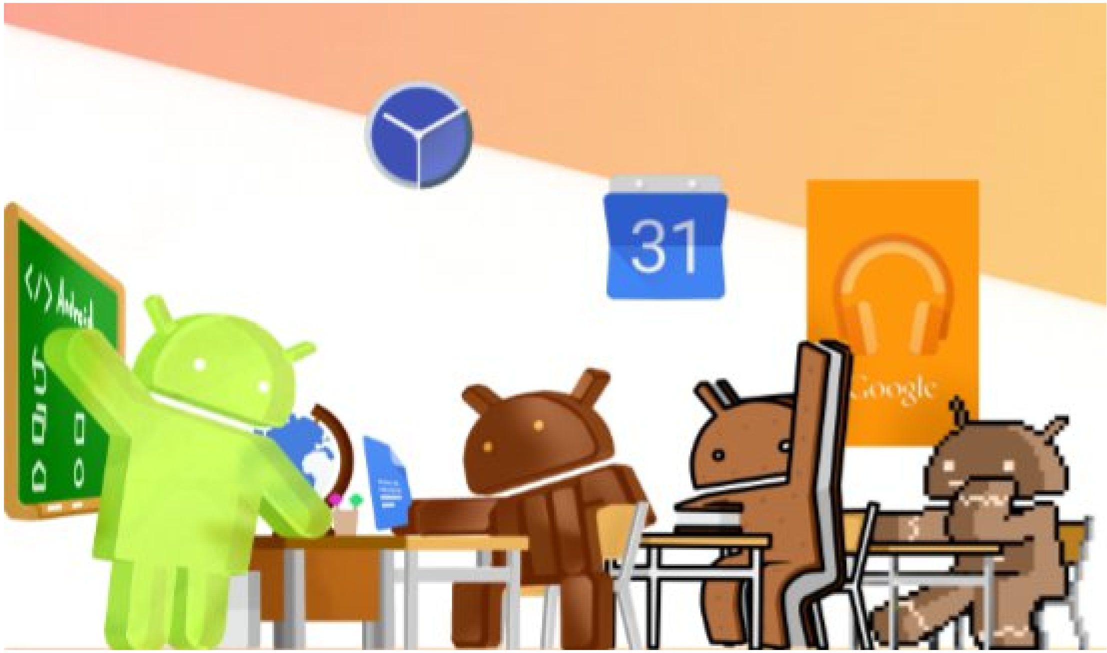
PROGRAMACIÓN MULTIMEDIA Y DE DISPOSITIVOS MÓVILES

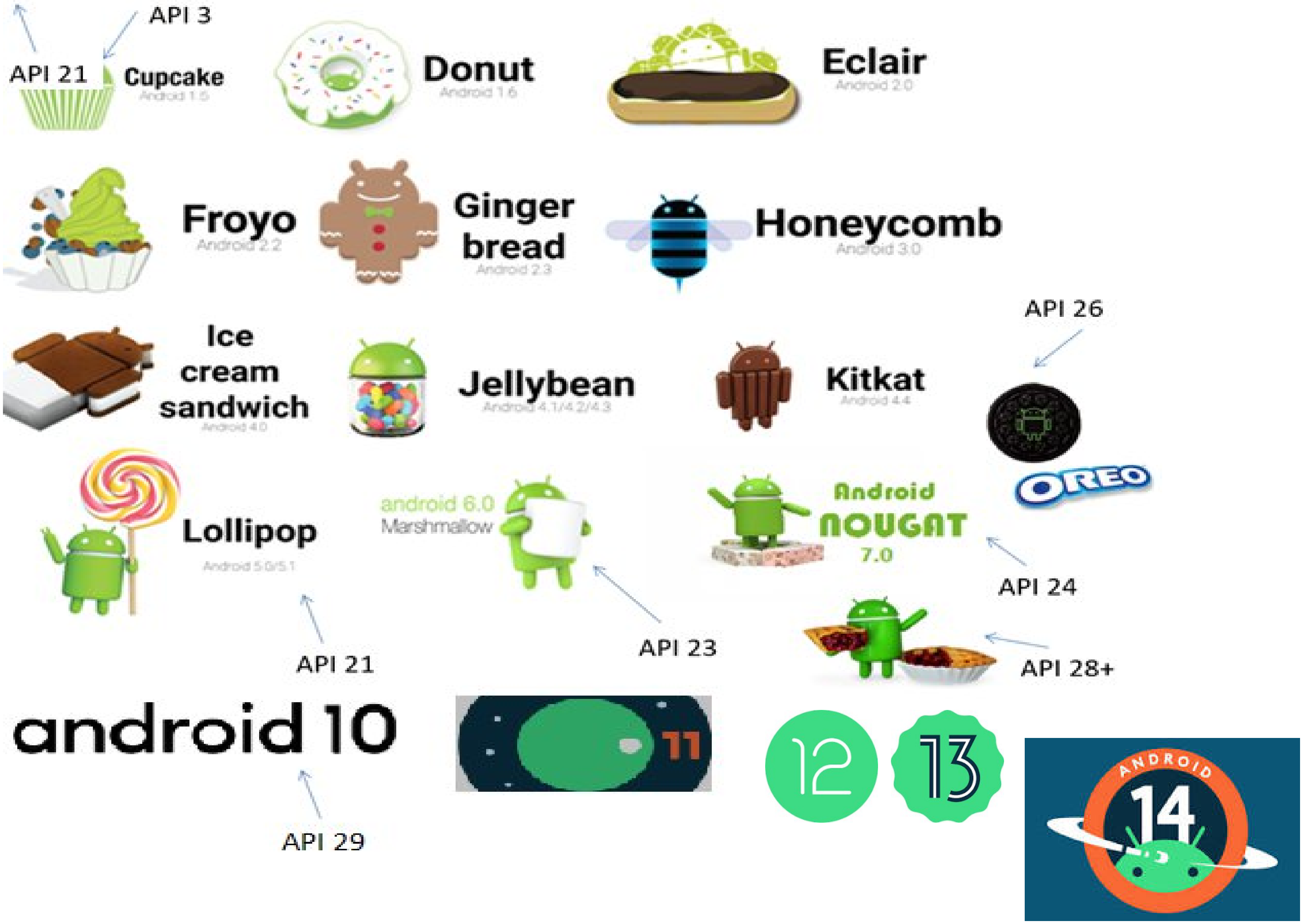
UT1. ANDROID. EL SISTEMA OPERATIVO



IES Ribera del Tajo
Enseñanza bilingüe

www.iesriberadeltajo.com
facebook.com/ies.riberadeltajo
@IESRiberaTajo

Versiones


API 3
API 21
Cupcake
Android 1.5
Donut
Android 1.6
Eclair
Android 2.0
Froyo
Android 2.2
Ginger bread
Android 2.3
Honeycomb
Android 3.0
Ice cream sandwich
Android 4.0
Jellybean
Android 4.1/4.2/4.3
Kitkat
Android 4.4
API 26
OREO
Lollipop
Android 5.0/5.1
android 6.0
Marshmallow
Android NOUGAT 7.0
API 24
API 21
API 23
API 28+
android 10
11
12
13
ANDROID 14
API 29

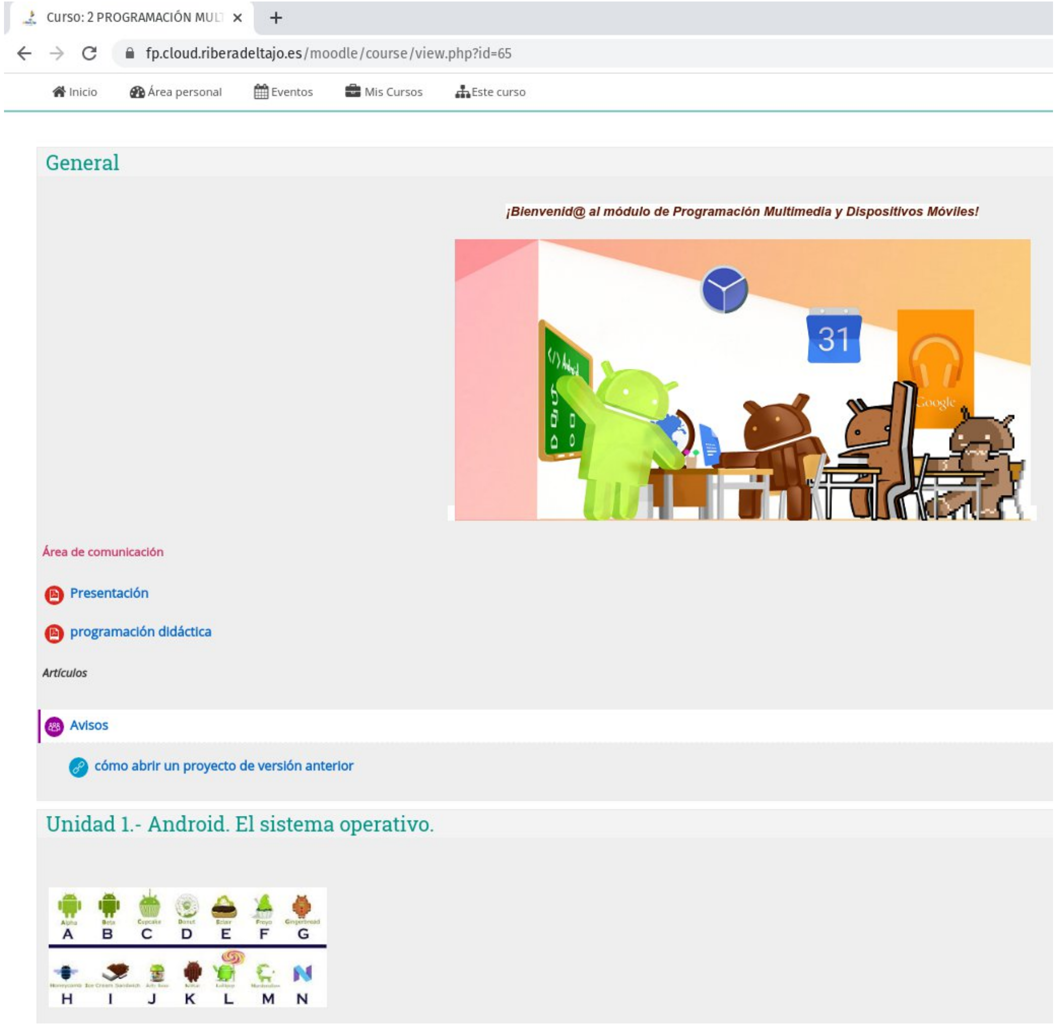
Compatibilidad

Cupcake
Android 1.5
Donut
Android 1.6
Eclair
Android 2.0
Froyo
Android 2.2
Ginger bread
Android 2.3
Honeycomb
Android 3.0
Ice cream sandwich
Android 4.0
Jellybean
Android 4.1/4.2/4.3
Kitkat
Android 4.4
Lollipop
Android 5.0/5.1
android 6.0
Marshmallow

- De dispositivo
  - *Android* Compatible
  - Características (sensores, bluetooth, etc)
- Cada versión agrega nuevas APIs
    - minSdkVersion
- De pantalla
  - Small, normal, large, xlarge
  - mdpi, hdpi, xhdpi, xxhdpi y otras

Documentación

- El propio curso


Curso: 2 PROGRAMACIÓN MULT ×
+
fp.cloud.riberadeltajo.es/moodle/course/view.php?id=65
Inicio
Área personal
Eventos
Mis Cursos
Este curso

General

¡Bienvenid@ al módulo de Programación Multimedia y Dispositivos Móviles!

