

## ...: Configurar OpenCV en Android Studio ...:



==:: By Surflaweb ==::

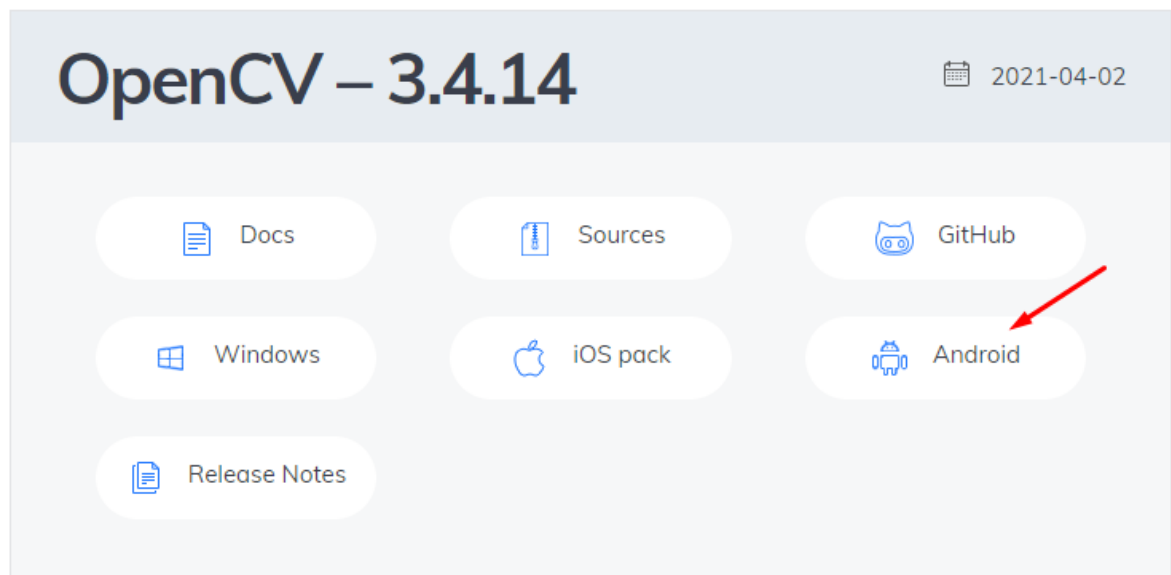
:: Actualizado 8 de agosto de 2021 :::

Aclaración:

*Se configuró OpenCV a la versión 4.2.2 de Android Studio.*

Video Tutorial: <https://youtu.be/ZUhoxBI6qmQ>

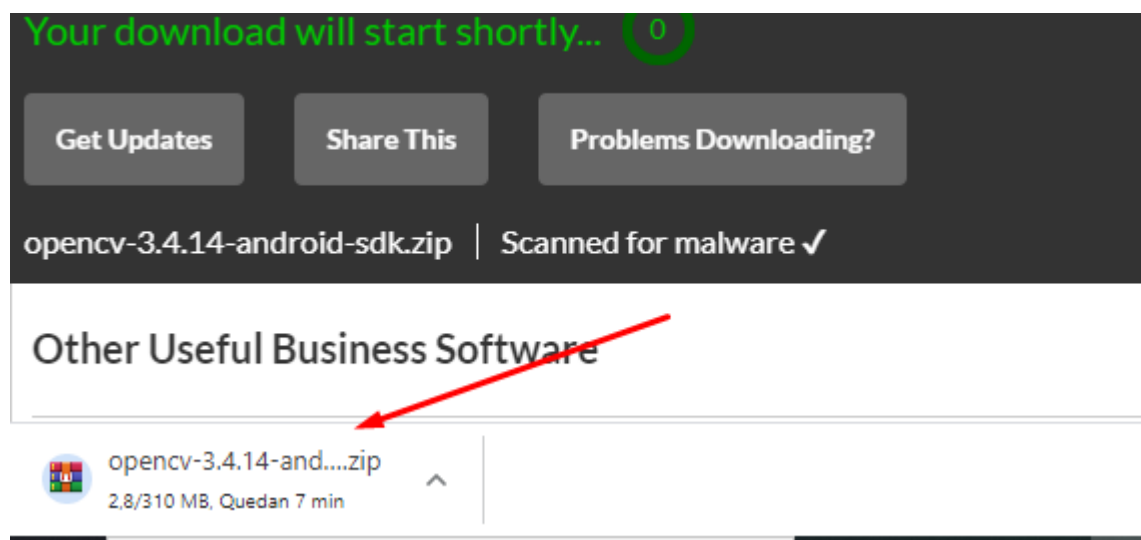
1. Ir a la página <https://opencv.org/releases/> y descargar la versión de OpenCV para Android:
  - Elegir OpenCV – 3.4.13 para Android



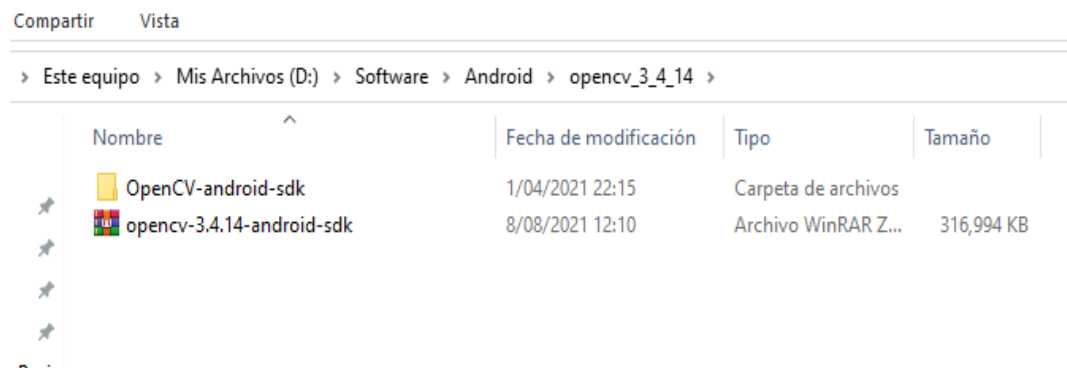
Hacer click en “Android” esta opción nos llevará a la página de sourceforge

<https://sourceforge.net/projects/opencvlibrary/files/3.4.14/opencv-3.4.14-android-sdk.zip/download>

y se iniciará la descarga:

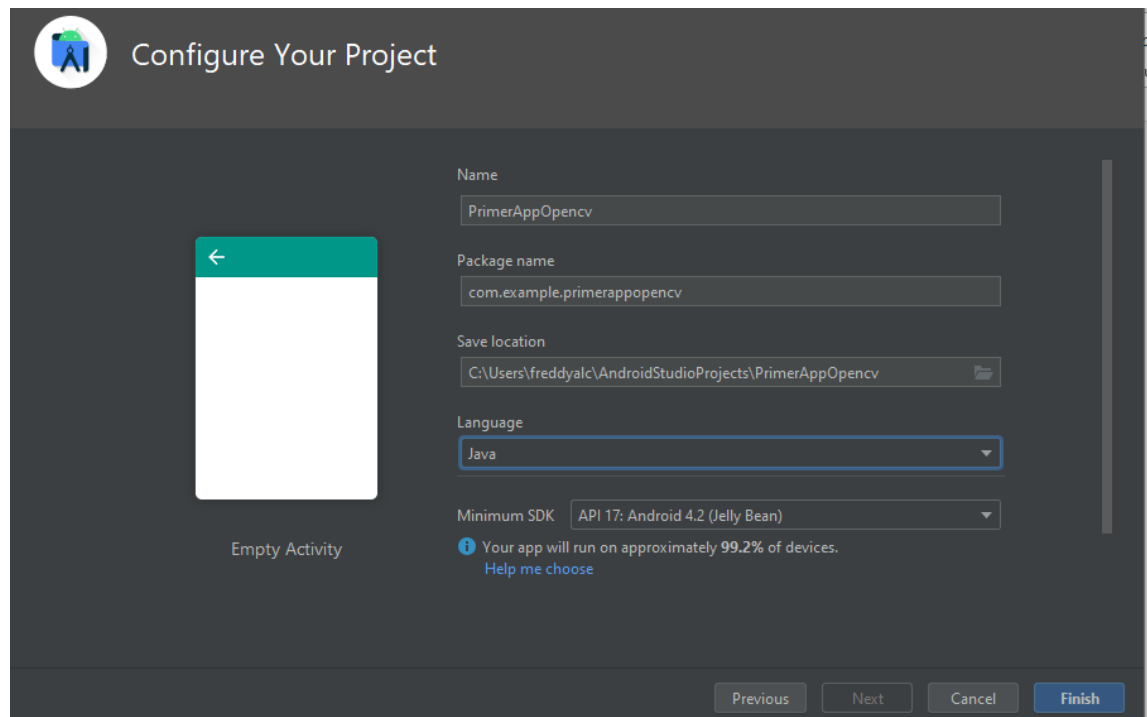


## 2. Poner el archivo descargado en una carpeta y descomprimirlo



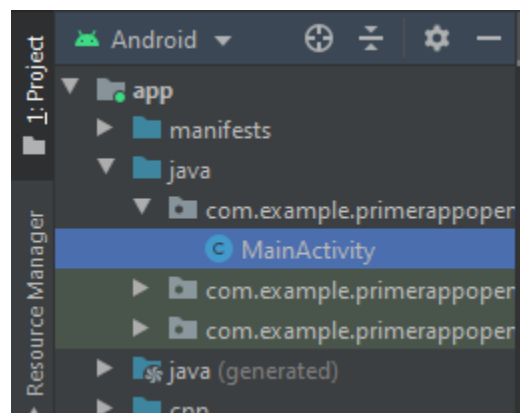
## 3. Crear un proyecto en Android Studio

Nota: Elegir el lenguaje java.

