uc3m | Universidad Carlos III de Madrid

Grupo de investigación: Computer Security Lab

Mobile Devices Security

Degree in Computer Engineering

2019

Agenda

- Platform Installation - Android Studio
- Creating an Android application
  - Create the project
  - manifest.xml
  - Inicio.java
  - emulator
  - creating layout
  - Adding Activity
  - Add music
  - Use 2 Activities and add timer

Android Studio

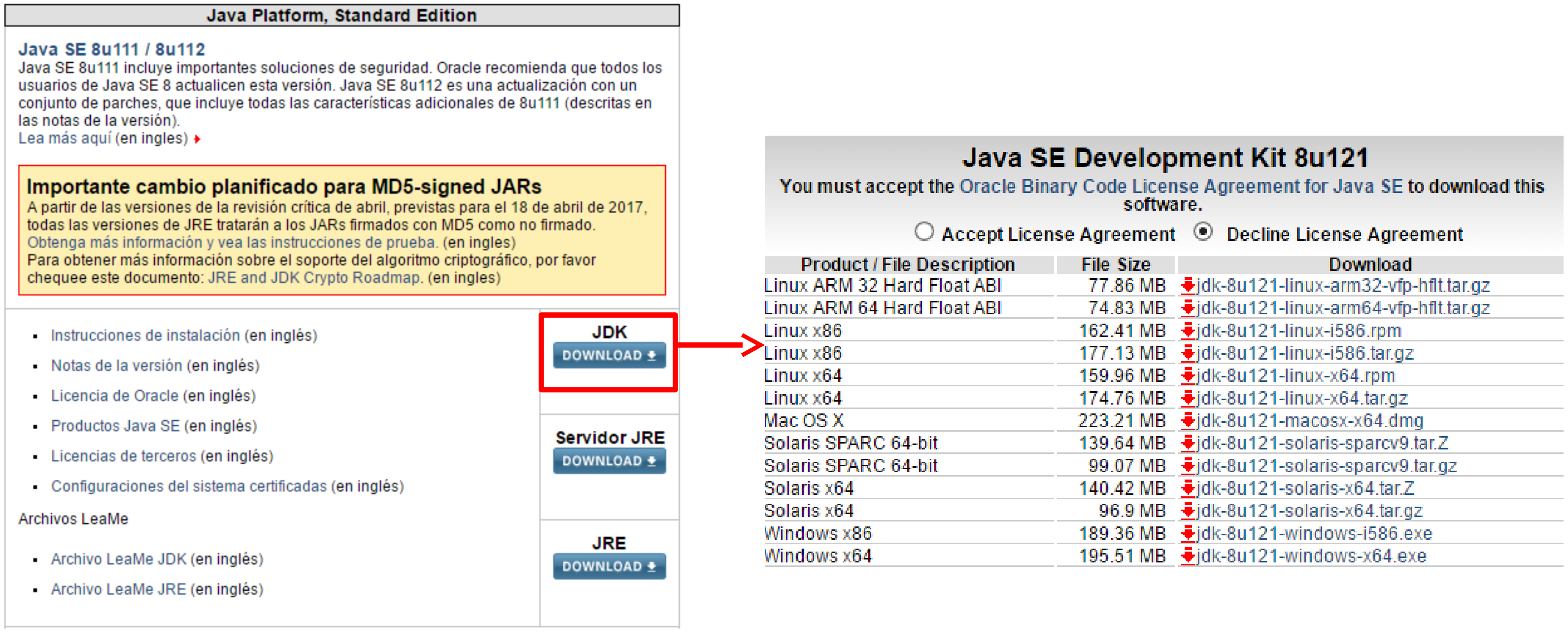
Android Studio Platform Installation

Introduction

- Download and install the Java JDK
- Install Android Studio
- Configure the SDK
- Create an Android Project

Install Java JDK

- http://www.oracle.com/technetwork/es/java/javase/downloads/index.html



Java Platform, Standard Edition

Java SE 8u111 / 8u112
Java SE 8u111 incluye importantes soluciones de seguridad. Oracle recomienda que todos los usuarios de Java SE 8 actualicen esta versión. Java SE 8u112 es una actualización con un conjunto de parches, que incluye todas las características adicionales de 8u111 (descritas en las notas de la versión).
Lea más aquí (en ingles)

Importante cambio planificado para MD5-signed JARs
A partir de las versiones de la revisión crítica de abril, previstas para el 18 de abril de 2017, todas las versiones de JRE tratarán a los JARs firmados con MD5 como no firmado.
Obtenga más información y vea las instrucciones de prueba. (en ingles)
Para obtener más información sobre el soporte del algoritmo criptográfico, por favor chequee este documento: JRE and JDK Crypto Roadmap. (en ingles)