Área de comunicación

Presentación

programación didáctica

Artículos

Avisos

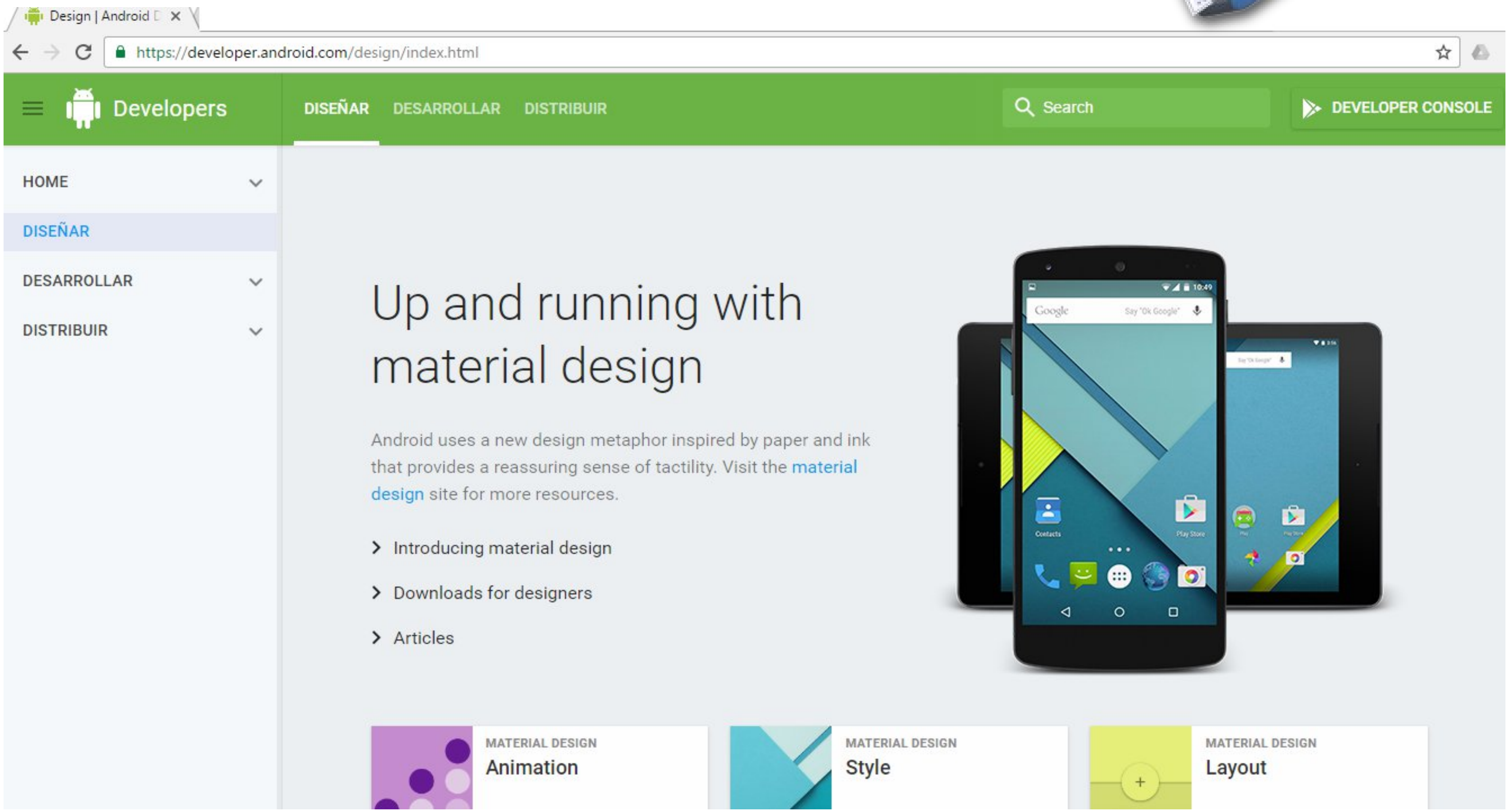
cómo abrir un proyecto de versión anterior

Unidad 1.- Android. El sistema operativo.

31
Google
A B C D E F G
H I J K L M N

Documentación

- Developer.android.com



Up and running with material design

Android uses a new design metaphor inspired by paper and ink that provides a reassuring sense of tactility. Visit the material design site for more resources.

- Introducing material design
- Downloads for designers
- Articles

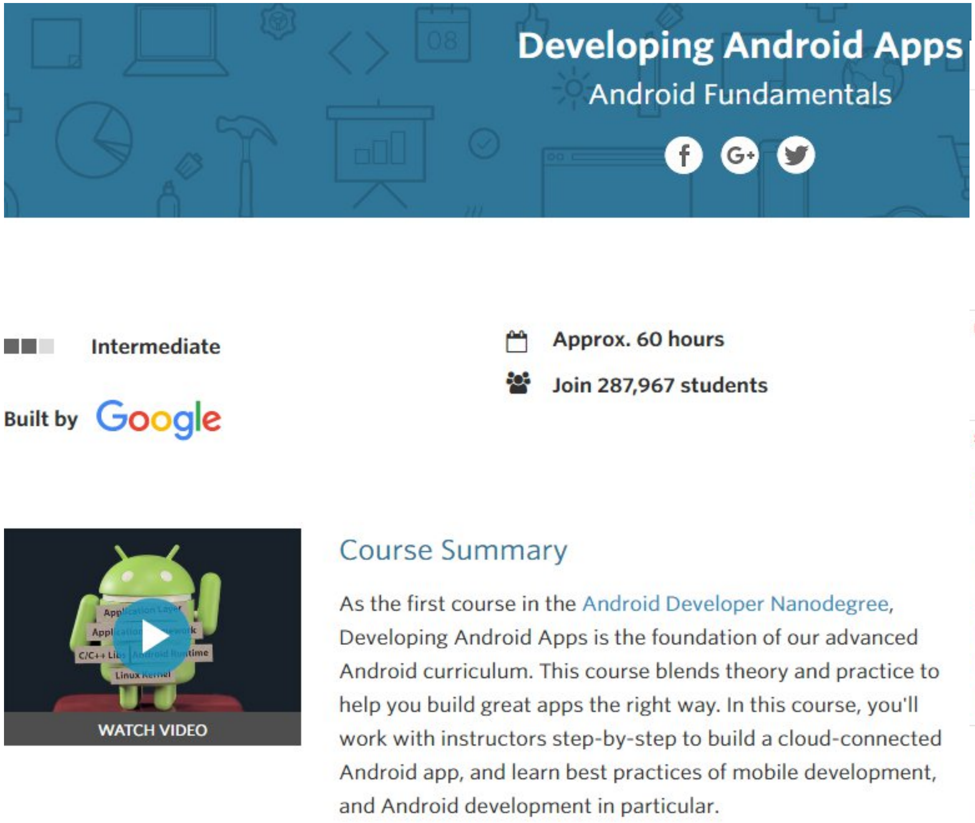
MATERIAL DESIGN
Animation

MATERIAL DESIGN
Style

MATERIAL DESIGN
Layout

Documentación

- Otros
  - – Cursos online / Videotutoriales



Developing Android Apps

Android Fundamentals

Intermediate

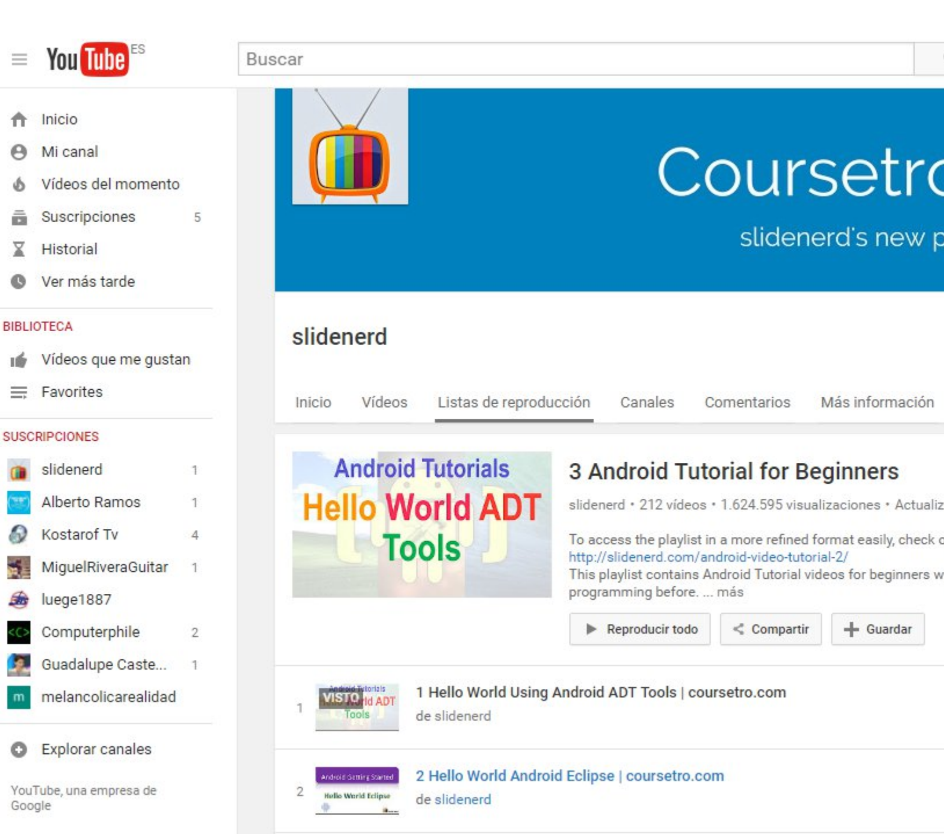
Built by Google

Approx. 60 hours

Join 287,967 students

Course Summary

As the first course in the Android Developer Nanodegree, Developing Android Apps is the foundation of our advanced Android curriculum. This course blends theory and practice to help you build great apps the right way. In this course, you'll work with instructors step-by-step to build a cloud-connected Android app, and learn best practices of mobile development, and Android development in particular.