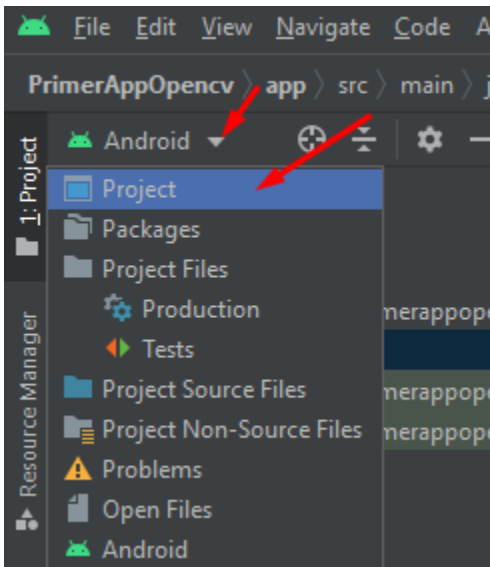


#### 4. Modificar el archivo gradle

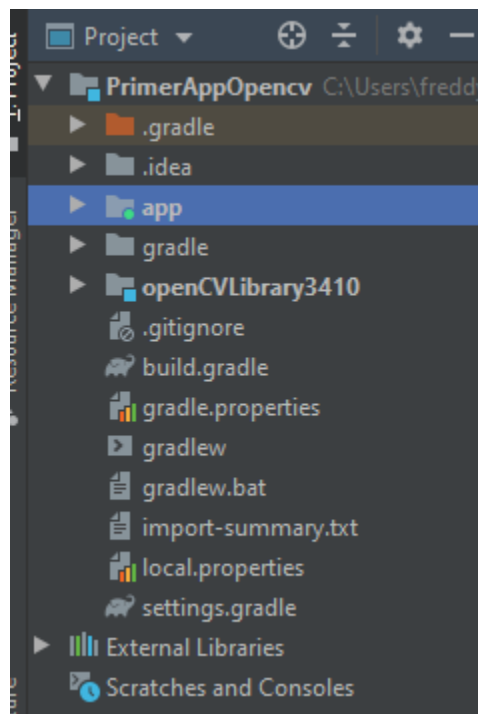
Antes agregar/modificar el gradle tenemos que cambiar la vista de nuestro proyecto, por defecto esta así:



Cambiamos a la vista "Project":

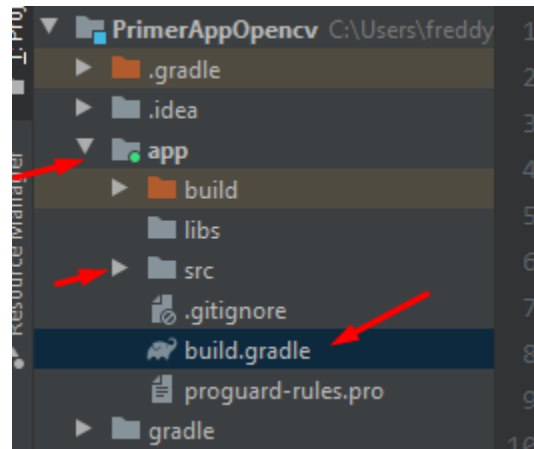


Y ahora la estructura del proyecto se mostrará así:



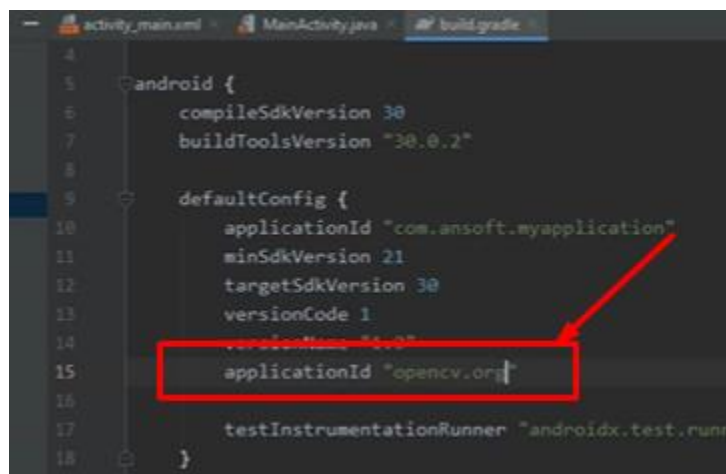
No se preocupen si ven archivos en esta imagen que ustedes no tienen, es por que ya agregué opencv. Luego al final volveremos a cambiar la vista a como estaba antes eligiendo “Android”.

Ahora si nos vamos al archivo gradle:



Agregar dentro de 'defaultConfig':

- applicationId "opencv.org"



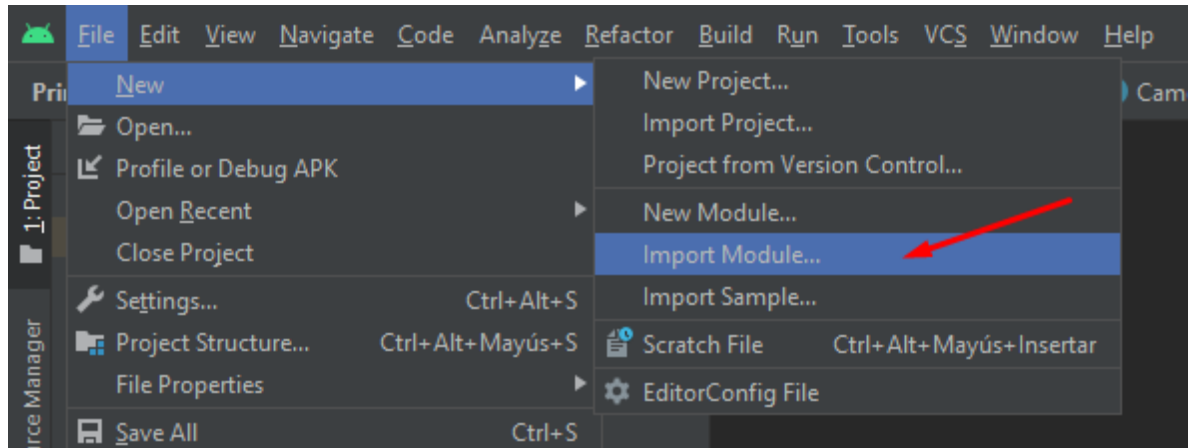
**También debemos de agregar la dependencia:** testImplementation 'junit:junit:4.13.2'  
o mayor pero no 4+ si no especificando la versión.

```
implementation 'androidx.appcompat:appcompat:1.3.1'  
implementation 'com.google.android.material:material:1.4.0'  
implementation 'androidx.constraintlayout:constraintlayout:2.1.0'  
testImplementation 'junit:junit:4.13.2'  
androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
```

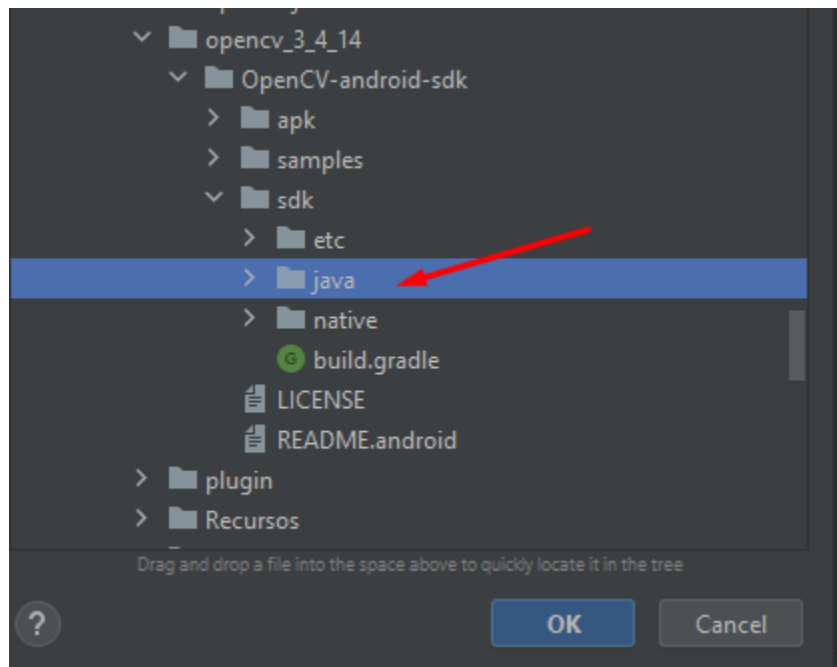
Luego sincronizar el gradle.

