

Crear y publicar librerías

Descargar estos apuntes [pdf](#) o [html](#)

Índice

- ▼ Crear y publicar librerías
 - ▼ Crear el proyecto de librería
 - Creación de repositorio y clonación en local
 - Creación del módulo de librería
 - Preparación del módulo para la publicación
 - ▼ Publicar la librería en JitPack
 - Crear la Release con la versión en GitHub
 - Repositorio de paquetes JitPack
 - ▼ Consumir la librería
 - Paso 1: En el fichero `settings.gradle.kts` añade...
 - Paso 2: Añade dependencia en `build.gradle.kts` del módulo de nuestra app

Crear el proyecto de librería

Creación de repositorio y clonación en local

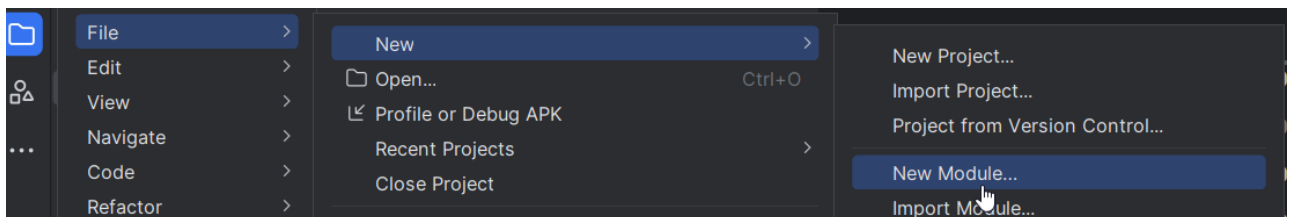
Lo primero que tendremos que hacer, es crear un repositorio público en GitHub en el que alojaremos nuestro proyecto. Los conocimientos para crear un repositorio en GitHub no corresponden a este módulo, por lo que omitiremos estos pasos que pueden encontrarse en la web fácilmente. Una vez creado el repositorio, tendremos que clonarlo en una carpeta local en donde alojaremos nuestro proyecto.

Crearemos un proyecto directamente en esta carpeta, **tendremos que tener un repositorio por módulo de librería**.

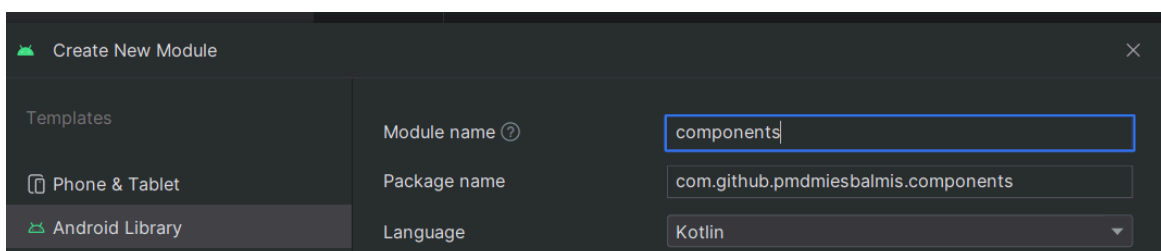
Creación del módulo de librería

Estos proyectos constarán de **dos** módulos:

1. El **módulo de aplicación**, que es el que se crea por defecto cuando creamos un proyecto de Android, en este alojaremos el código necesario para realizar un test de nuestra librería.
2. El **módulo de librería**, que será el que aloje el código de nuestra librería. Para crear un módulo de librería deberemos seguir los siguiente pasos:
 - Crearemos un **Nuevo Módulo**