slidenerd

3 Android Tutorial for Beginners

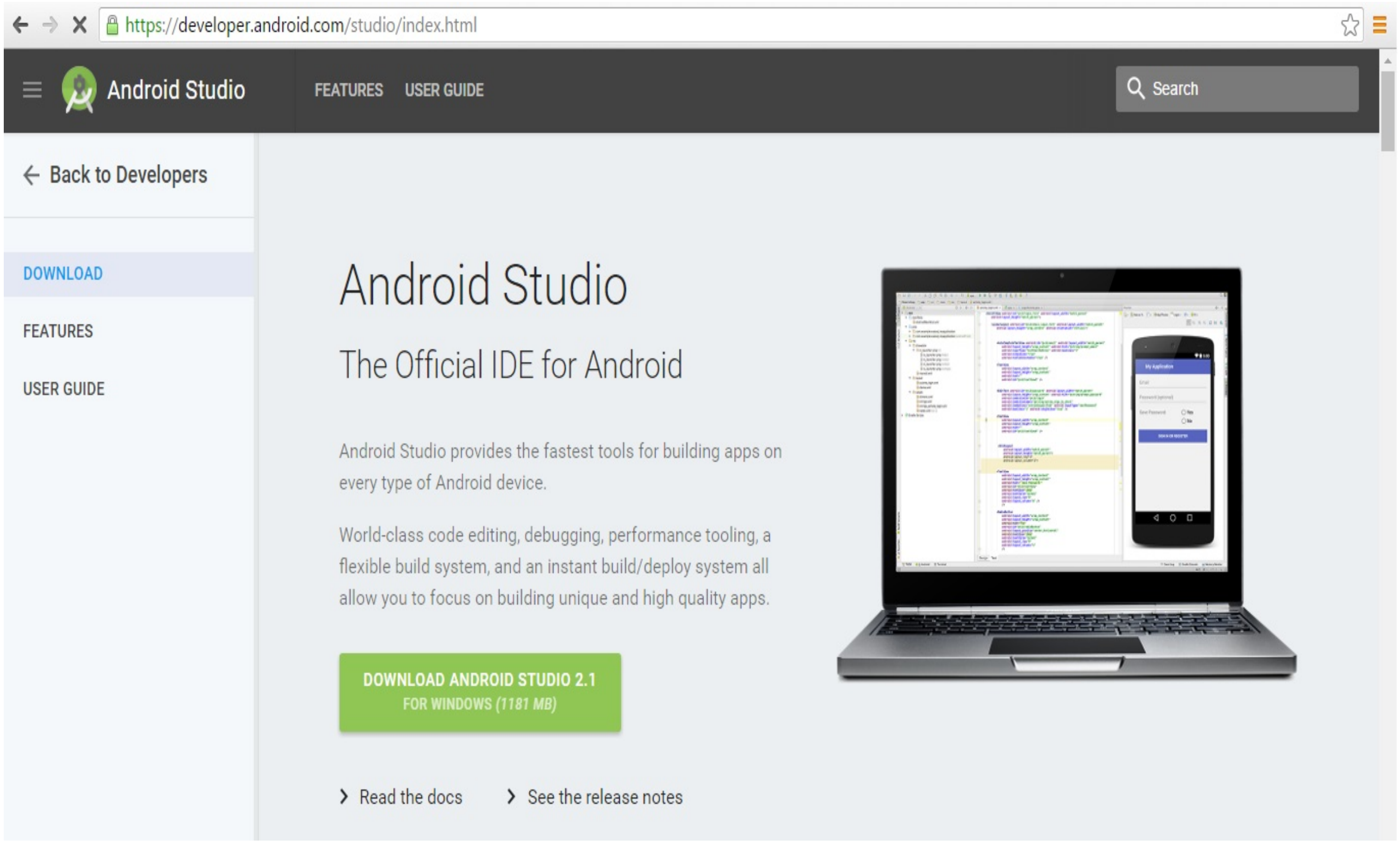
slidenerd • 212 vídeos • 1.624.595 visualizaciones • Actualiz

To access the playlist in a more refined format easily, check c
http://slidenerd.com/android-video-tutorial-2/
This playlist contains Android Tutorial videos for beginners w
programming before. ... más

1 1 Hello World Using Android ADT Tools | coursetro.com
de slidenerd

2 2 Hello World Android Eclipse | coursetro.com
de slidenerd

EL IDE



Android Studio

The Official IDE for Android

Android Studio provides the fastest tools for building apps on every type of Android device.

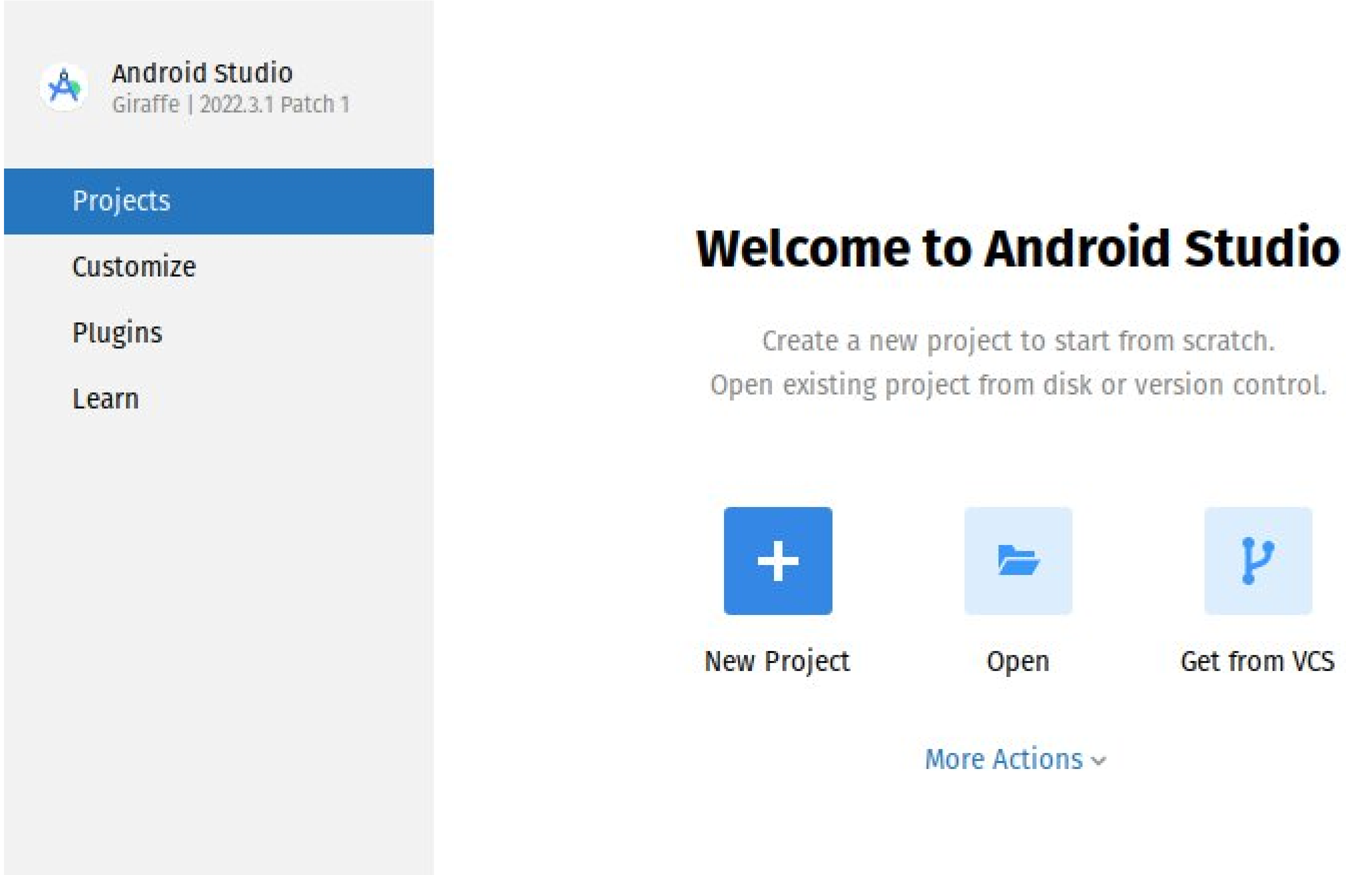
World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

- Instala Android Studio
- Sigues las instrucciones de la documentación
  - Prerrequisitos!!

Componentes de Android Studio

- Editor de texto y herramientas para desarrolladores:
  - – IntelliJ IDEA de JetBrains
- Compilador basado en Gradle
- Bibliotecas y herramientas
  - – androidx.* (jetpack)
- Emulador
- Integrado
  - – Con GitHub
  - – Desarrollo para múltiples dispositivos