## 5. Agregar módulo OpenCV


Ir a File->New->Import Module



Ahora tenemos que elegir la carpeta 'java' que esta dentro del sdk del OpenCV descargado:



Presionamos en next y finish.




## Import Module from Source

Gradle or Eclipse project

Source directory:

Module name:

Previous Next Cancel Finish



## ADT Import Preferences

The ADT project importer can identify some .jar files and even whole source copies of libraries, and replace them with Gradle dependencies. However, it cannot figure out which exact version of the library to use, so it will use the latest. If your project needs to be adjusted to compile with the latest library, you can either import the project again and disable the following options, or better yet, update your project.

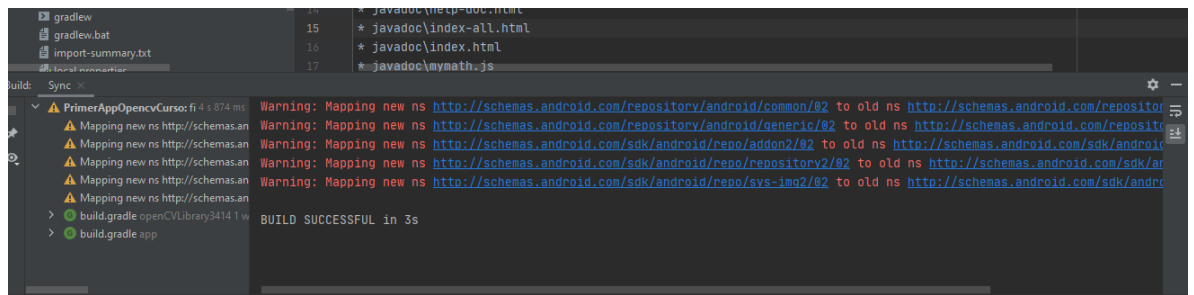
- ☒ Replace jars with dependencies, when possible
- ☒ Replace library sources with dependencies, when possible

Other Import Options:

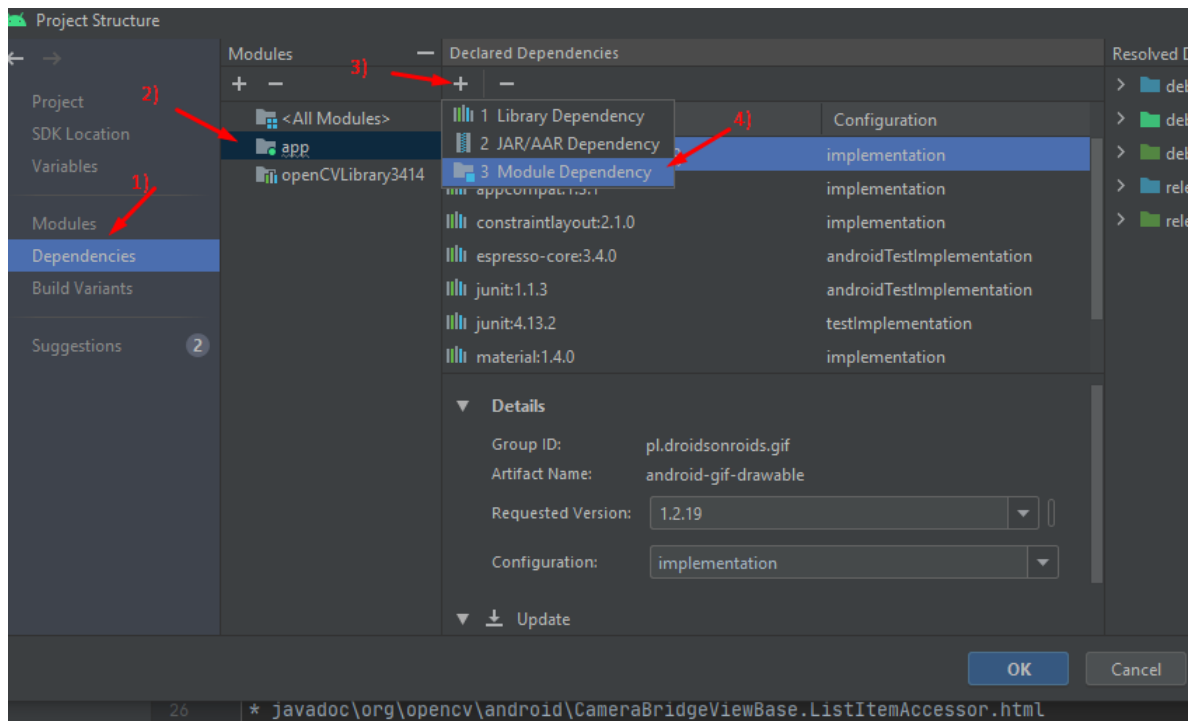
- ☒ Create Gradle-style (camelCase) module names

Previous Next Cancel Finish

Luego en la pestaña de Build aparecen unos warnings sin embargo no nos preocupamos:

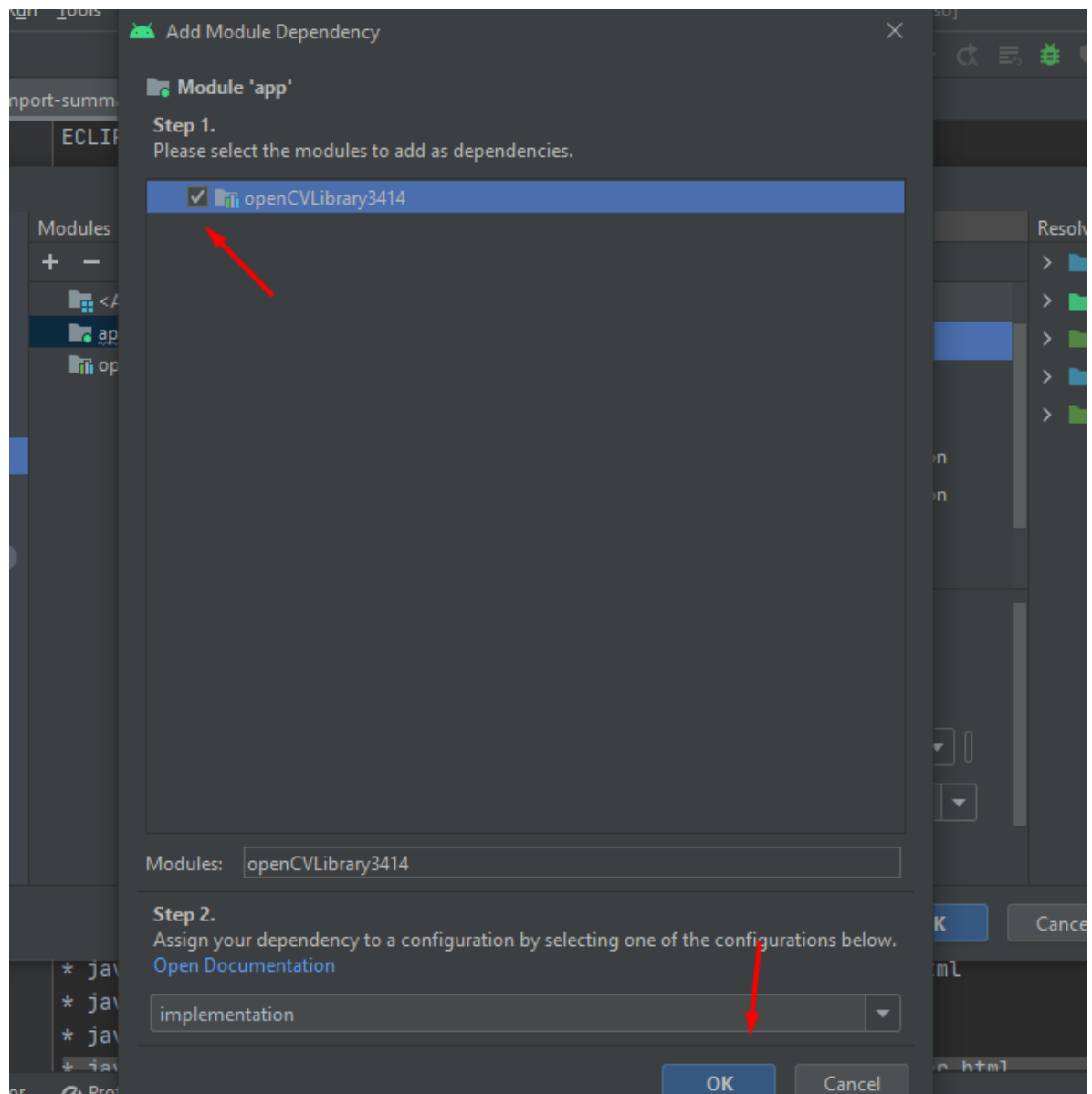


Luego nos vamos a 'File->Project Structure->Dependencies-> Click en App-> Luego click en el icono + y seleccionar "3 Module Dependency"



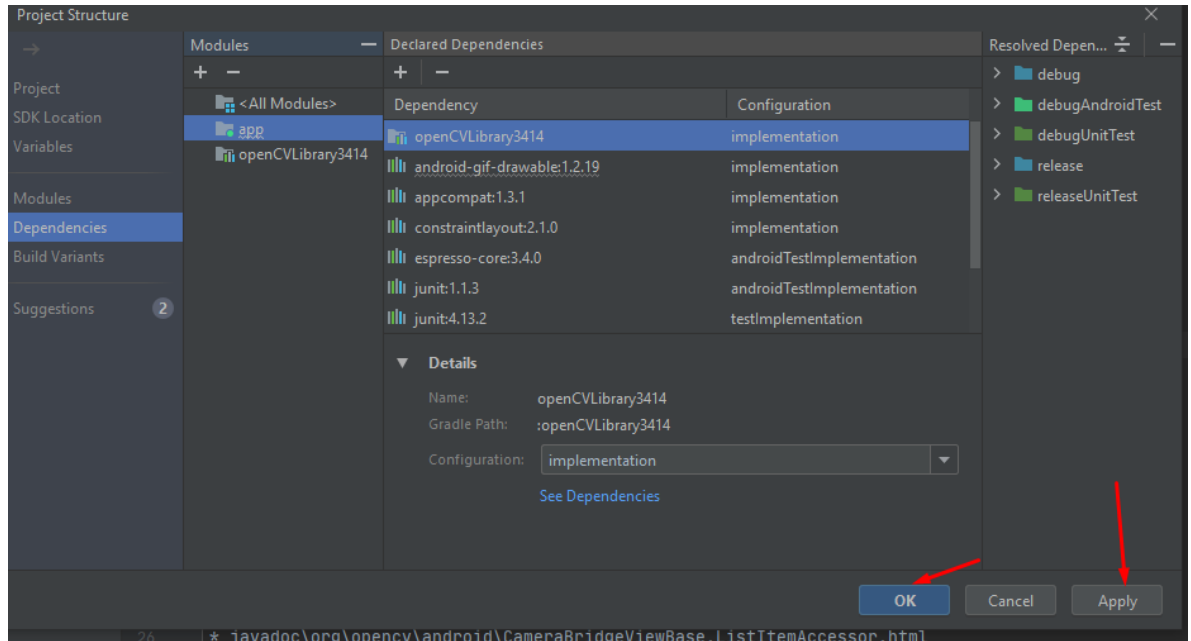
Aquí agregamos OpenCV:





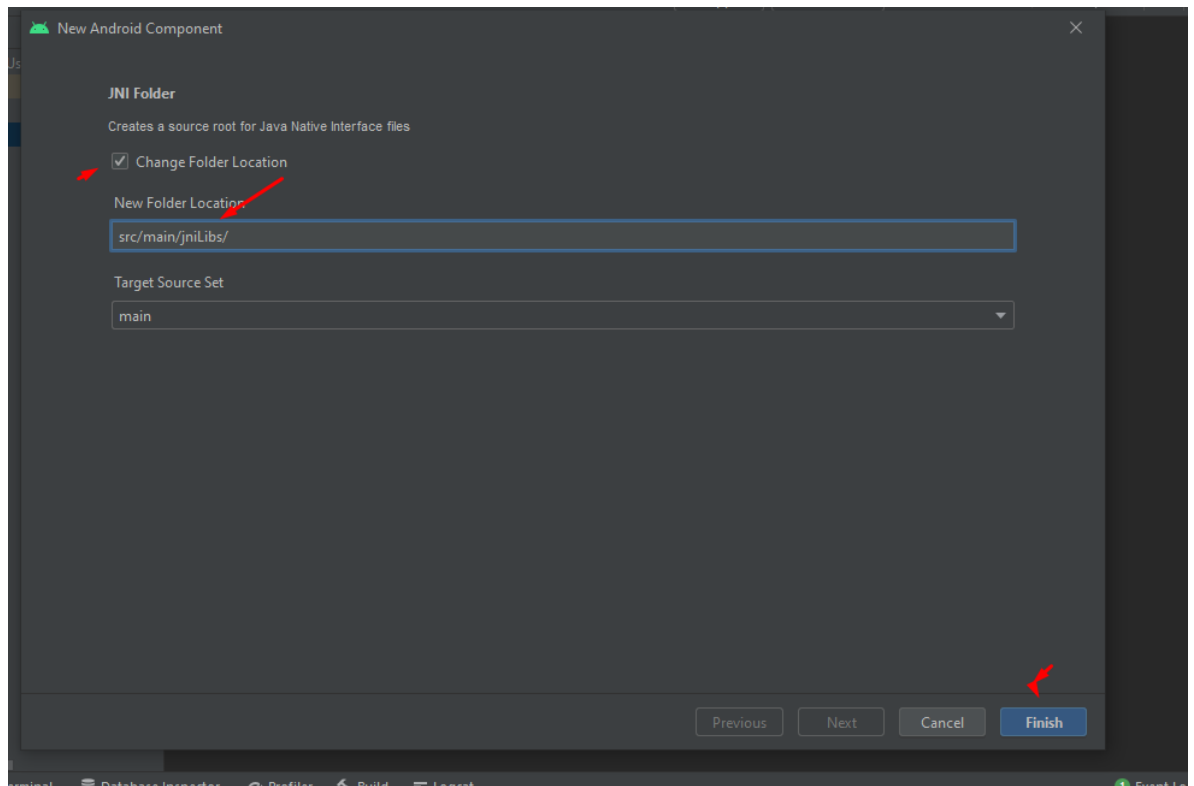
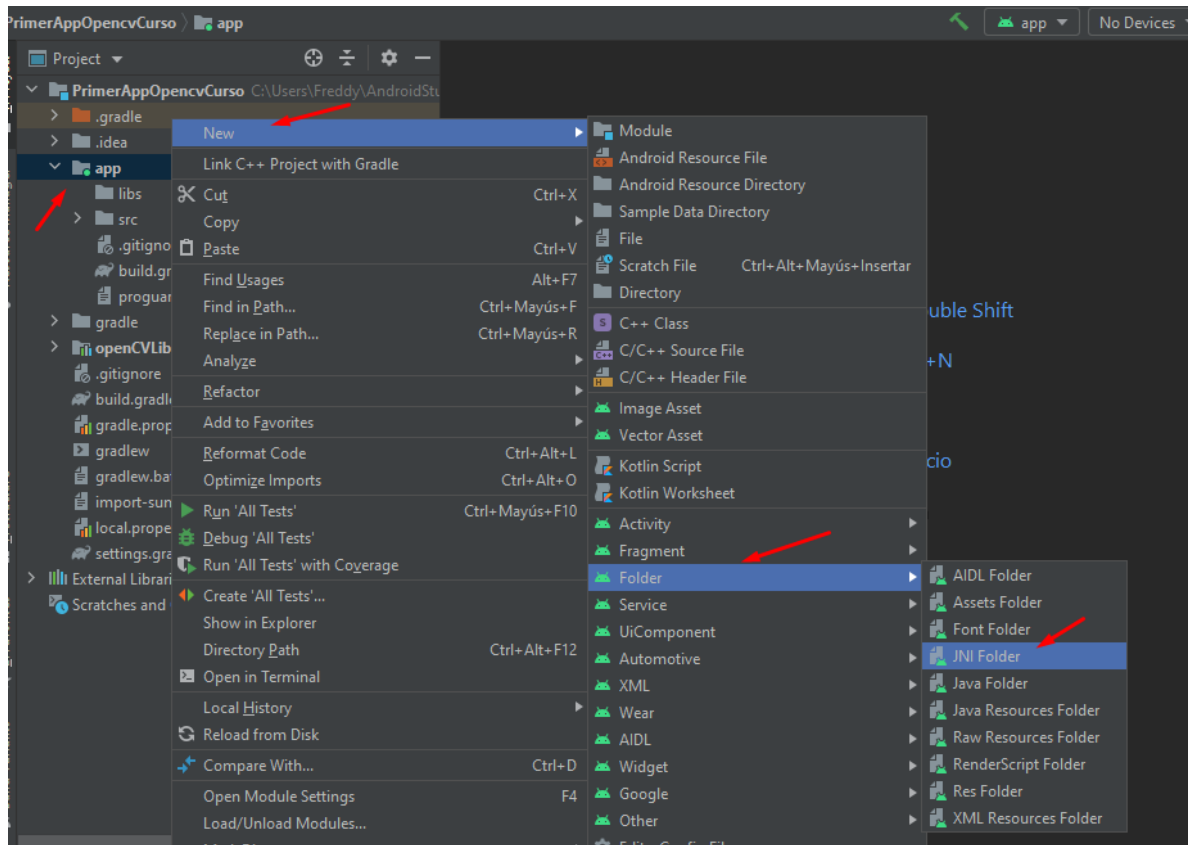
Seleccionar 'OpenCV' y ok.