- Instrucciones de instalación (en inglés)
- Notas de la versión (en inglés)
- Licencia de Oracle (en inglés)
- Productos Java SE (en inglés)
- Licencias de terceros (en inglés)
- Configuraciones del sistema certificadas (en inglés)

Archivos LeaMe

- Archivo LeaMe JDK (en inglés)
- Archivo LeaMe JRE (en inglés)

Java SE Development Kit 8u121
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement [✓ Decline License Agreement]

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.86 MB	jdk-8u121-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u121-linux-arm64-vfp-hflt.tar.gz
Linux x86	162.41 MB	jdk-8u121-linux-i586.rpm
Linux x86	177.13 MB	jdk-8u121-linux-i586.tar.gz
Linux x64	159.96 MB	jdk-8u121-linux-x64.rpm
Linux x64	174.76 MB	jdk-8u121-linux-x64.tar.gz
Mac OS X	223.21 MB	jdk-8u121-macosx-x64.dmg
Solaris SPARC 64-bit	139.64 MB	jdk-8u121-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.07 MB	jdk-8u121-solaris-sparcv9.tar.gz
Solaris x64	140.42 MB	jdk-8u121-solaris-x64.tar.Z
Solaris x64	96.9 MB	jdk-8u121-solaris-x64.tar.gz
Windows x86	189.36 MB	jdk-8u121-windows-i586.exe
Windows x64	195.51 MB	jdk-8u121-windows-x64.exe

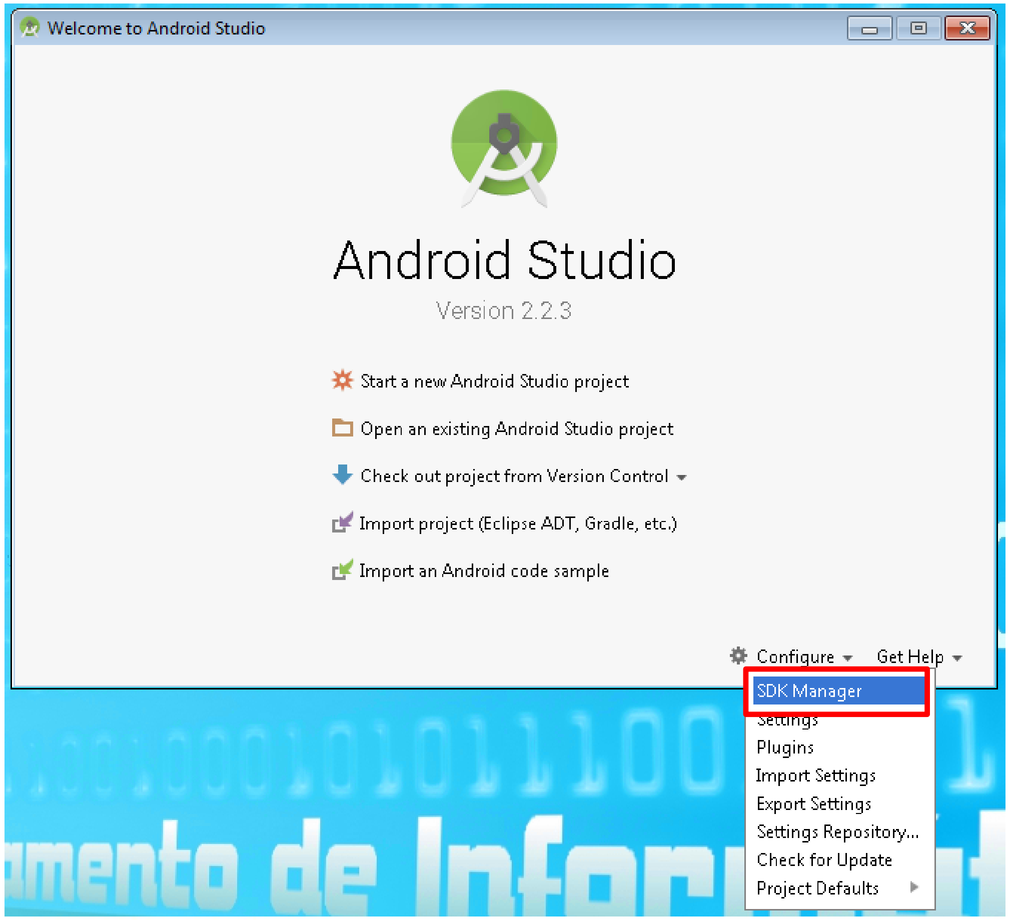
Android Studio

- http://developer.android.com/intl/es/sdk/index.html

- Minimum requirements (Windows):
  - Windows 8/7 / Vista (32/64 bit)
  - 2GB RAM
  - 400MB (IDE) + 1GB (SDK) HDD
  - 1280x800 screen resolution
  - JDK 7

