

{desafío}
latam_

Publicación de aplicaciones y analítica _

Parte I



Gradle

¿Qué es Gradle?

Gradle es una herramienta de automatización de construcción (*Build automation tool*) diseñada para ser flexible en la construcción de cualquier tipo de *software*.

Ciclo de vida de la construcción con Gradle



Inicialización

Gradle soporta construcciones para uno o varios proyectos.

El archivo **settings.gradle** referencia a los proyectos que serán construidos.

Durante esta etapa:

- Determina qué proyectos van a tomar parte de la construcción
- Crea una instancia de la clase [Project](#) para cada uno de estos proyectos

Configuración

El script de construcción es ejecutado contra la instancia de Project previamente creada.

Ejecución

Se ejecutan en forma secuencial y paralela la lista de tareas configuradas.

Android Gradle Plugin

- Android Studio utiliza Gradle para construir sus proyectos
- El Android Gradle Plugin entrega funcionalidades específicas en la creación de proyectos Android
- Por convención, se utiliza el archivo **build.gradle**

build.gradle

En un proyecto Android hay 2 archivos **build.gradle** que son creados por defecto

- 1) El build.gradle de la raíz del proyecto es para agregar las configuraciones comunes a todos los módulos o subproyectos
- 2) El build.gradle dentro de cada subproyecto o módulo, que tiene la configuración particular

Tipos de compilación (Build Types)

Cada **variante de compilación representa una versión diferente de la app** que se puede compilar y surgen del uso de conjuntos de reglas específicas de Gradle que combinan configuración, códigos y recursos configurados.

Build Types

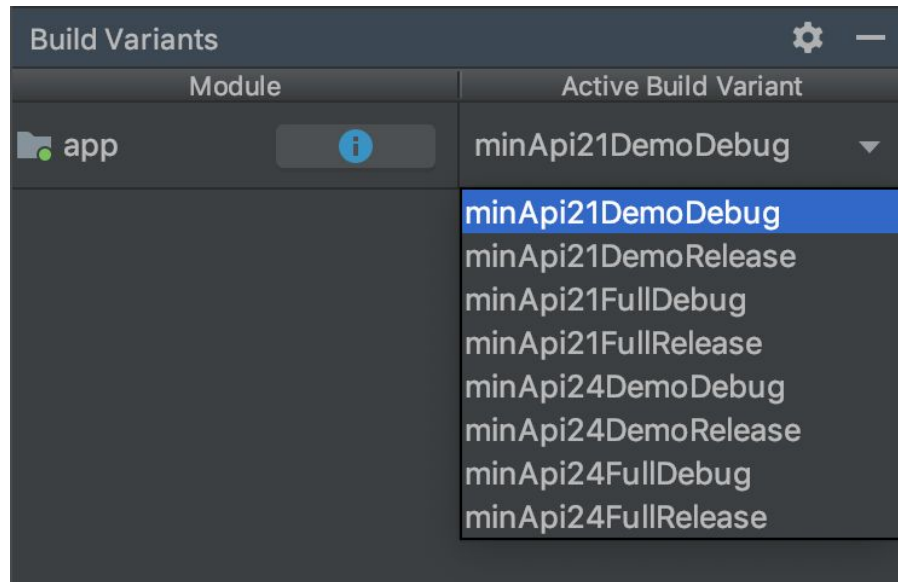
```
android {  
    compileSdkVersion 29  
    defaultConfig { ... }  
    buildTypes {  
        release {  
            minifyEnabled false  
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),  
            'proguard-rules.pro'  
        }  
  
        debug {  
            debuggable true  
        }  
    }  
}
```

Flavors

```
android {  
    ...  
    defaultConfig {...}  
    buildTypes {  
        debug{...}  
        release{...}  
    }  
  
    flavorDimensions "version"  
    productFlavors {  
        demo {  
            dimension "version"  
            applicationIdSuffix ".demo"  
            versionNameSuffix "-demo"  
        }  
        full {  
            dimension "version"  
            applicationIdSuffix ".full"  
            versionNameSuffix "-full"  
        }  
    }  
}
```

Build Variants

Es el producto cruzado entre los tipos de compilación y tipo de producto, y es la configuración que se utiliza para compilar el APK



Dependencias

build.gradle

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

android { ... }
dependencies {
    // Dependencia a módulo local llamado mylibrary
    implementation project(":mylibrary")