**Luego Apply y Ok:**



**Luego creamos el folder “JNI” y lo renombramos a “JniLibs”:**

Click en la carpeta App-> Click derecho->New->Folder->JNI Folder

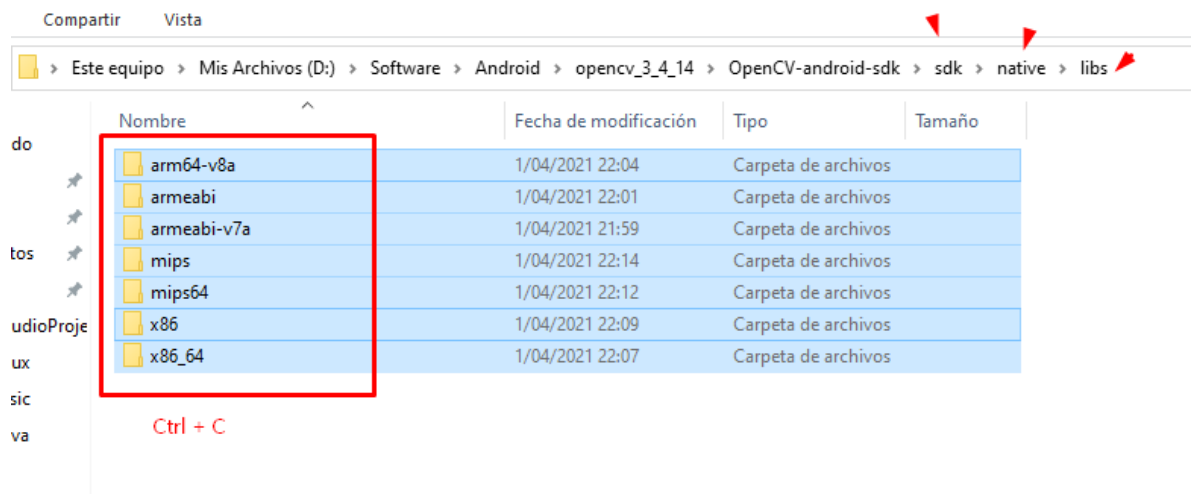


**“Clicke en Finish”**

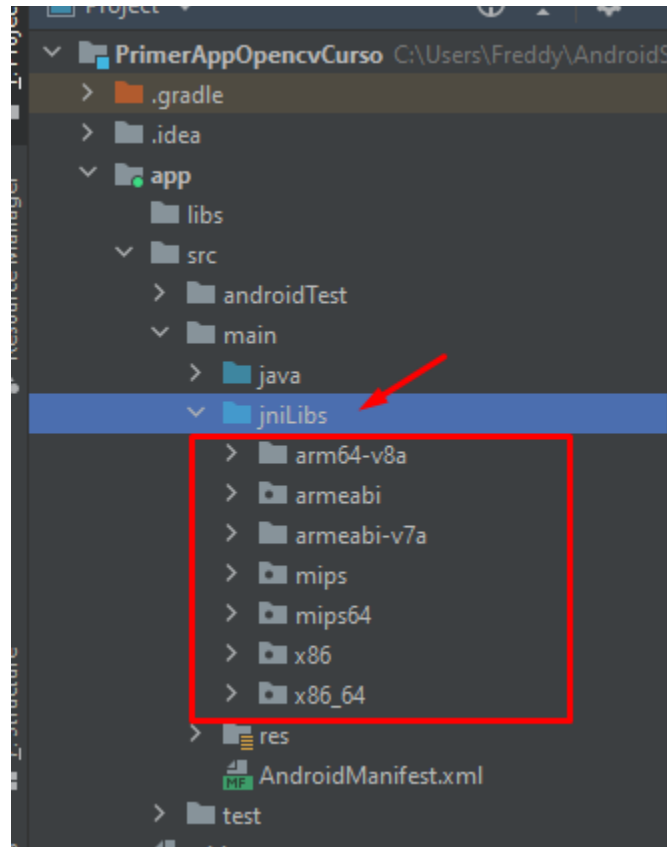
**Ahora vamos a copiar los siguientes archivos de la carpeta:**

**OpenCV-android-sdk->sdk>Native->Libs:**

Copiar todas las carpetas:



**Pegar dentro de “App->src->main->jniLibs”:**



Ahora OpenCV ya se ha agregado exitosamente. Solo resta probar.

## 6. Probando

Agregar el siguiente código antes del método `onCreate()` del `MainActivity.java`, esto es para verificar si OpenCV esta cargado correctamente.. Sin embargo a veces puede dar false y tranquilamente podemos trabajar.. "`OpenCVLoader.initDebug()`" no debe de estar en rojo ya que indicaría que no instalamos correctamente el modulo de OPENCV.

```
private static String TAG = "MainActivity";
static {
    if(OpenCVLoader.initDebug()){
        Log.d(TAG, "OpenCV instalado exitosamente.");
    }else{
        Log.d(TAG, "OpenCV no se instalo Error..");
    }
}
```

```

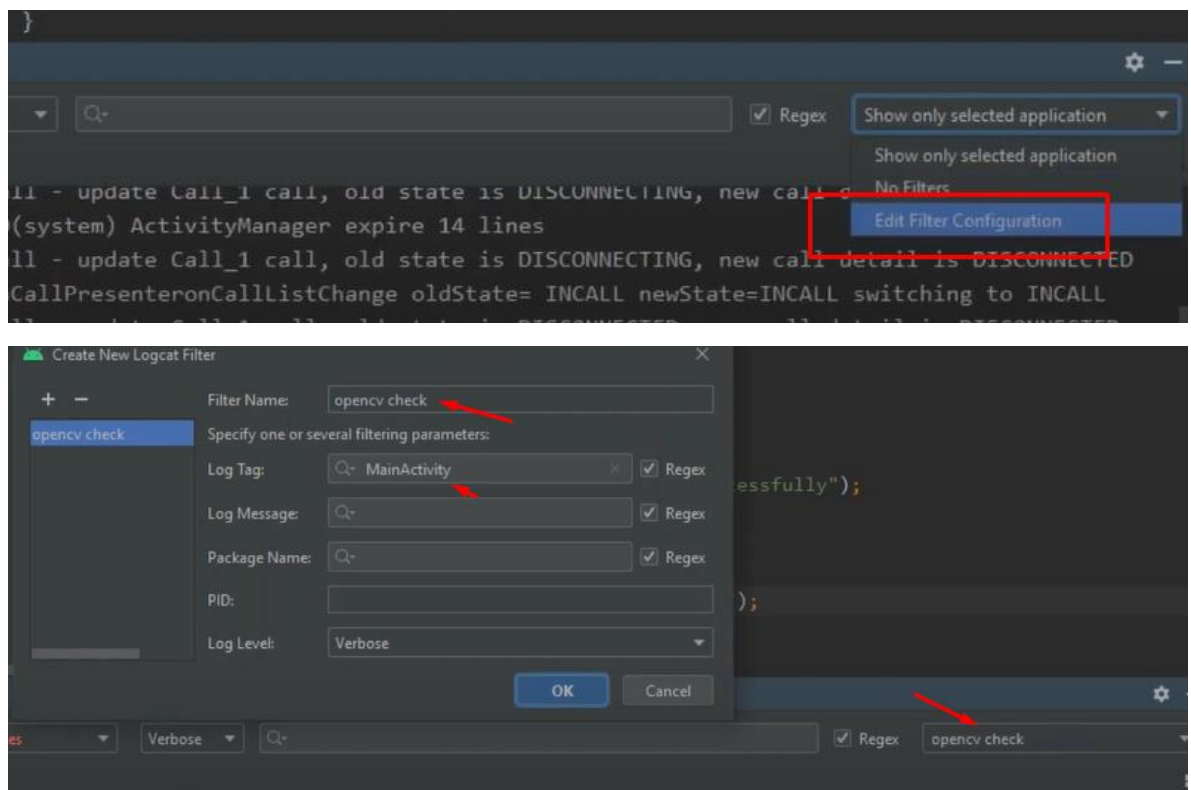
10 public class MainActivity extends AppCompatActivity {
11     private static String TAG = "MainActivity";
12     static {
13         if(OpenCVLoader.initDebug()){
14             Log.d(TAG, msg: "OpenCV instalado exitosamente.");
15         }else{
16             Log.d(TAG, msg: "OpenCV no se instalo Error..");
17         }
18     }
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main);
23     }
24 }