Tu primer proyecto

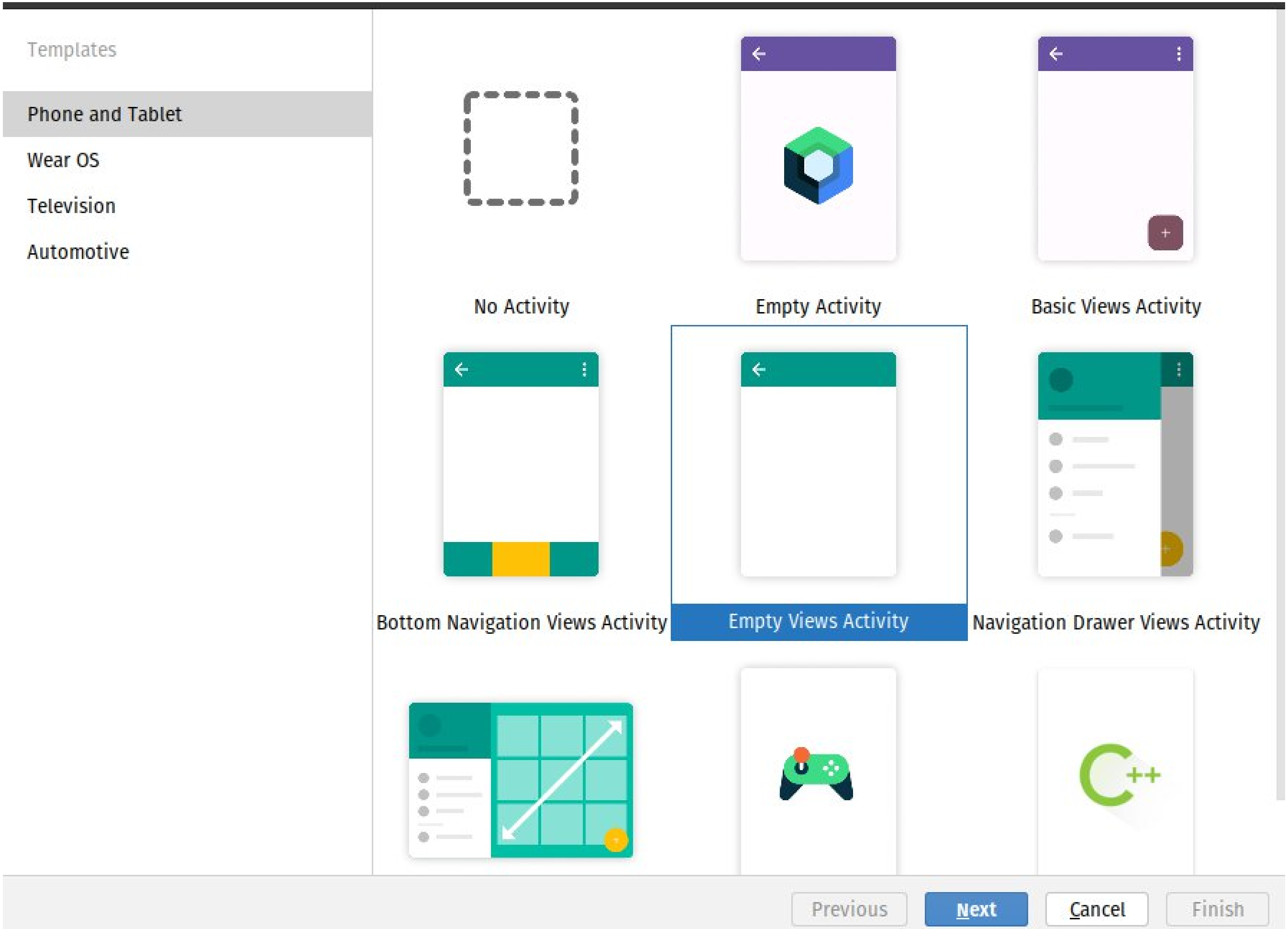


Android Studio
Giraffe | 2022.3.1 Patch 1

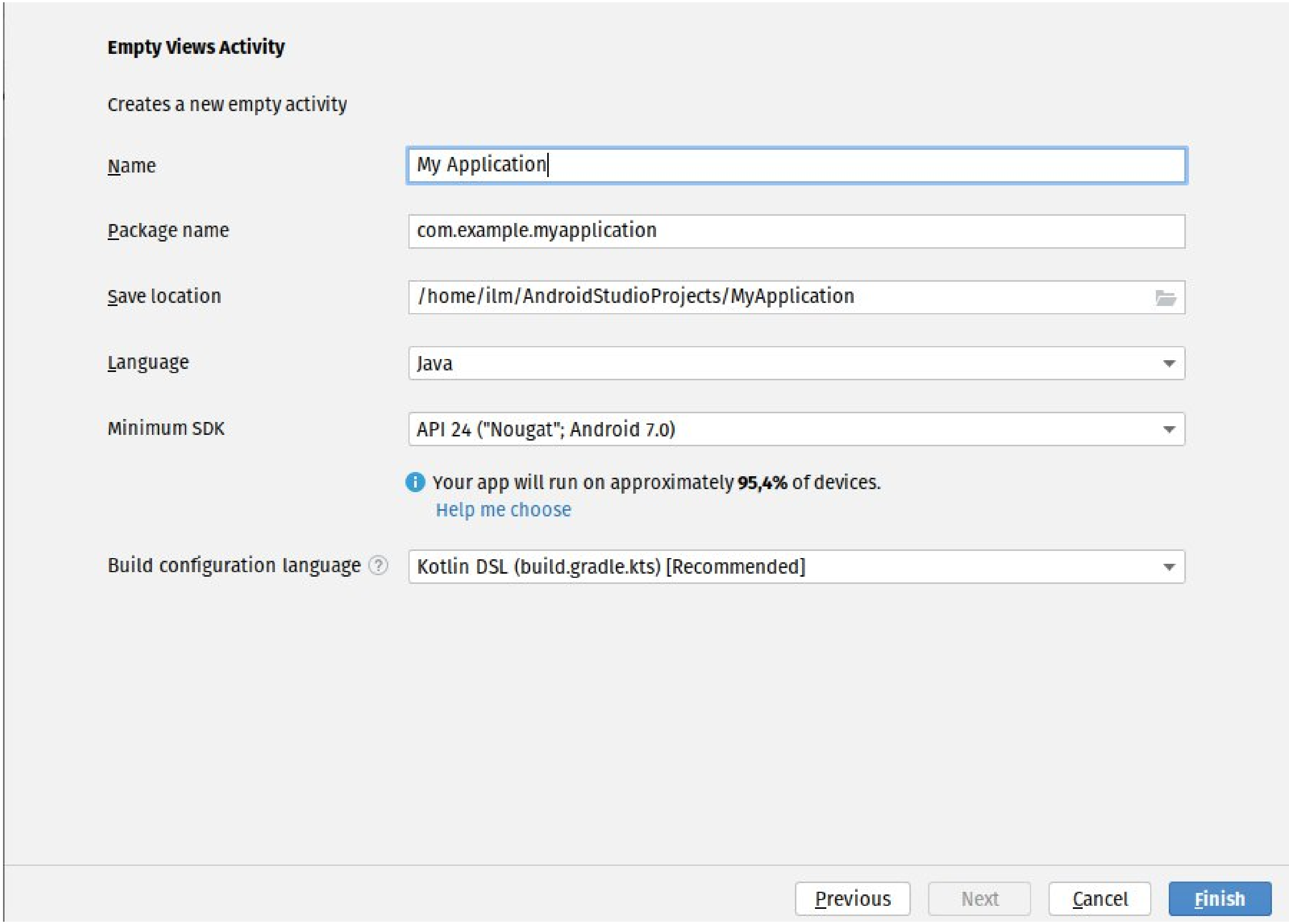
Welcome to Android Studio

Create a new project to start from scratch.
Open existing project from disk or version control.

Añade una actividad en blanco



Selecciona lenguaje y API level



Empty Views Activity

Creates a new empty activity

Name: My Application

Package name: com.example.myapplication

Save location: /home/ilm/AndroidStudioProjects/MyApplication

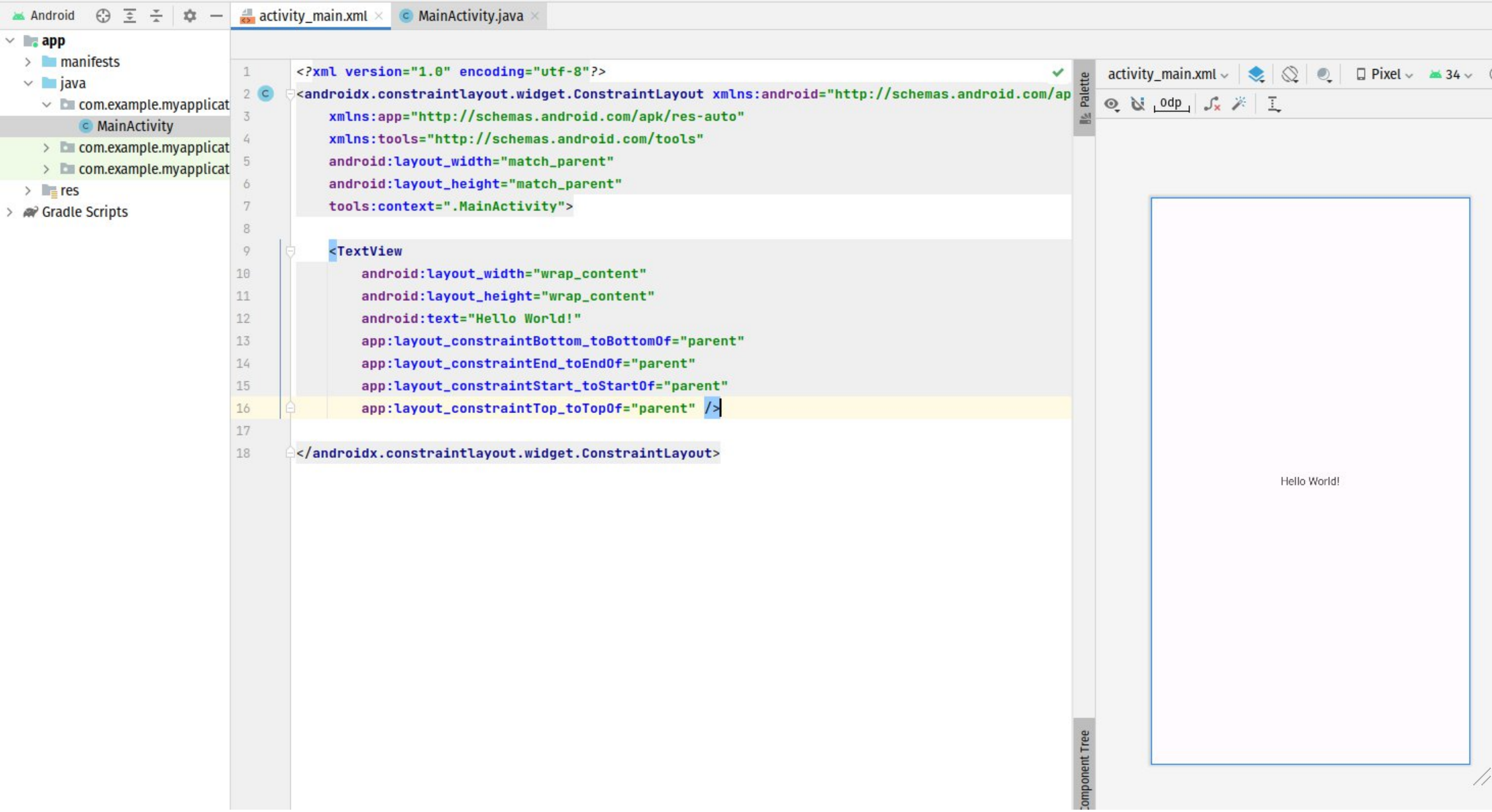
Language: Java

Minimum SDK: API 24 ("Nougat"; Android 7.0)

Your app will run on approximately **95,4%** of devices.
Help me choose

Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

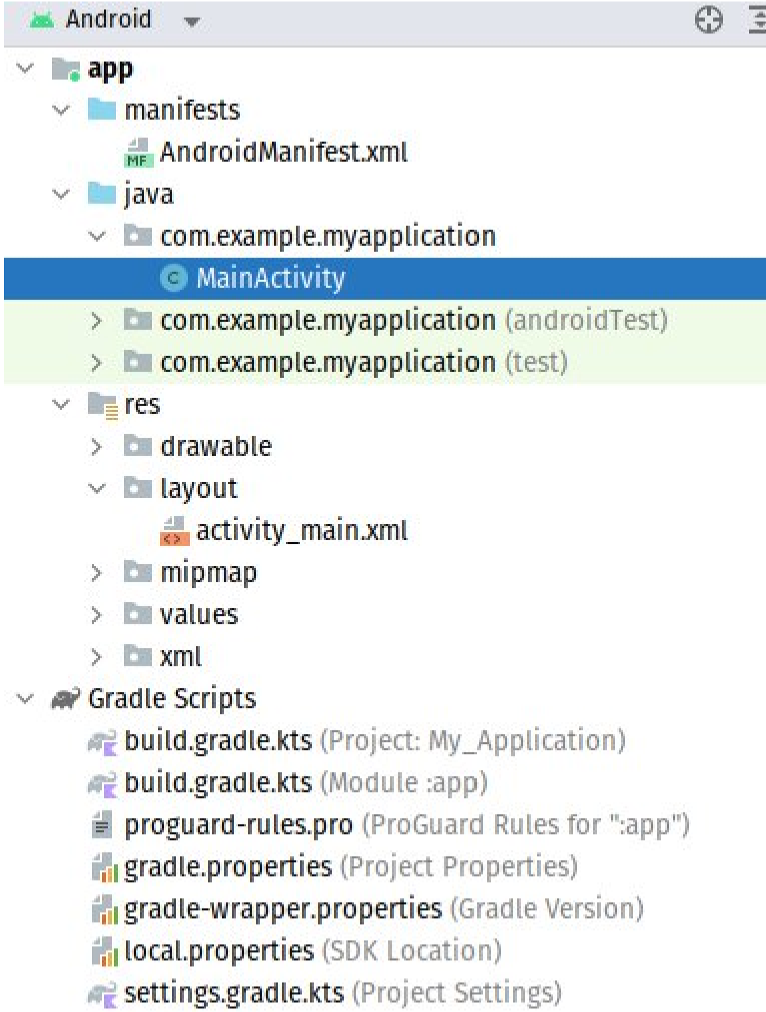
Primer contacto



```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/ap
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

¿Dónde está el código?

- AndroidManifest.xml
  - descripción de la aplicación que estamos creando y qué componentes (servicios, actividades, imágenes, etc.)
- JAVA
  - – MainActivity.java
  - – Comportamiento de la aplicación
- Res
  - – Recursos de la aplicación (imágenes, sonidos, menús…)
  - – Layout / cuadrícula
    - Descripción de la interfaz de usuario
      - – activity_main.xml