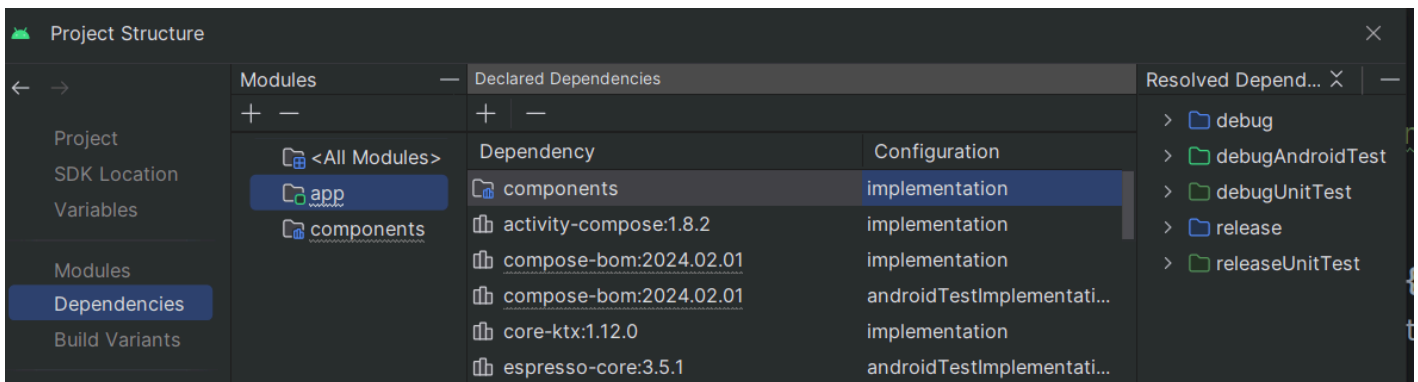


- Indicaremos que el módulo debe de ser de **librería**, añadiéndole el nombre al módulo y el nombre del paquete que debe de coincidir con el alojamiento de GitHub. En ejemplo de la imagen la librería se llamará **components**.



Una vez creado el módulo de librería, tendremos que añadir la dependencia entre el módulo de aplicación y el de librería, para ello deberemos acceder a la ventana de

Project Structure -> Dependencies y sobre app añadir una dependencia **(+)** al módulo creado.

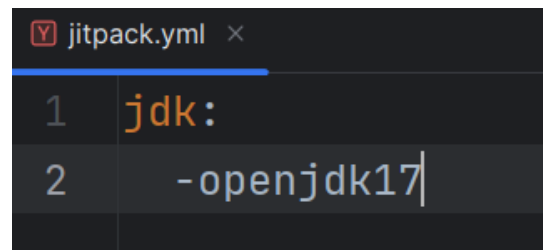
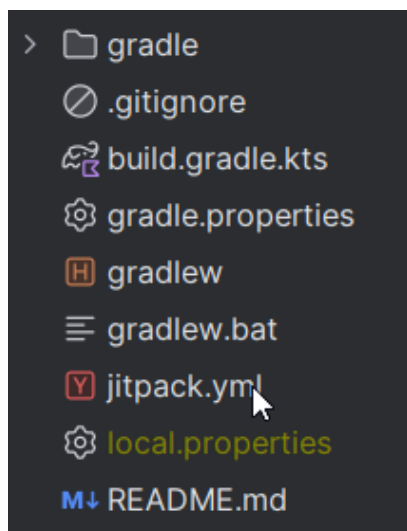


Una vez añadida la dependencia, podremos utilizar las clases de módulo de librería en la app y probar el funcionamiento correcto de esta.

Preparación del módulo para la publicación

Los siguientes pasos a realizar serán los de preparar la librería para que pueda ser publicada, para eso deberemos de añadir...

- El archivo **jitpack.yml** en la misma ubicación que **build.gradle.kts** de la librería y donde indicamos el jdk que se usará para compilar la librería en JitPack como se muestra en la imagen.



- Añadiremos las siguientes líneas de código en el archivo `build.gradle.kts`, que permitirá que sea publicado en el repositorio Maven.