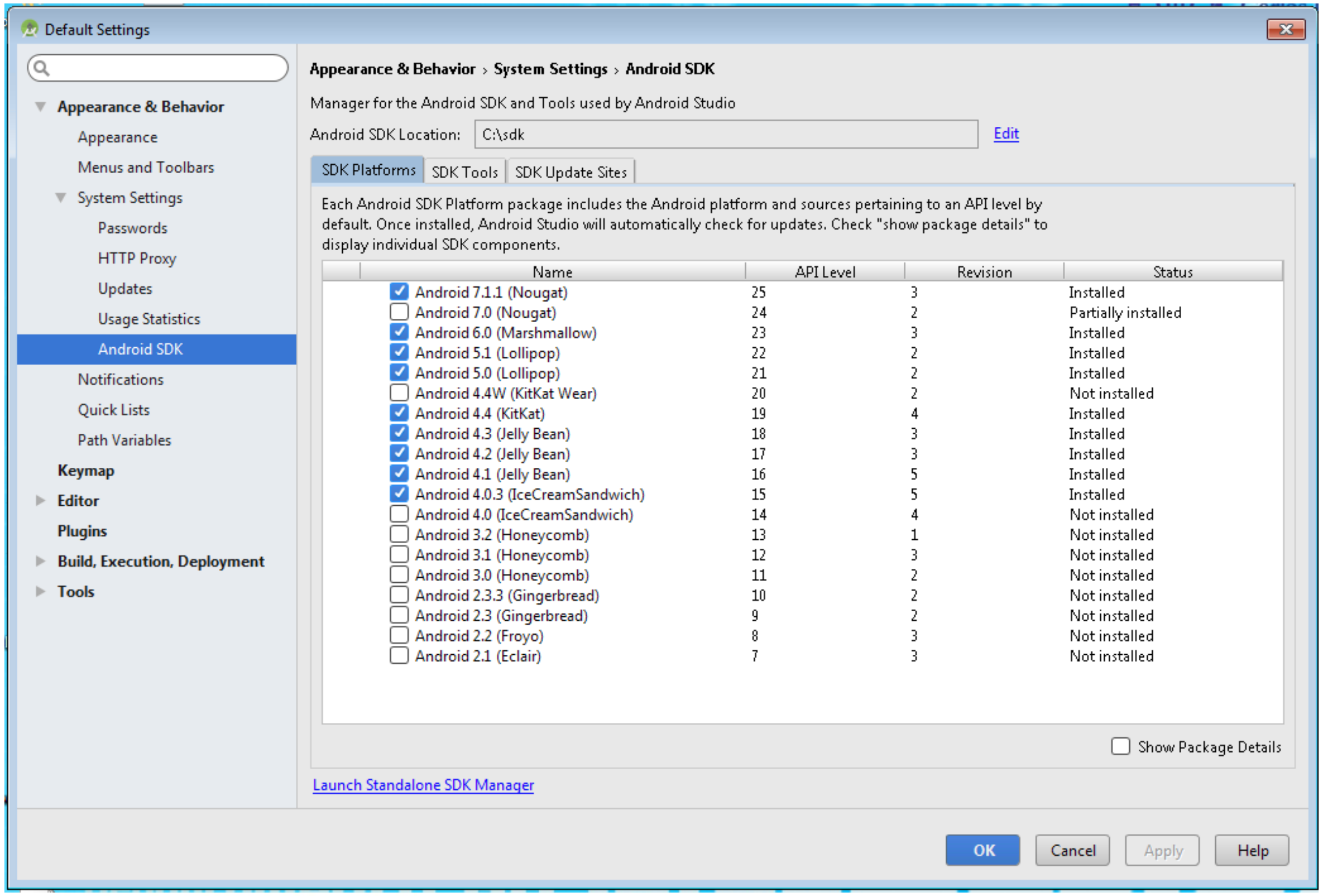
Android SDK

- We configure the SDK


Welcome to Android Studio
Android Studio
Version 2.2.3
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Get Help
SDK Manager
Settings
Plugins
Import Settings
Export Settings
Settings Repository...
Check for Update
Project Defaults