    // Dependencia de binarios alojados localmente en el directorio libs
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    // Dependencia a un binario remoto
    implementation 'cl.desafiolatam.android:cool-app:4.2'
}
```

Proguard

Es una herramienta de línea de comandos que encoge, optimiza y ofusca código.

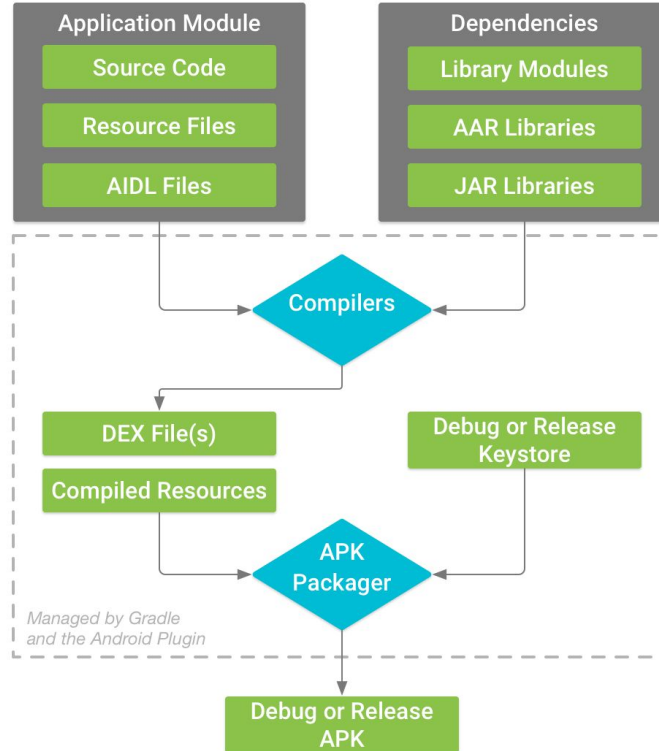
Con cada compilación, ProGuard crea los siguientes archivos:

- **dump.txt**: Describe la estructura interna de todos los archivos de clase del APK.
- **mapping.txt**: Proporciona una traducción entre la clase original y la oculta, el método y los nombres de campos. Se sobrescribe cada vez que se construye el proyecto
- **seeds.txt**: Indica las clases y los miembros que no se ocultaron.
- **usage.txt**: Indica el código que se quitó del APK.

Ofuscar código