```

Compilar el proyecto en un emulador o teléfono.

**Opcional:**

**Crear Filtro para encontrar el Log más rápido:**



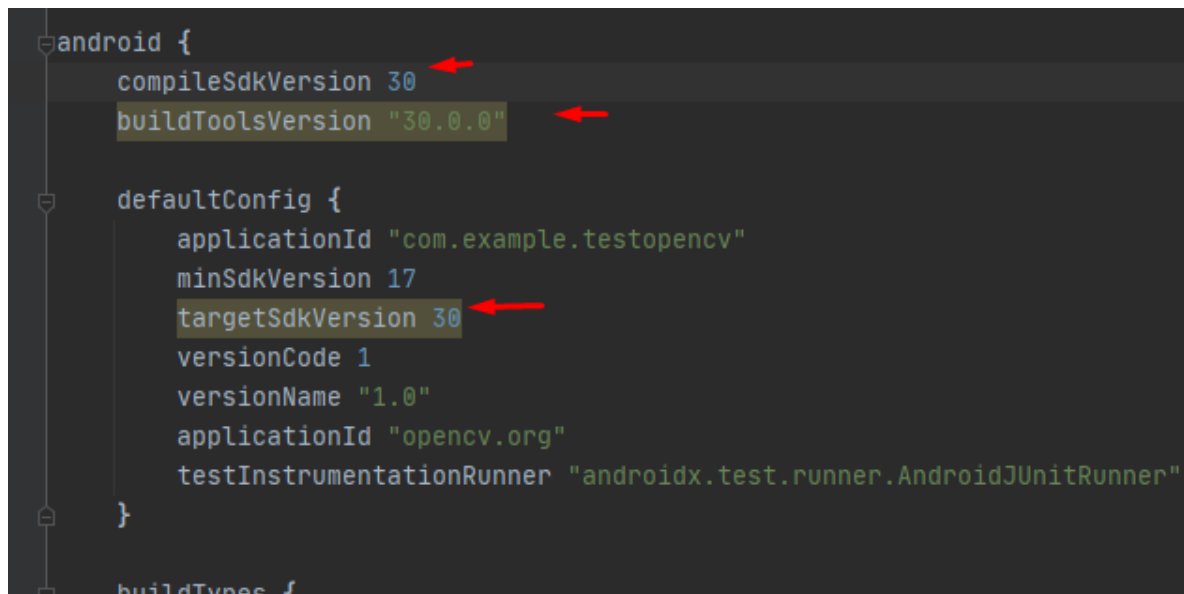
**Solucionando el siguiente error si les da a ustedes:**

Conectamos nuestro teléfono y compilamos:

Si les sale este error:

**Installed Build Tools revision 31.0.0 is corrupted. Remove and install again using the SDK Manager.**

Cambia el `targetSdkVersion`, `compileSdkVersion` y `BuildTools` a la versión 30.0.0, a mi sale xq desinstale Android studio artic fox e instale Android studio 4.2.2 el cual es la versión estable, no se si deba a eso. Pero si ha ustedes no les sale ese mensaje de que Build Tools esta corrupta no cambie la versión y déjenlo al más actual en este caso es la 31.0.0



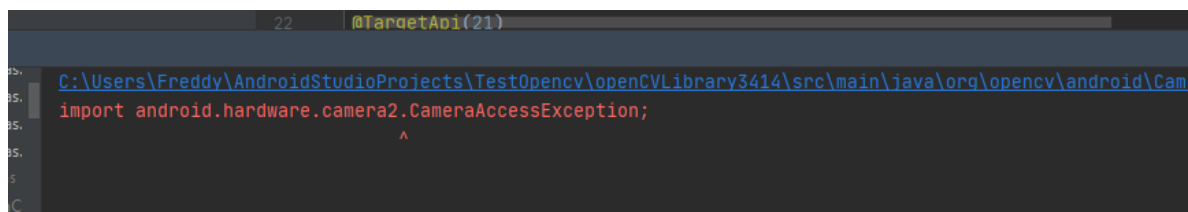
The screenshot shows the `android` block in `AndroidManifest.xml`. Red arrows point to the following values:

- `compileSdkVersion 30`
- `buildToolsVersion "30.0.0"`
- `targetSdkVersion 30`

```
android {  
    compileSdkVersion 30  
    buildToolsVersion "30.0.0"  
  
    defaultConfig {  
        applicationId "com.example.testopencv"  
        minSdkVersion 17  
        targetSdkVersion 30  
        versionCode 1  
        versionName "1.0"  
        applicationId "opencv.org"  
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
    }  
}
```

¡Sincronizamos otra vez y compilamos!

Oh! Sorpresa otro error en esta versión de OpenCV 3:

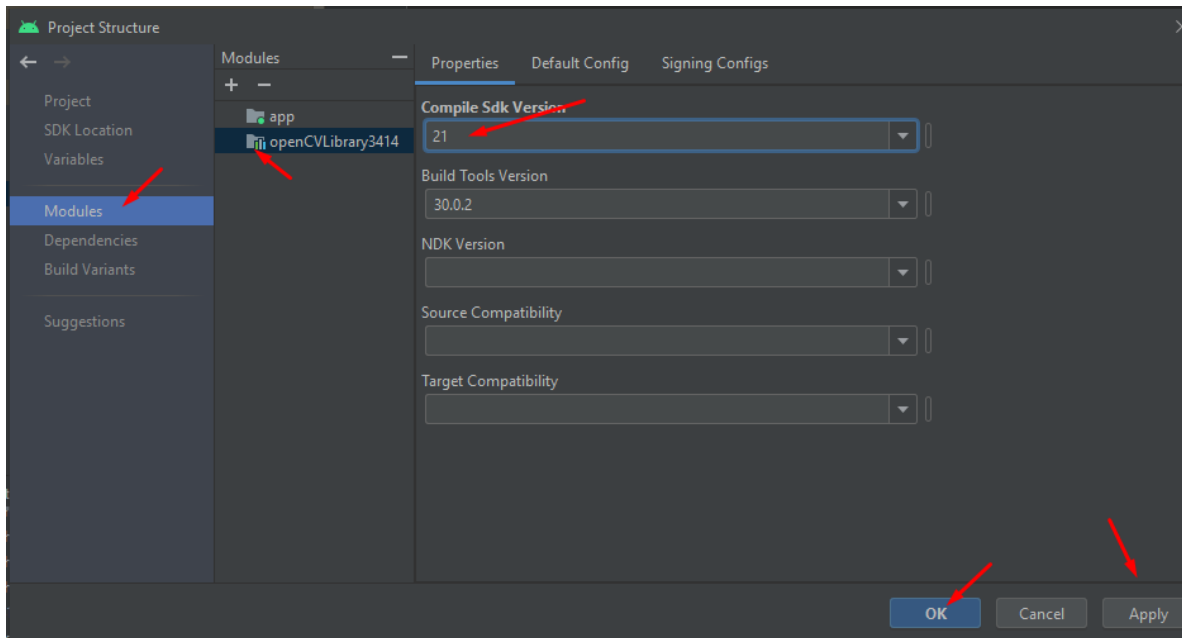


The screenshot shows a Java file in an IDE. The top bar indicates the target API level is 21. The code shows an import statement for `android.hardware.camera2.CameraAccessException`. A red squiggly line under the package name indicates an error.

```
22 | @TargetApi(21)  
23 |  
24 | C:\Users\Freddy\AndroidStudioProjects\TestOpencv\openCVLibrary3414\src\main\java\org\opencv\android\Cam  
25 | import android.hardware.camera2.CameraAccessException;  
26 |  
27 |  
28 |  
29 |
```