Android SDK

- Adding components to the SDK



Default Settings

- Appearance & Behavior
  - Appearance
  - Menus and Toolbars
  - System Settings
    - Passwords
    - HTTP Proxy
    - Updates
    - Usage Statistics
    - Android SDK
  - Notifications
  - Quick Lists
  - Path Variables
- Keymap
- Editor
- Plugins
- Build, Execution, Deployment
- Tools

Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\sdk Edit

SDK Platforms | SDK Tools | SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
✓	Android 7.1.1 (Nougat)	25	3	Installed
☐	Android 7.0 (Nougat)	24	2	Partially installed
✓	Android 6.0 (Marshmallow)	23	3	Installed
✓	Android 5.1 (Lollipop)	22	2	Installed
✓	Android 5.0 (Lollipop)	21	2	Installed
☐	Android 4.4W (KitKat Wear)	20	2	Not installed
✓	Android 4.4 (KitKat)	19	4	Installed
✓	Android 4.3 (Jelly Bean)	18	3	Installed
✓	Android 4.2 (Jelly Bean)	17	3	Installed
✓	Android 4.1 (Jelly Bean)	16	5	Installed
✓	Android 4.0.3 (IceCreamSandwich)	15	5	Installed
☐	Android 4.0 (IceCreamSandwich)	14	4	Not installed
☐	Android 3.2 (Honeycomb)	13	1	Not installed
☐	Android 3.1 (Honeycomb)	12	3	Not installed
☐	Android 3.0 (Honeycomb)	11	2	Not installed
☐	Android 2.3.3 (Gingerbread)	10	2	Not installed
☐	Android 2.3 (Gingerbread)	9	2	Not installed
☐	Android 2.2 (Froyo)	8	3	Not installed
☐	Android 2.1 (Eclair)	7	3	Not installed