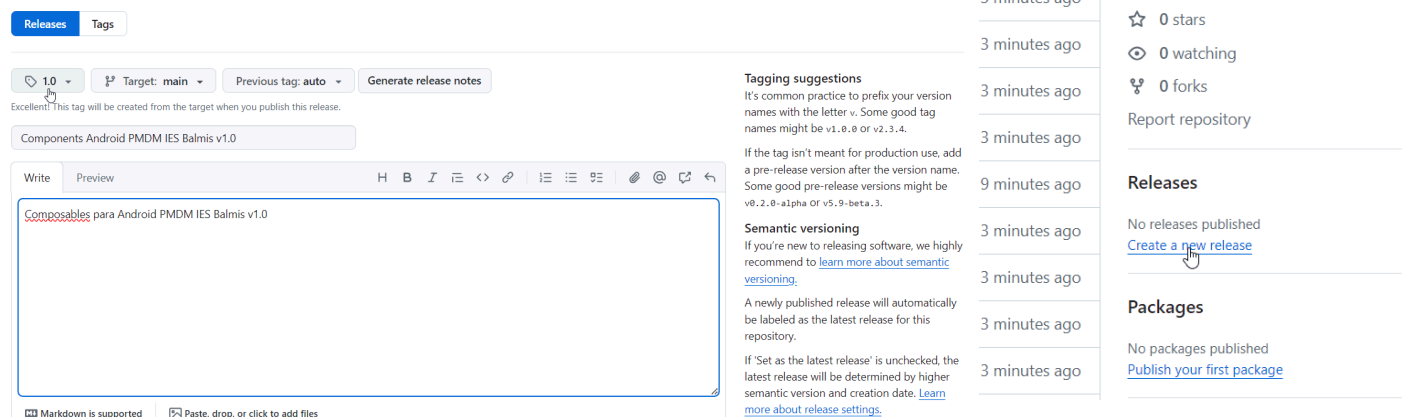
```
plugins {  
    ...  
    3 id("maven-publish")  
}  
  
...  
  
8 // Al final del archivo y fuera de cualquier bloque  
publishing {  
    publications {  
        register<MavenPublication>("release") {  
  
13 // Observa que se deberá poner la ruta de github al repositorio  
        groupId = "com.github.pmdmiesbalmis"  
  
16 // Nombre exacto del repositorio y de la versión  
        artifactId = "components"  
        version = "1.0"  
  
        afterEvaluate {  
            from(components["release"])  
        }  
    }  
}  
}
```

Una vez tengamos una versión definitiva de la librería, deberemos actualizar la rama `main` en GitHub para poder publicarla.

Publicar la librería en JitPack

Crear la Release con la versión en GitHub

Ahora desde GitHub y en el repositorio que queremos publicar, vamos a realizar la primera **'Release'**. Para ello, pulsaremos sobre el enlace habilitado en la parte derecha del repositorio, este enlace nos llevará a la pantalla en la que podremos crear el Tag con la versión de nuestra release . Además añadiremos la descripción y un comentario sobre la librería.

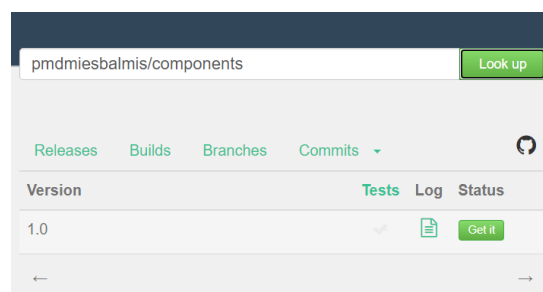


Nota: Para crear el Tag deberemos seguir los convenios específicos que vienen explicados en la misma página.

Repositorio de paquetes JitPack

En el siguiente paso vamos a enlazar la versión del **'Release'** con el repositorio de paquetes gratuito y de fácil uso **JitPack**, lo primero que tendremos que hacer es iniciar sesión con la cuenta de GitHub en **JitPack**, una vez iniciada deberemos buscar el repositorio con el que queremos enlazar indicando la ubicación en **Look Up** .

JitPack detectará la versión y procederá a construir el artefacto, después de algunos minutos y si todo ha ido de forma correcta el Log no mostrará ningún error y el proceso habrá terminado.



Consumir la librería

En cuanto a la parte de consumir la librería por terceros, es muy sencillo ya que solamente habrá que incluir el código que se especifica en la página de JitPack cuando se crea el paquete.

How to

To get a Git project into your build:

Step 1. Add the JitPack repository to your build file

gradle

maven

sbt

leiningen

Add it in your root build.gradle at the end of repositories:

```
dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        mavenCentral()
        maven { url 'https://jitpack.io' }
    }
}
```

Step 2. Add the dependency

```
dependencies {
    implementation 'com.github.pmdmiesbalmis:components:Tag'
}
```

Es decir, deberemos añadir lo siguiente:

Paso 1: En el fichero `settings.gradle.kts` añade...

```
dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
        // AÑADIR -----
        maven(url = "https://jitpack.io")
        // -----
    }
}
```

Paso 2: Añade dependencia en `build.gradle.kts` del módulo de nuestra app

```
dependencies {  
    // Dependencia referida a la librería y versión que queremos acceder  
    implementation("com.github.pmdmiesbalmis:components:1.0")  
}
```