```
Android
app
  manifests
    AndroidManifest.xml
  java
    com.example.myapplication
      MainActivity
    com.example.myapplication (androidTest)
    com.example.myapplication (test)
  res
    drawable
    layout
      activity_main.xml
    mipmap
    values
    xml
Gradle Scripts
  build.gradle.kts (Project: My_Application)
  build.gradle.kts (Module :app)
  proguard-rules.pro (ProGuard Rules for ":app")
  gradle.properties (Project Properties)
  gradle-wrapper.properties (Gradle Version)
  local.properties (SDK Location)
  settings.gradle.kts (Project Settings)
```

Dónde comienzo a escribir mi código

- CALLBACK: Código que responde a un evento o suceso en la activity
- Ciclo de vida de una actividad

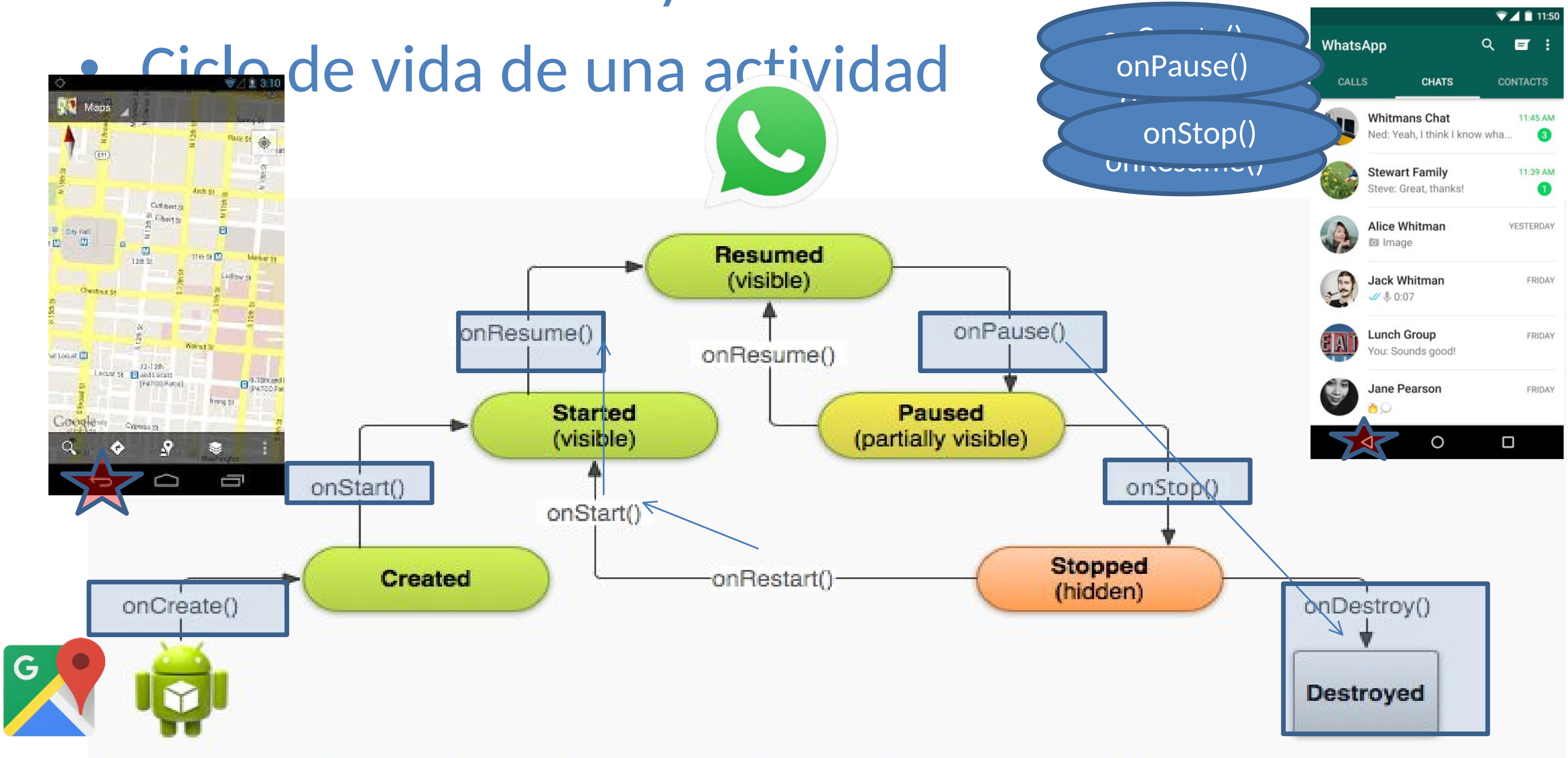
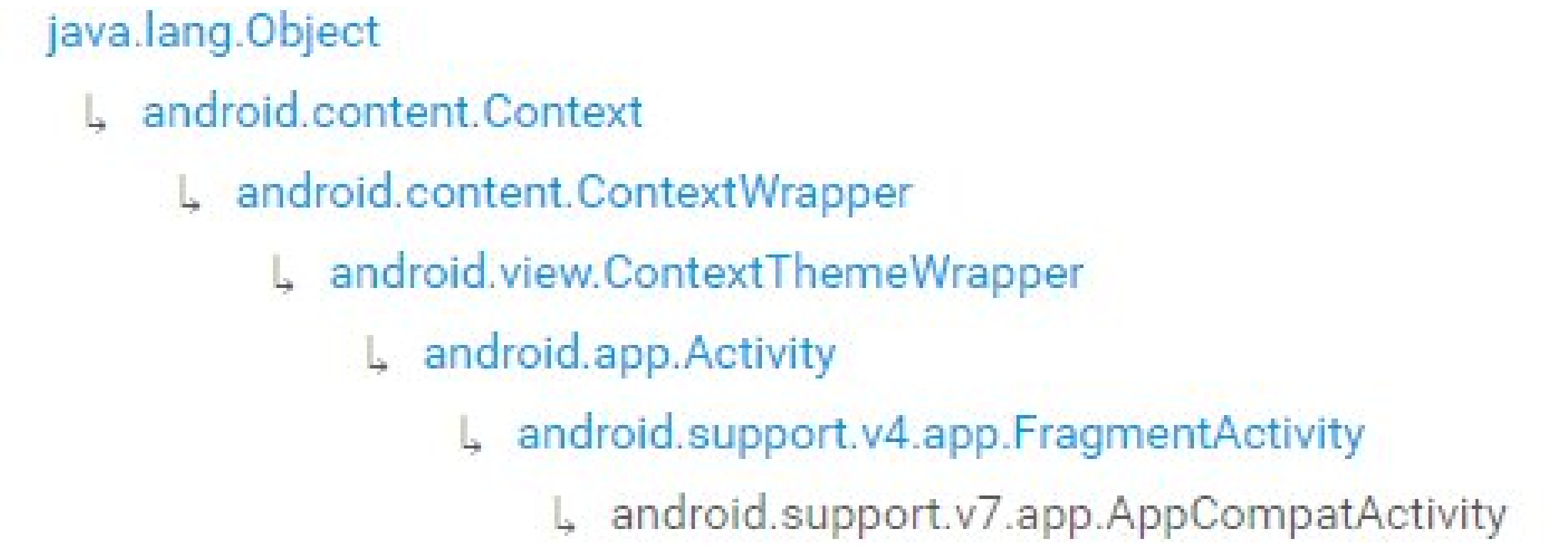


Figure 1. A simplified illustration of the Activity lifecycle, expressed as a step pyramid. This shows how, for every callback used to take the activity a step toward the Resumed state at the top, there's a callback method that takes the activity a step down. The activity can also return to the resumed state from the Paused and Stopped state.

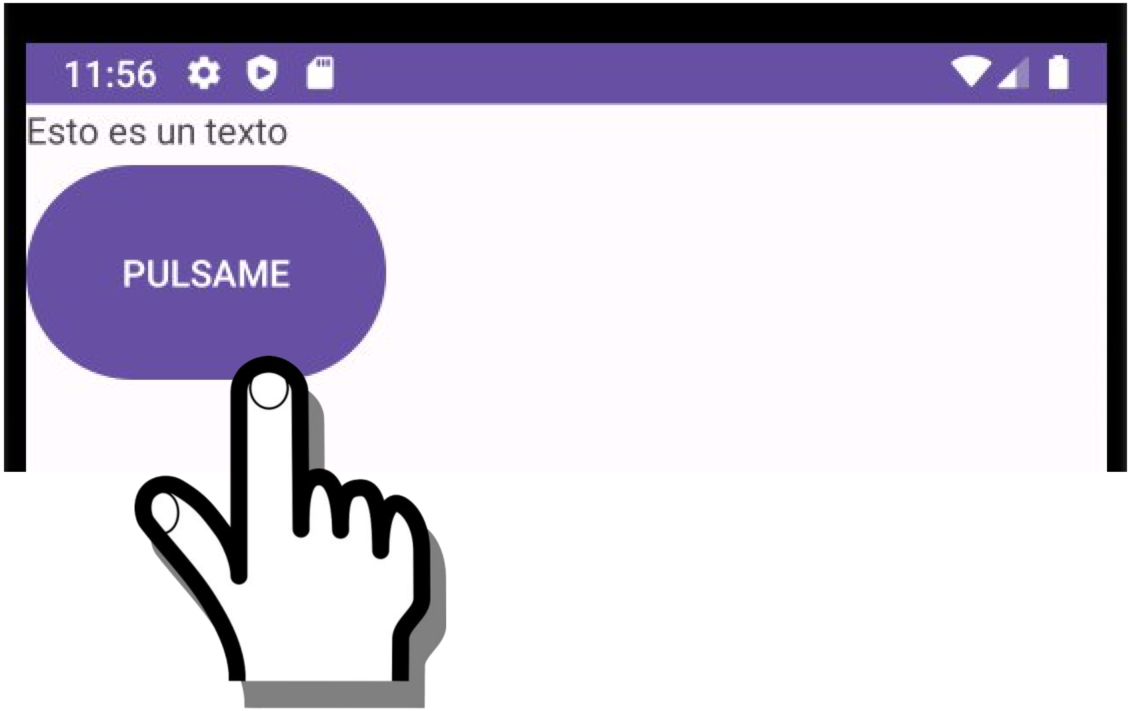
Activity y subclases

AppCompatActivity



```
java.lang.Object
   ↳ android.content.Context
       ↳ android.content.ContextWrapper
           ↳ android.view.ContextThemeWrapper
               ↳ android.app.Activity
                   ↳ android.support.v4.app.FragmentActivity
                       ↳ android.support.v7.app.AppCompatActivity
```

Creando mi primera App

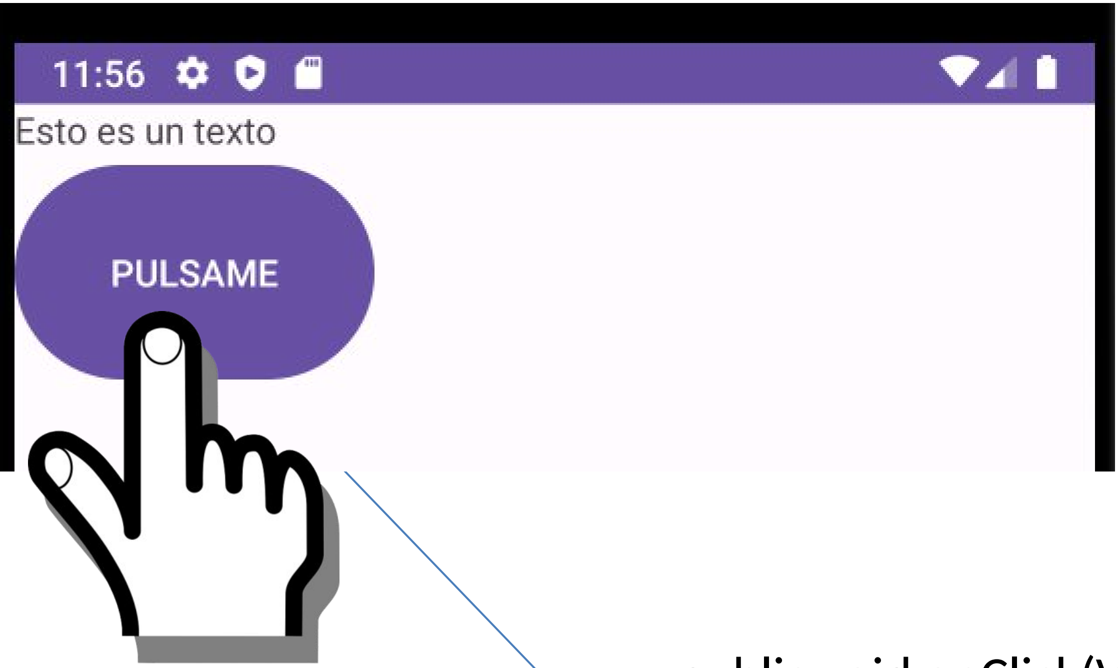


Esto es un texto

Creando mi primera App (Código)