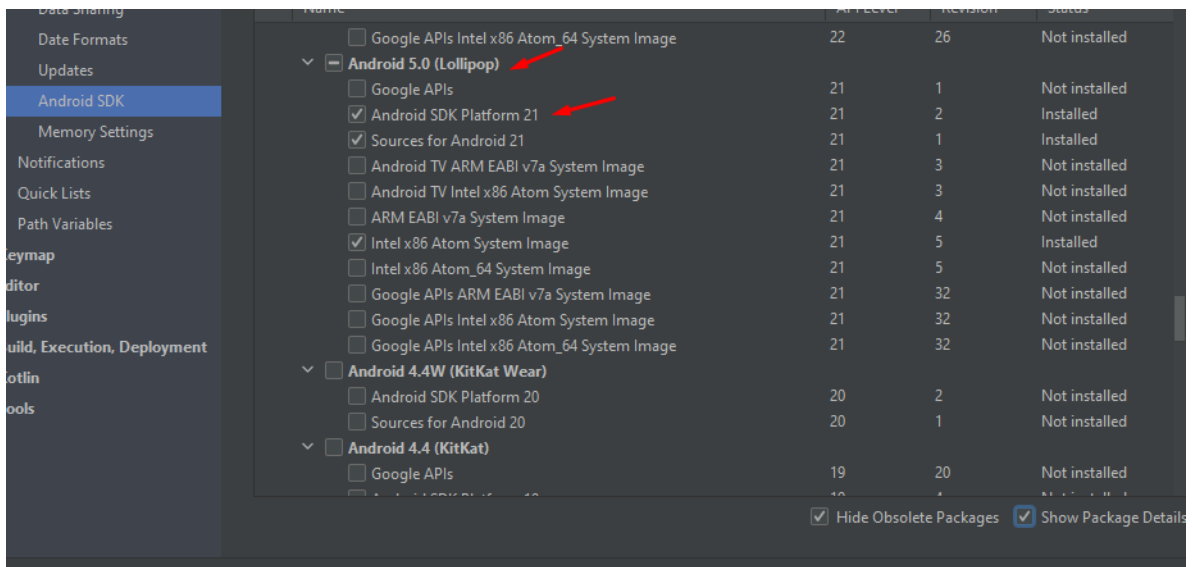
**error: package android.hardware.camera2 does not exist import android.hardware.camera2.CameraAccessException;**

La solución es ir a File->Project Structure -> Modules -> seleccionar `OpencvLibrary3414` y cambiar el `compilSdkVersion` de 14 a 21 ó Android 5.0



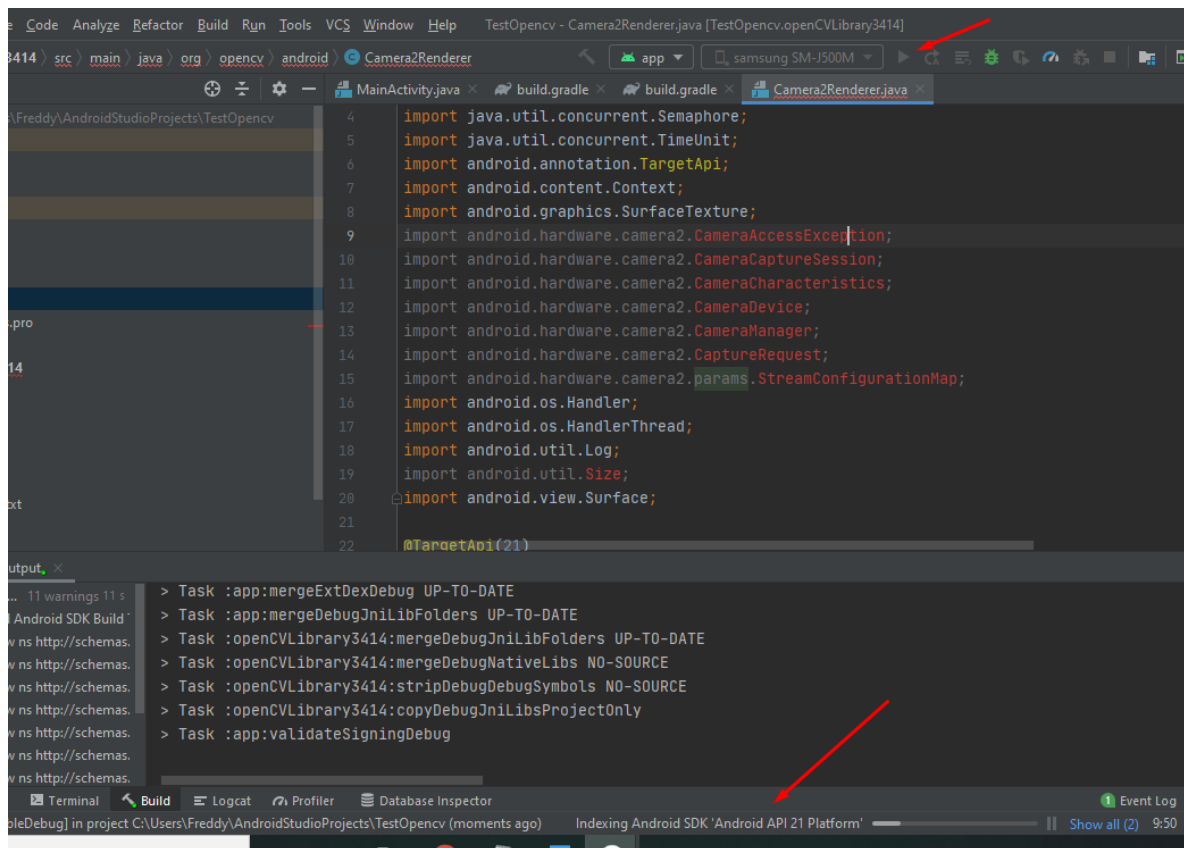
Esperamos que se indexe y luego volvemos a compilar y los errores desaparecerán. O bien hacemos File-> Invalidate Cache/Restart

Ojo: Desde el SDK manager deben de tener instalado Android 5.0 para solucionar esto.

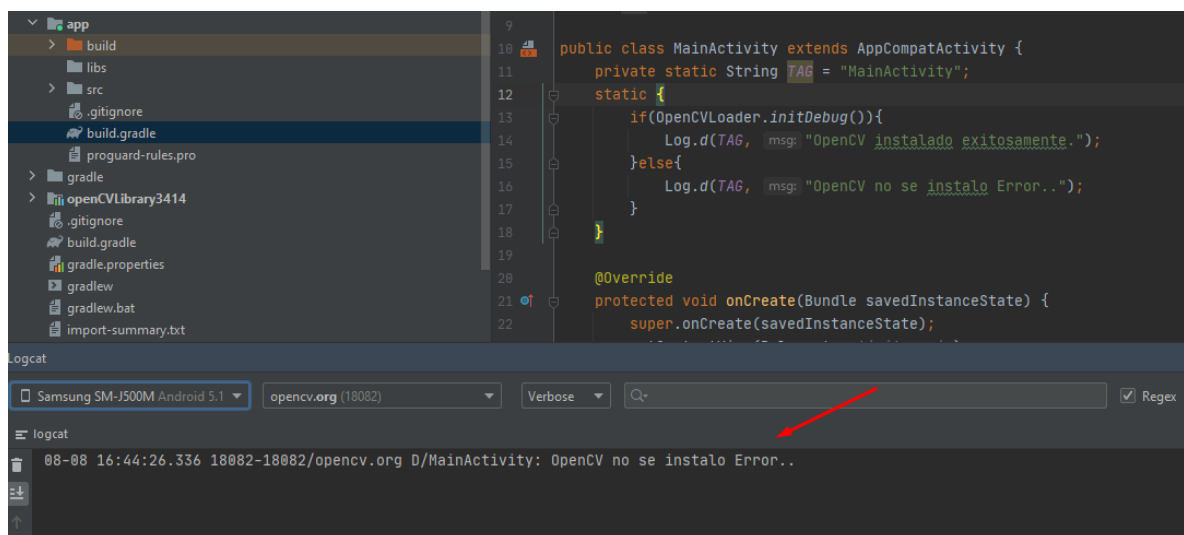




## Conpilamos!



Oh Sorpresa! No obtuvimos de “Instalado correctamente” sin embargo ya podríamos hacer un app que haga uso de opencv, como aplicar filtros, segmentar etc..



