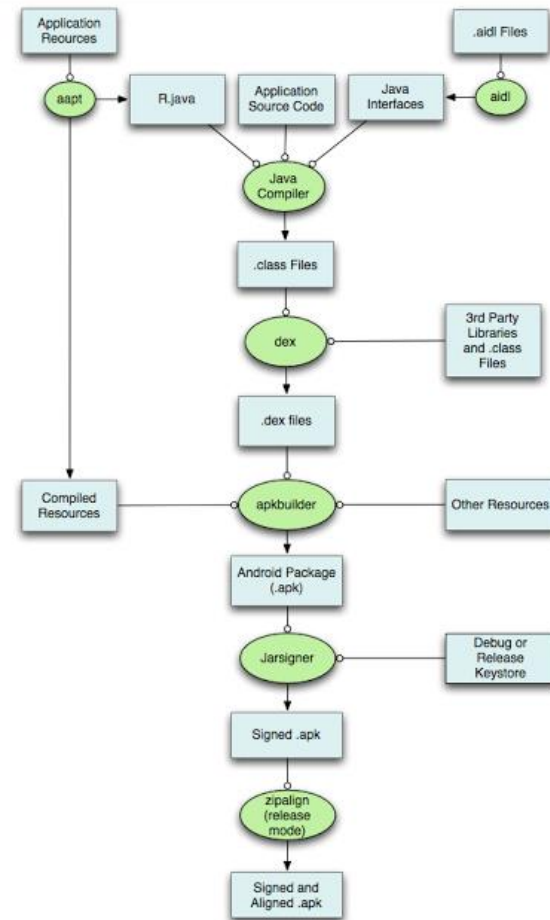
- Android Studio está constante inspeccionando elementos en segundo plano con el consiguiente consumo de batería en portátiles.
- Algunas de las tareas que realiza pueden comprobarse desde "File->Settings->Editor->Inspections":



- Solución:
 - "File ->Power save mode" para desactivar la inspección automática
 - "Analyze->Inspect code" cuando deseamos que se realice la inspección.
 - Obviamente, perdemos una de las mejoras características de AS, el análisis "en vivo".

Pruebas lentas

- Una de las situaciones más desesperantes desarrollando para la plataforma Android es lo lento que puede resultar compilar el proyecto y probarlo en el emulador (sobre todo en equipos bajos de memoria).



Configuración personalizada a fondo



- [Enlace](#)

Buscar tus archivos de configuración

Personalizar las opciones del VM

Tamaño máximo de montón

Exportar e importar configuraciones de IDE

Personalizar tus propiedades de IDE

Configurar el IDE para equipos de memoria insuficiente

Configurar tu proyecto para Instant Run

Establecer la versión de JDK

Establecer configuraciones de proxy

Configurar el proxy de Android Studio

Complemento de Android para las configuraciones de proxy HTTP de Gradle

Consejos básicos



- Configurar antivirus para que no revise carpeta de proyectos
- Un solo proyecto abierto
- "Clean Project" antes de cada ejecución en el emulador (desde pestaña Build)
- Dispositivo real mejor que emulador Android
- Emulador GeanyMotion mejor que emulador Android ?
- No cerrar el emulador después de usar una vez, utilizar el mismo emulador para ejecutar la aplicación cada vez