```
protected void onCreate(Bundle savedInstanceState) {
    Button miBoton;
    miBoton=(Button)findViewById(R.id.button);
    miBoton.setOnClickListener(this);
}
```

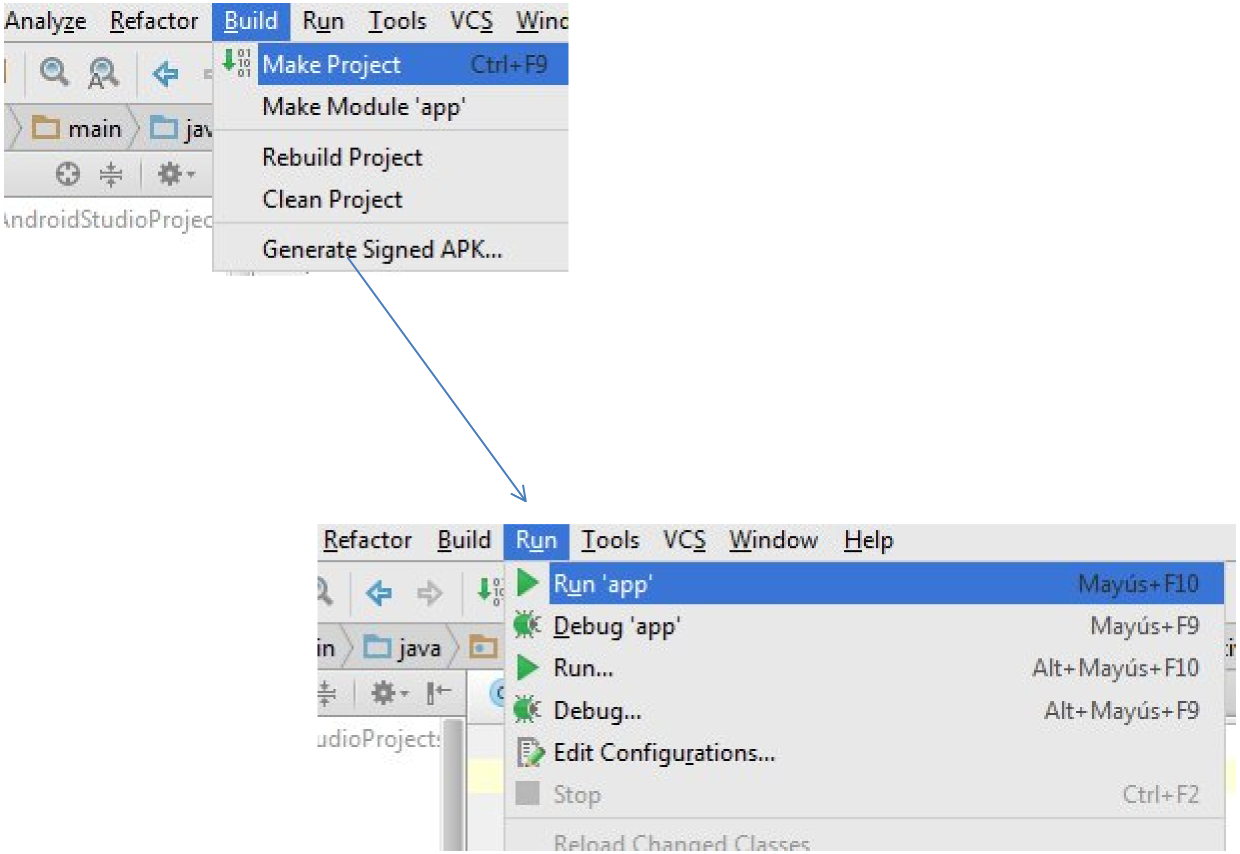
implements
View.OnClickListener



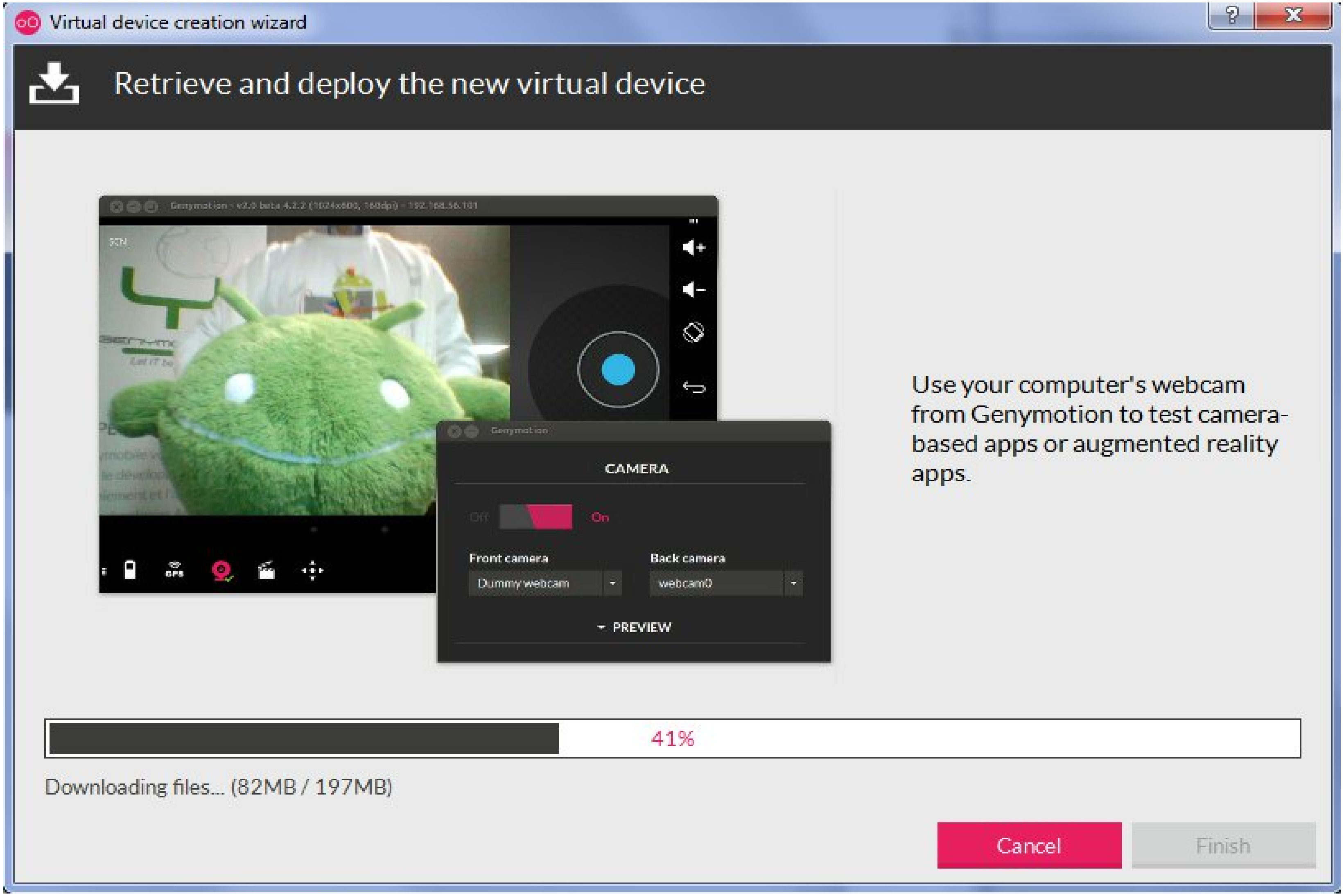
Esto es un texto