```
androidx.appcompat.app.ActionBarDrawerToggle$DelegateProvider -> a.a.a.b:  
androidx.appcompat.app.AlertController -> androidx.appcompat.app.AlertController:  
    android.content.Context mContext -> a  
    int mListItemLayout -> 0  
    int mViewSpacingRight -> l  
    android.widget.Button mButtonNeutral -> w  
    int mMultiChoiceItemLayout -> M  
    boolean mShowTitle -> P  
    int mViewSpacingLeft -> j  
    int mButtonPanelSideLayout -> K
```


Proceso de compilación



Preparando la publicación de tu app

Pasos para publicar una app release

Lanzamiento

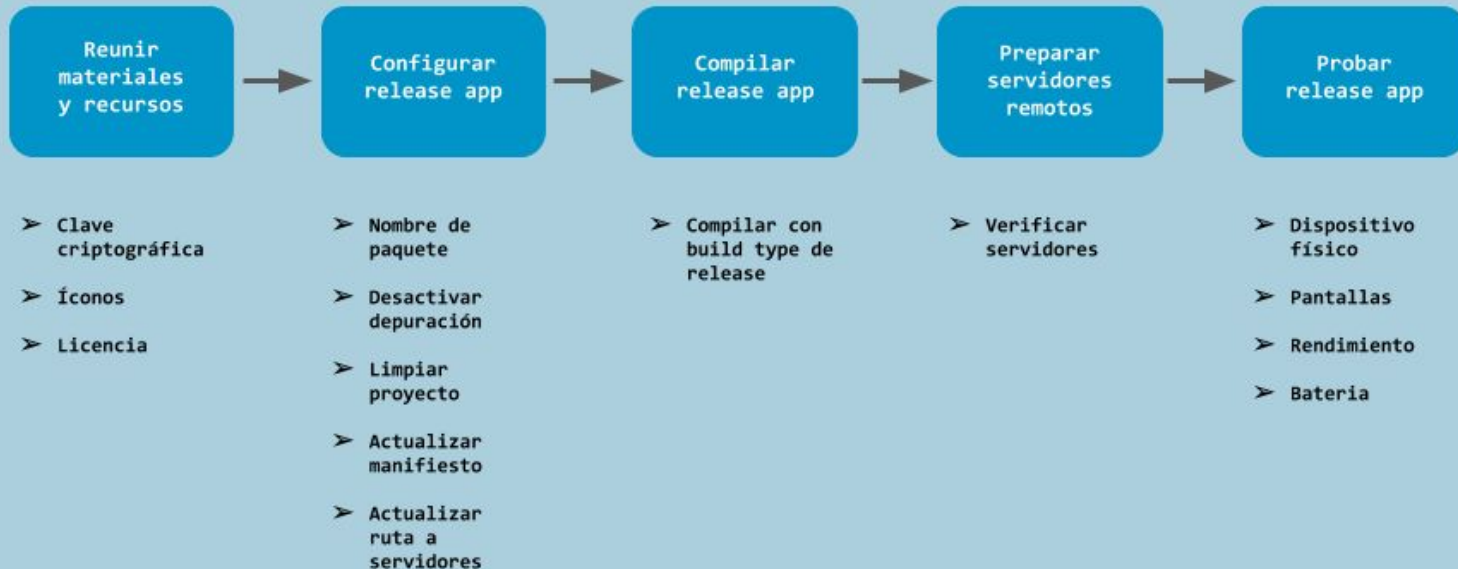
Preparar la app

Se configura,
construye y prueba
la app de release

Lanzamiento de la app

Se publicita,
vende y distribuye
la app a los
usuarios

Tareas para previas al lanzamiento (Release)



Versionamiento de la app

El versionamiento de la app es un componente fundamental de la estrategia de actualización y mantenimiento de la app.

Versionamiento de la app

Version code

- Valor entero que se usa como número interno de versión
- El número más grande la versión mas nueva
- Normalmente comienza con valor 1 y se incrementa para cada versión

Version name

- Es un texto usado exclusivamente como el número de versión que se le muestra al usuario
- Se puede usar versionamiento semántico (<major>.<minor>.<patch>)

Nivel de API

- **minSdkVersion**: versión mínima de la plataforma Android en la cual la app podrá ejecutarse
- **targetSdkVersion**: nivel de API con el cual la app está diseñada para funcionar.

Firma de la app

- A. Firma manual usando certificado
- B. Firma de apps de Google Play

Firma manual

- Al firmar un APK, la herramienta de firma adjunta el certificado de clave pública
- El propietario del certificado guarda la clave privada correspondiente
- El certificado de clave pública sirve como una "huella digital" que asocia de manera exclusiva el APK con la clave privada

Firma manual

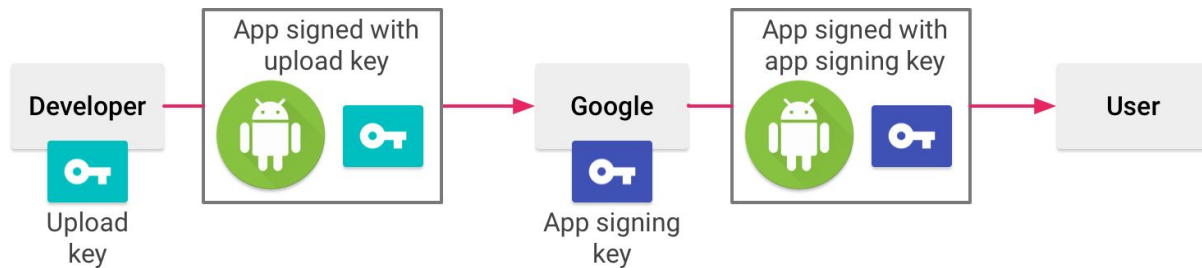
La pérdida de la clave privada significa que la app no puede seguir siendo actualizada en la tienda de aplicaciones.

Firma de app de Google Play

Para usar la firma de apps de Google Play se utilizan 2 claves:

- La firma de apps
- La firma de subida

Firma de app de Google Play



Formas de distribución



APK



Google Play

{desafío}
latam_

*Academia de
talentos digitales*

www.desafiolatam.com