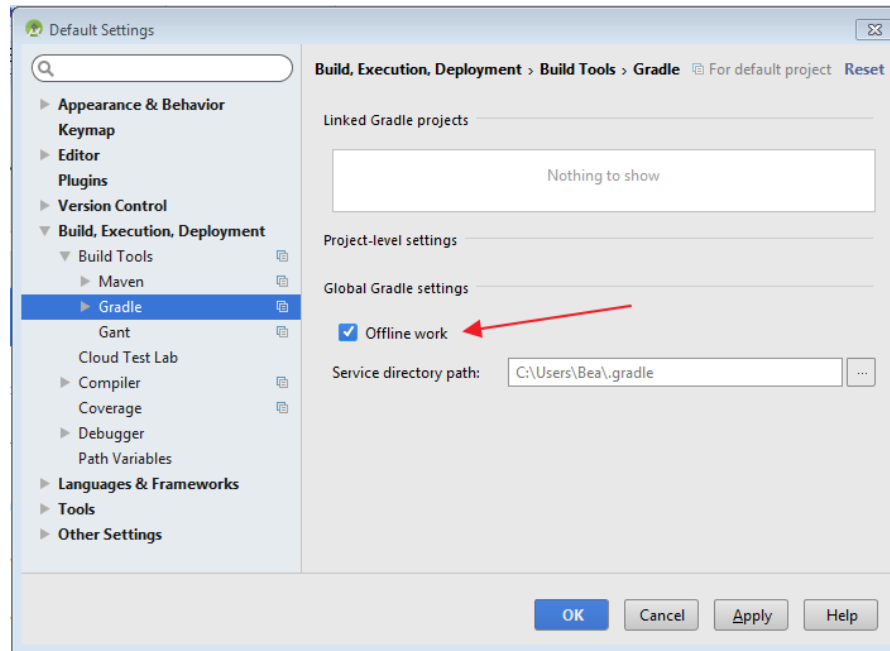
Acelerar "Build"



- Soluciones:
 - Aumentar la memoria RAM del equipo (Ja!)
 - Instalar un disco duro SSD (más ja!)
 - Asegurar que la virtualización está activada (no siempre se puede!)
 - Aumentar la memoria virtual del ordenador (Bueno!)
 - Activar ReadyBoost en un dispositivo de almacenamiento
 - Optimizar Gradle en Android Studio

Gradle “Offline”

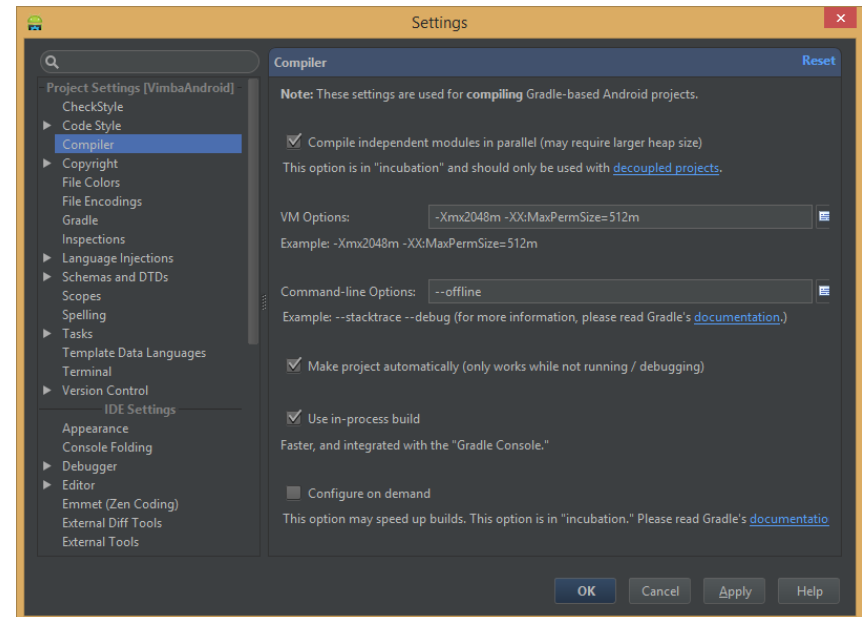
- **Ajustar las preferencias de gradle desde Android Studio para que trabaje "Offline"** (sin salir a la red a buscar las bibliotecas de las dependencias y que se utilicen las almacenadas en cache del ordenador).
- **NOTA:** Si la biblioteca no está en cache, el proyecto no se construirá!