Otros:

El siguiente error salía cuando se utilizase OpenCV 3.4.10 en este caso se tenía que instalar Android 4.0 desde el SDK Manager.

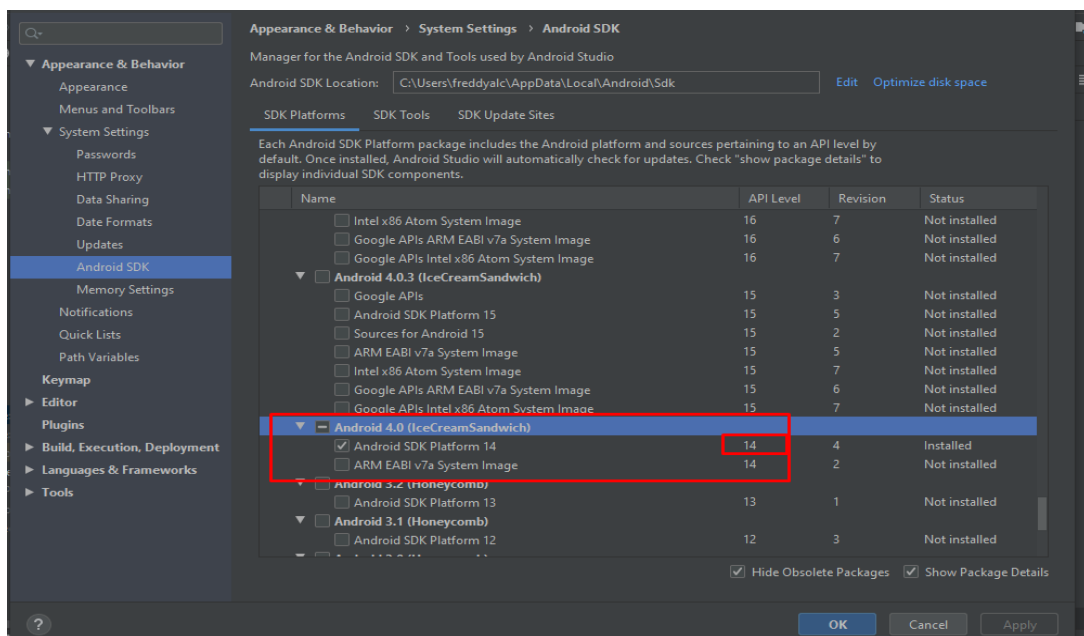
Gradle sync failed:

com.android.tools.idea.gradle.project.sync.idea.issues.SdkPlatformNotFoundException:  
Module: 'openCVLibrary3410' platform 'android-14' not found. (13 s 448 ms)

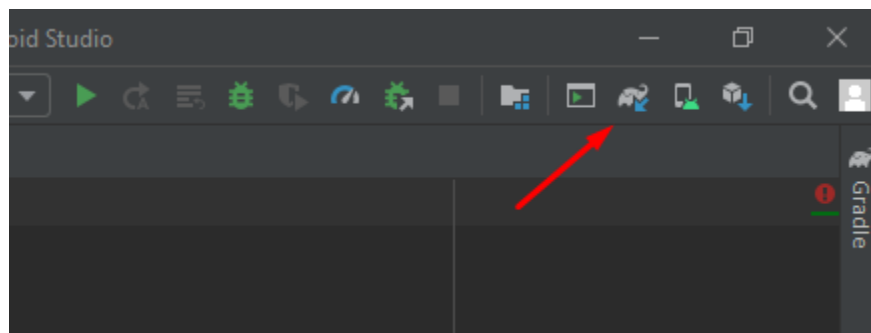
Resulta que no habíamos abierto el layout xml del activity main y al abrirlo me salía que sincronizara entonces sincronice, pero salió el error:

Module: 'openCVLibrary3410' platform 'android-14' not found.

En la parte inferior de android studio, para solucionarlos abrimos el SDK Manager e instalamos Android 4.0:

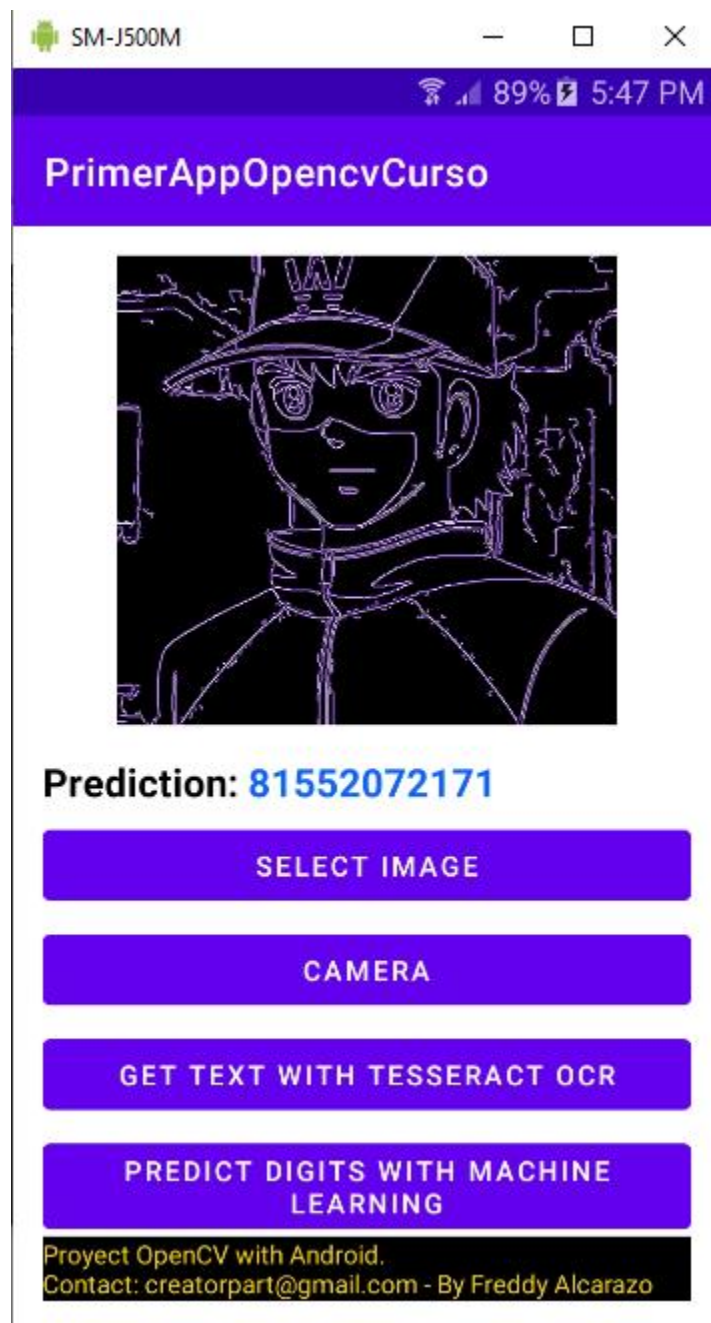


Luego sincronizamos el **gradle** presionando este icono en la parte superior derecha:



Y listo ya podremos diseñar nuestra aplicación con OpenCV:

Muestra de que se agregó opencv a una aplicación que lo requería en Android Studio 4.2.2



¡Si les interesa la aplicación inscribáanse a mi curso de Machine Learning con Android en Udemy!

Informática y software > Informática y software, otros > Inteligencia artificial

## Machine Learning con Android utilizando Tensorflow Lite

Descubre cómo implementar modelos de inteligencia artificial en una aplicación android utilizando Tensorflow Lite.

5,0 ★★★★★ (1 calificación) 9 estudiantes

Creado por [Freddy Daniel Alcarazo Ibañez](#)

🕒 Fecha de la última actualización: 4/2021 🌐 Español 🗣️ Español [automático]

Lista de deseos ❤️

Compartir ➔

Regalar este curso



Vista previa de este curso

19,99 US\$

Añadir a la cesta

Comprar ahora

Link: <https://www.udemy.com/course/machine-learning-con-android-utilizando-tensorflow-lite/>