```
public void onClick(View view) {
    //responde al evento Click
    miTexto=(TextView)findViewById(R.id.textView);
    miTexto.setText("pulsado");
}
```

Ejecutar el proyecto en el emulador



Otras opciones



Virtual device creation wizard

Retrieve and deploy the new virtual device

Genymotion - v2.0 beta 4.2.2 (1024x600, 160dpi) - 192.168.56.101

CAMERA

Off ● On

Front camera: Dummy webcam

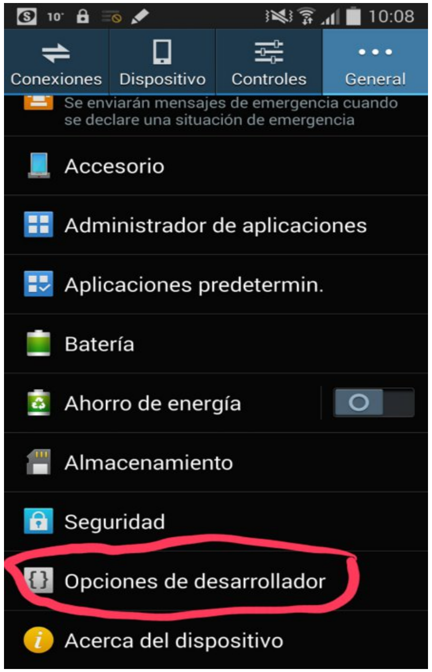
Back camera: webcam0

Use your computer's webcam from Genymotion to test camera-based apps or augmented reality apps.

41%

Downloading files... (82MB / 197MB)

Depuración modo USB



10:08

Conexiones | Dispositivo | Controles | General

Se enviarán mensajes de emergencia cuando se declare una situación de emergencia

Accesorio

Administrador de aplicaciones

Aplicaciones predetermin.

Batería

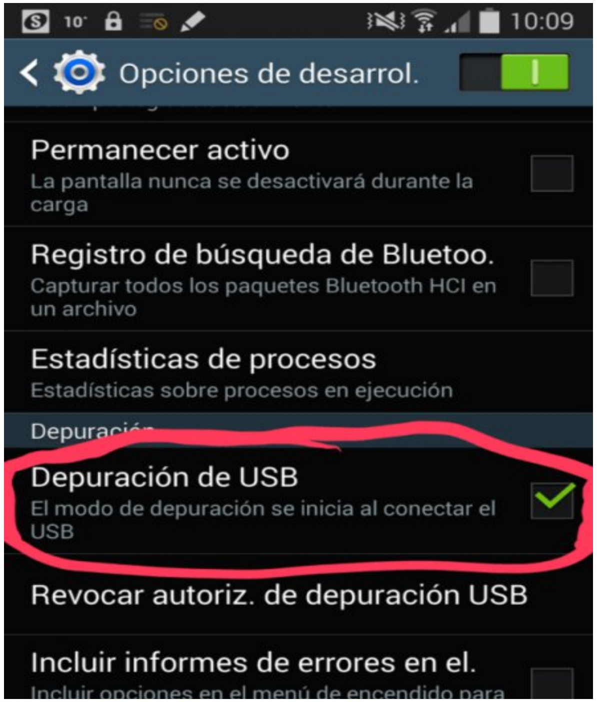
Ahorro de energía ○

Almacenamiento

Seguridad

Opciones de desarrollador

Acerca del dispositivo



10:09

Opciones de desarrol.

Permanecer activo
La pantalla nunca se desactivará durante la carga □

Registro de búsqueda de Bluetoo.
Capturar todos los paquetes Bluetooth HCI en un archivo □

Estadísticas de procesos
Estadísticas sobre procesos en ejecución

Depuración

Depuración de USB
El modo de depuración se inicia al conectar el USB ☑

Revocar autoriz. de depuración USB

Incluir informes de errores en el.
Incluir opciones en el menú de encendido para □

Now, please follow these steps to debug your Huawei Mate 7/8.

Step 1 : Turn on your Huawei Mate 7 or Huawei Mate 8.

Step 2 : Find and open "**Settings**" option.

Step 3 : Select "**About phone**", at the bottom of the interface.

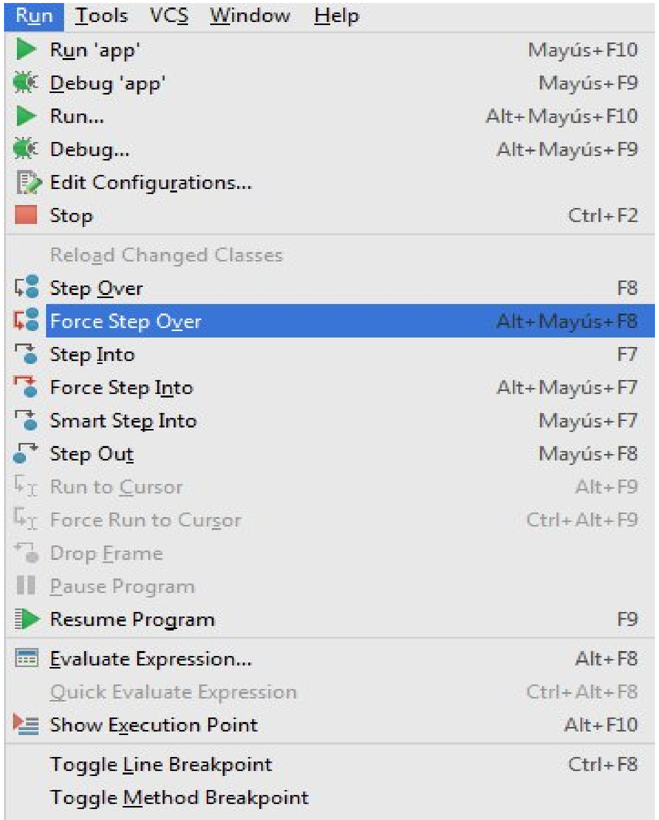
Step 4 : Tap "**Build number**" almost seven times you will see a prompt "**Developer mode has been enabled**".

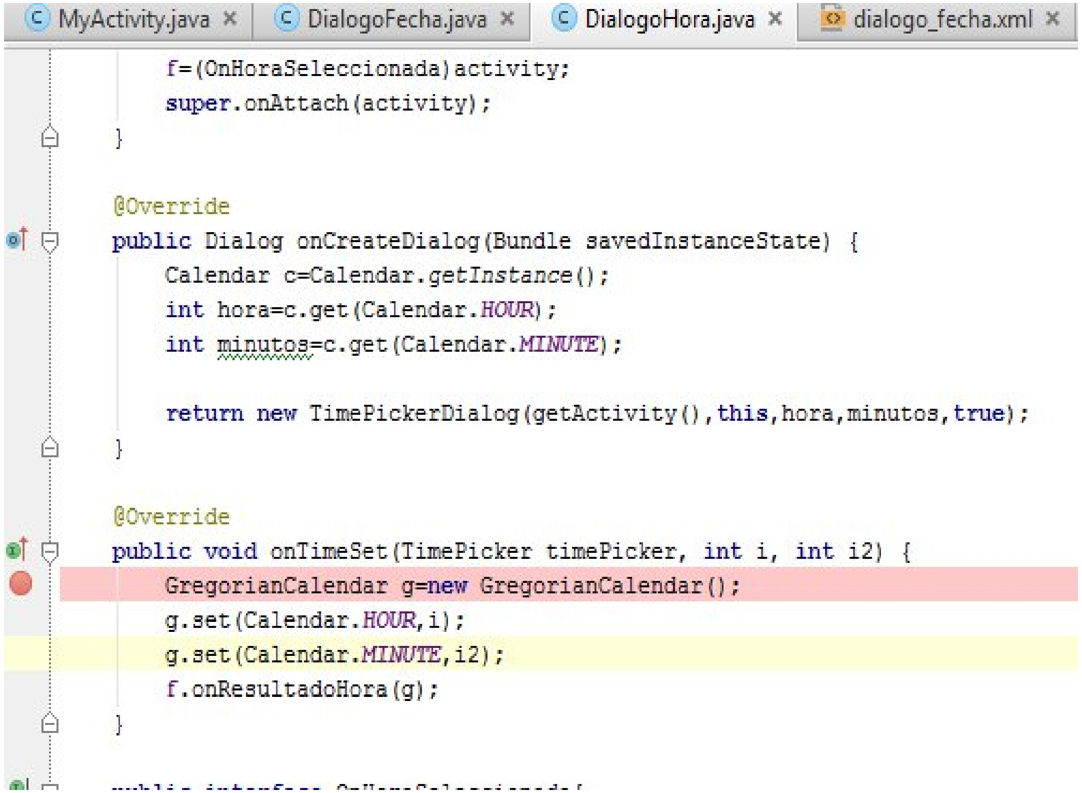
Step 5 : Back to setting panel where you can see the "**Developer options**", select **Developer options**.

Step 6 : Slide to turn the "**USB debugging**" on, and you can use your phone as a developer.

Step 7 : With all these steps done, you have debuged your Huawei Mate 7 or Huawei Mate 8 successfully.

Depuración en dispositivos reales



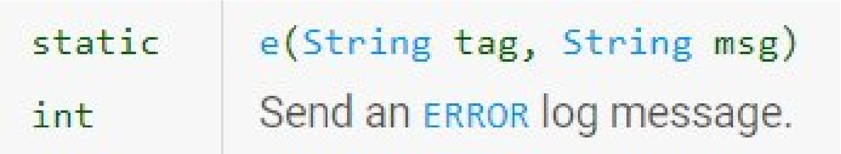


```
        f=(OnHoraSeleccionada)activity;
        super.onAttach(activity);
    }

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        Calendar c=Calendar.getInstance();
        int hora=c.get(Calendar.HOUR);
        int minutos=c.get(Calendar.MINUTE);

        return new TimePickerDialog(getActivity(),this,hora,minutos,true);
    }

    @Override
    public void onTimeSet(TimePicker timePicker, int i, int i2) {
        GregorianCalendar g=new GregorianCalendar();
        g.set(Calendar.HOUR,i);
        g.set(Calendar.MINUTE,i2);
        f.onResultadoHora(g);
    }

    public interface OnHoraSeleccionada{
```

- Depuración sencilla a través del IDE

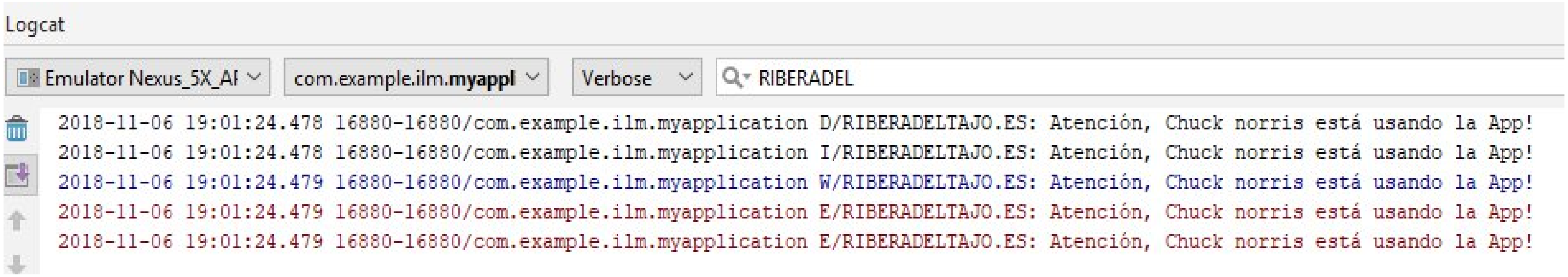
LOG CAT

android.util.Log

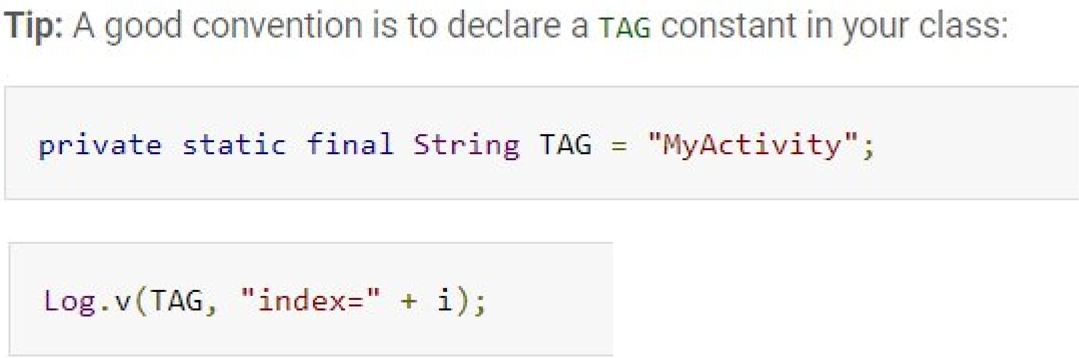


static int	e(String tag, String msg) Send an ERROR log message.