Fijar propiedades Gradle



- Crea el archivo de texto llamado **gradle.properties** dentro de la carpeta **.gradle**, (que está en la carpeta del usuario, según corresponda cada sistema operativo, C:\Users\pepito en W o /home/pepito en L) con el siguiente contenido:
- **Utilizar Gradle como un proceso en segundo plano:**
`org.gradle.daemon=true`
NOTA: Si no construyes nada con Gradle en el lapso de 3 horas (puede variar), el servicio en segundo plano se pondrá inactivo, y tomará su tiempo volver a despertarlo.
- **Si el proyecto tiene varios módulos:**
`org.gradle.parallel=true`
- **Ajustar los parámetros de la máquina virtual de Java (memoria mínima y máxima). Ejemplos:**
`org.gradle.jvmargs=-Xms256m -Xmx1024m`
`org.gradle.jvmargs=-Xmx4096m -XX:MaxPermSize=1024m -XX:+HeapDumpOnOutOfMemoryError -Dfile.encoding=UTF-8`
O descomentar la línea del fichero gradle.properties del proyecto
- Enables new incubating mode that makes Gradle selective when configuring projects.
`org.gradle.configureondemand=true`
- Set to true or false to enable or disable the build cache. If this parameter is not set, the build cache is disabled by default.
`android.enableBuildCache=true`
- Algunas de estas propiedades también se pueden configurar desde AS



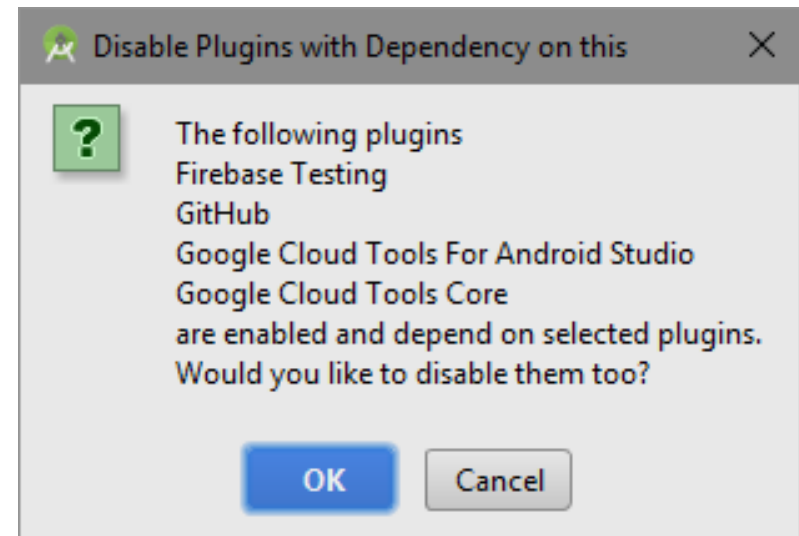
Fijar minSdk



- Fijar la minSdk más alta.
- Pero el inconveniente es que no sabremos si la aplicación funciona en dispositivos antiguos, ni si estamos utilizando los métodos compatibles.
- Solución:
 1. <https://gist.github.com/cesarferreira/8480ea6fd0b95ba57f98>
 2. Una vez comprobado el funcionamiento, recordar cambiar minSdk y construir al menos una vez

Desactivar plugins de CVS

- Settings->Plugins para desactivar:
 - CVS Integration
 - Git Integration
 - Subversion Integration



Compilación incremental



- Añadir en el archivo *Gradle* de la app y dentro de la sección de android lo siguiente:

```
dexOptions { incremental true }
```

Aumentar RAM para AS



- [Enlaces](#)

Usar la última versión de Gradle



- <https://services.gradle.org/distributions>
- Descargar, descomprimir y copiar en la carpeta gradle de la carpeta Android Studio.
- Hay que cambiar en cada proyecto, la carpeta desde Project Structure