```
public final String TAG="RIBERADELTAJO.ES";

Log.d(TAG,"Atención, Chuck norris está usando la App!");
Log.i(TAG, "Atención, Chuck norris está usando la App!");
Log.w(TAG,"Atención, Chuck norris está usando la App!");
Log.e(TAG,"Atención, Chuck norris está usando la App!");
Log.wtf(TAG,"Atención, Chuck norris está usando la App!");
```



```
2018-11-06 19:01:24.478 16880-16880/com.example.ilm.myapplication D/RIBERADELTAJO.ES: Atención, Chuck norris está usando la App!
2018-11-06 19:01:24.478 16880-16880/com.example.ilm.myapplication I/RIBERADELTAJO.ES: Atención, Chuck norris está usando la App!
2018-11-06 19:01:24.479 16880-16880/com.example.ilm.myapplication W/RIBERADELTAJO.ES: Atención, Chuck norris está usando la App!
2018-11-06 19:01:24.479 16880-16880/com.example.ilm.myapplication E/RIBERADELTAJO.ES: Atención, Chuck norris está usando la App!
2018-11-06 19:01:24.479 16880-16880/com.example.ilm.myapplication E/RIBERADELTAJO.ES: Atención, Chuck norris está usando la App!
```



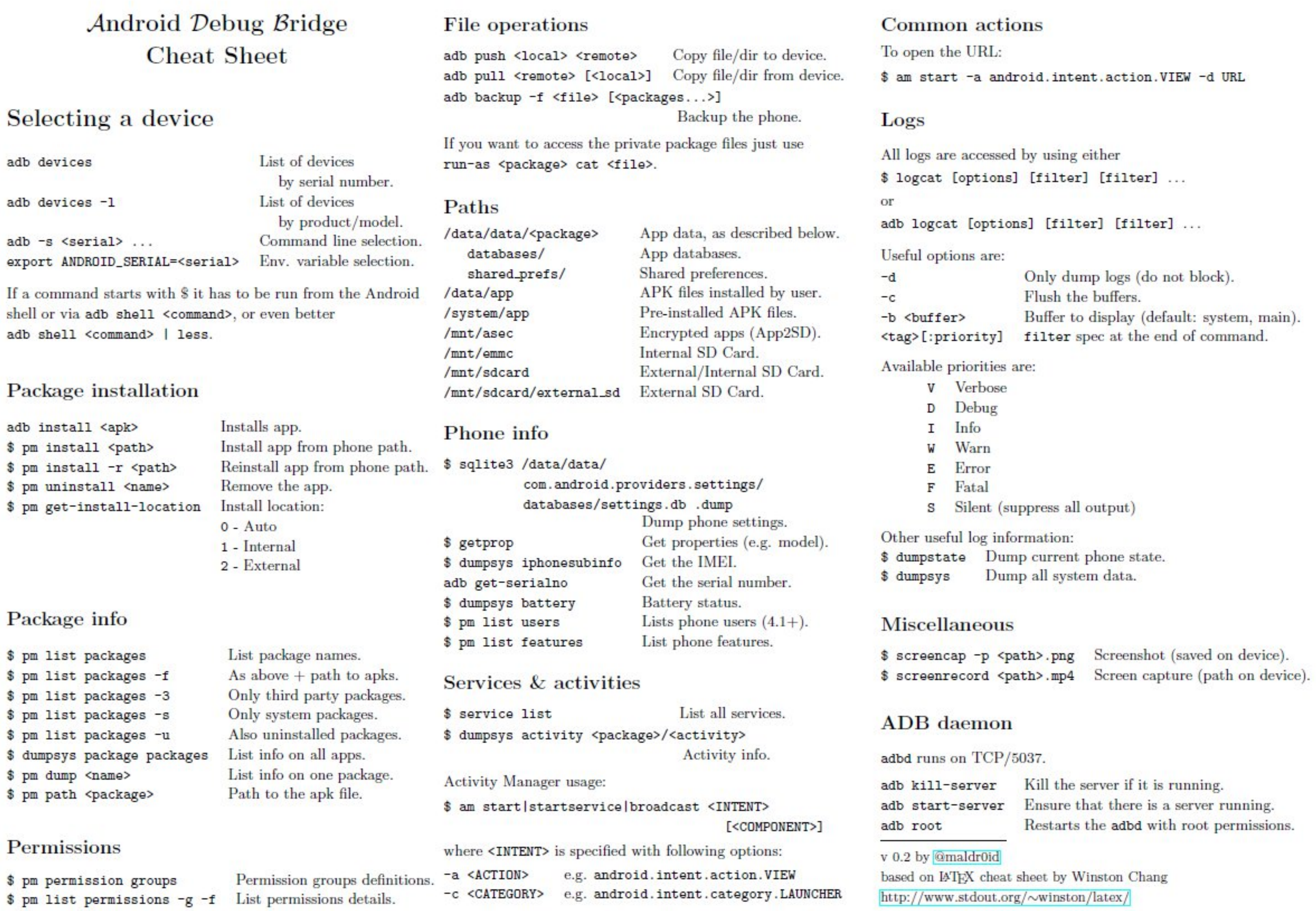
Tip: A good convention is to declare a `TAG` constant in your class:

```
private static final String TAG = "MyActivity";
```

```
Log.v(TAG, "index=" + i);
```

Android debug bridge

- te permite comunicarte con una instancia de un emulador o un dispositivo Android conectado



Android Debug Bridge Cheat Sheet

Selecting a device

`adb devices`	List of devices by serial number.
`adb devices -l`	List of devices by product/model.
`adb -s <serial> ...`	Command line selection.
`export ANDROID_SERIAL=<serial>`	Env. variable selection.

If a command starts with $ it has to be run from the Android shell or via `adb shell <command>`, or even better `adb shell <command> | less`.

Package installation

`adb install <apk>`	Installs app.
`$ pm install <path>`	Install app from phone path.
`$ pm install -r <path>`	Reinstall app from phone path.
`$ pm uninstall <name>`	Remove the app.
`$ pm get-install-location`	Install location: 0 - Auto 1 - Internal 2 - External

Package info

`$ pm list packages`	List package names.
`$ pm list packages -f`	As above + path to apks.
`$ pm list packages -3`	Only third party packages.
`$ pm list packages -s`	Only system packages.
`$ pm list packages -u`	Also uninstalled packages.
`$ dumpsys package packages`	List info on all apps.
`$ pm dump <name>`	List info on one package.
`$ pm path <package>`	Path to the apk file.

Permissions

`$ pm permission groups`	Permission groups definitions.
`$ pm list permissions -g -f`	List permissions details.

File operations

`adb push <local> <remote>`	Copy file/dir to device.
`adb pull <remote> [<local>]`	Copy file/dir from device.
`adb backup -f <file> [<packages...>]`	Backup the phone.

If you want to access the private package files just use `run-as <package> cat <file>`.

Paths

`/data/data/<package>`	App data, as described below.
`databases/`	App databases.
`shared_prefs/`	Shared preferences.
`/data/app`	APK files installed by user.
`/system/app`	Pre-installed APK files.
`/mnt/asec`	Encrypted apps (App2SD).
`/mnt/emmc`	Internal SD Card.
`/mnt/sdcard`	External/Internal SD Card.
`/mnt/sdcard/external_sd`	External SD Card.

Phone info

`$ sqlite3 /data/data/com.android.providers.settings/databases/settings.db .dump`	Dump phone settings.
`$ getprop`	Get properties (e.g. model).
`$ dumpsys iphonesubinfo`	Get the IMEI.
`adb get-serialno`	Get the serial number.
`$ dumpsys battery`	Battery status.
`$ pm list users`	Lists phone users (4.1+).
`$ pm list features`	List phone features.

Services & activities

`$ service list`	List all services.
`$ dumpsys activity <package>/<activity>`	Activity info.

Activity Manager usage:

```
$ am start|startservice|broadcast <INTENT>
                                  [<COMPONENT>]
```

where `<INTENT>` is specified with following options:

`-a <ACTION>`	e.g. `android.intent.action.VIEW`
`-c <CATEGORY>`	e.g. `android.intent.category.LAUNCHER`

Common actions

To open the URL:

```
$ am start -a android.intent.action.VIEW -d URL
```

Logs

All logs are accessed by using either

```
$ logcat [options] [filter] [filter] ...
```

or

```
adb logcat [options] [filter] [filter] ...
```

Useful options are:

`-d`	Only dump logs (do not block).
`-c`	Flush the buffers.
`-b <buffer>`	Buffer to display (default: system, main).
`<tag>[:priority]`	`filter` spec at the end of command.

Available priorities are:

`V`	Verbose
`D`	Debug
`I`	Info
`W`	Warn
`E`	Error
`F`	Fatal
`S`	Silent (suppress all output)

Other useful log information:

`$ dumpstate`	Dump current phone state.
`$ dumpsys`	Dump all system data.

Miscellaneous

`$ screencap -p <path>.png`	Screenshot (saved on device).
`$ screenrecord <path>.mp4`	Screen capture (path on device).

ADB daemon

`adbd` runs on TCP/5037.

`adb kill-server`	Kill the server if it is running.
`adb start-server`	Ensure that there is a server running.
`adb root`	Restarts the `adbd` with root permissions.

v 0.2 by @maldr0id
based on LaTeX cheat sheet by Winston Chang
http://www.stdout.org/~winston/latex/

Adb te permite entre otras cosas…

Para copiar un archivo o directorio (y sus subdirectorios) *desde* el emulador o dispositivo, usa

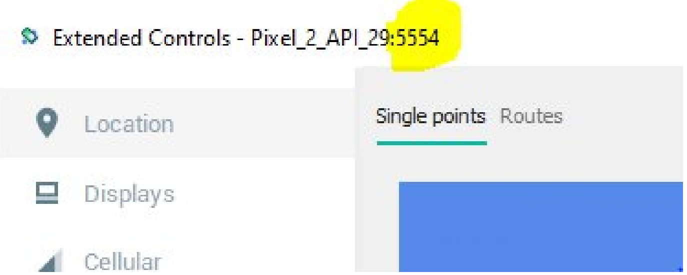
```
adb pull remote local
```

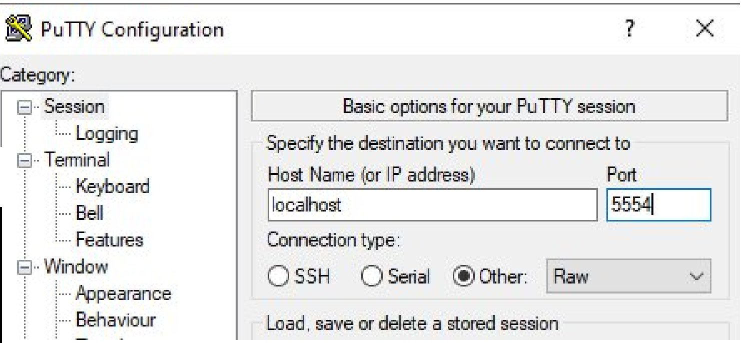
Para copiar un archivo o directorio (y sus subdirectorios) *al* emulador o dispositivo, usa

```
adb push local remote
```

```
screencap filename
```

```
screenrecord [options] filename
```

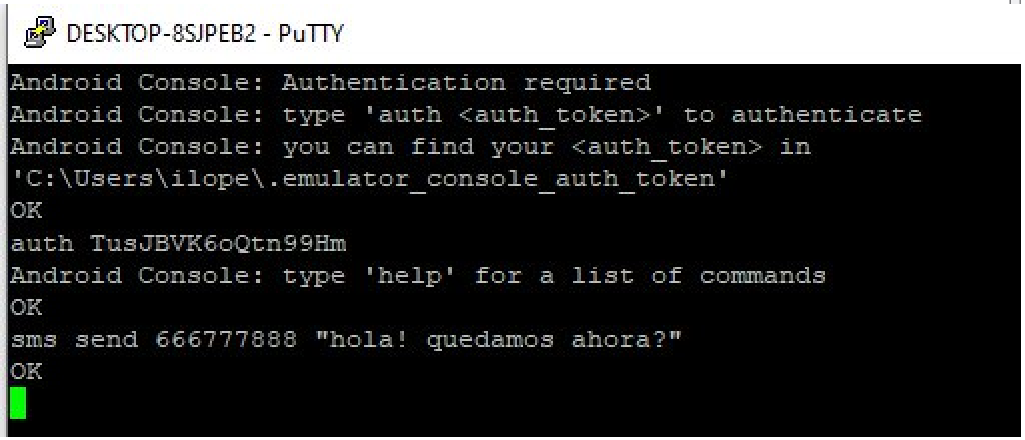



PuTTY Configuration
?
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Basic options for your PuTTY session
Specify the destination you want to connect to
Load, save or delete a stored session

Host Name (or IP address): localhost

Port: 5554

Connection type: ○ SSH ○ Serial ● Other: Raw



```
Android Console: Authentication required
Android Console: type 'auth <auth_token>' to authenticate
Android Console: you can find your <auth_token> in
'C:\Users\ilope\.emulator_console_auth_token'
OK
auth TusJBVK6oQtn99Hm
Android Console: type 'help' for a list of commands
OK
sms send 666777888 "hola! quedamos ahora?"
OK
```