☐ Show Package Details

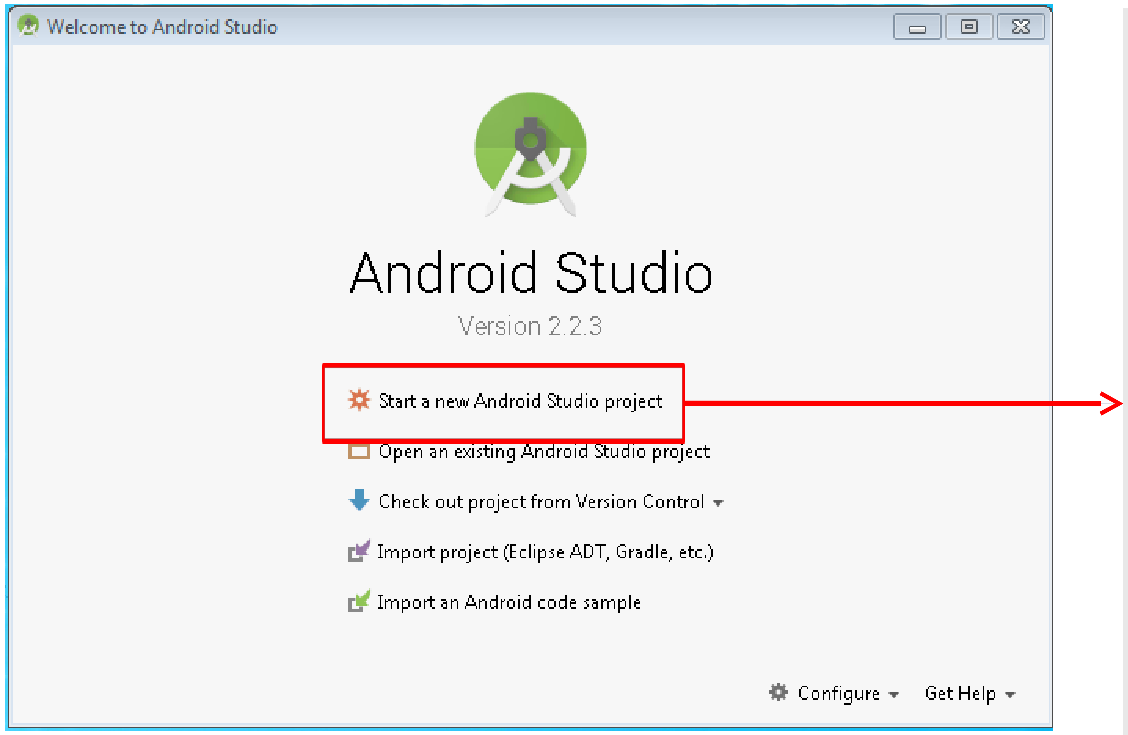
Launch Standalone SDK Manager

Android applications

Creating an Android application

Creating a Project

- We are ready to create a project



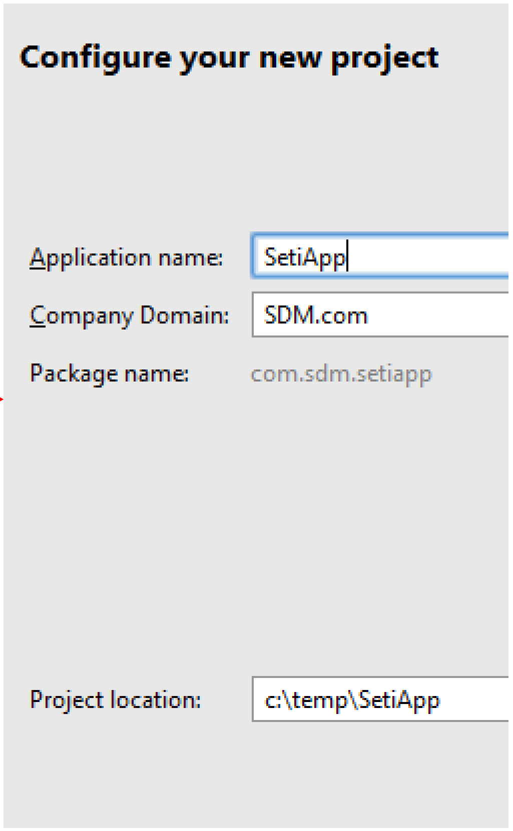
Welcome to Android Studio

Android Studio

Version 2.2.3

- Start a new Android Studio project
- Open an existing Android Studio project
- Check out project from Version Control
- Import project (Eclipse ADT, Gradle, etc.)
- Import an Android code sample

Configure   Get Help



Configure your new project

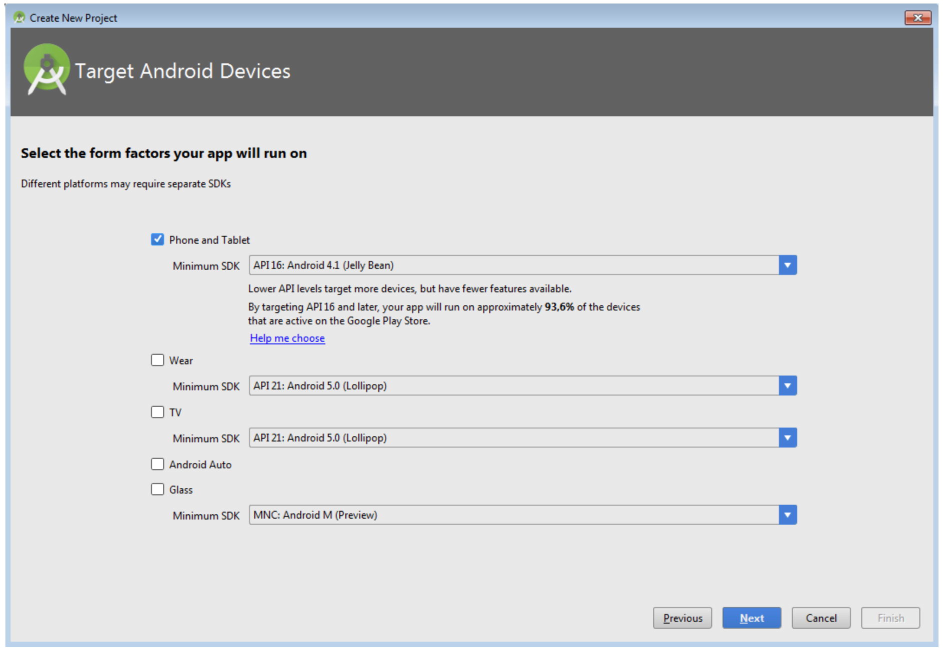
Application name:	SetiApp
Company Domain:	SDM.com
Package name:	com.sdm.setiapp
Project location:	c:\temp\SetiApp

Creating a Project

Application name: the name displayed on the mobile

Company Domain: Company name, used to generate the package name

Package name: Package name (unique)



Select the form factors your app will run on

Different platforms may require separate SDKs

[✓] Phone and Tablet
Minimum SDK: API 16: Android 4.1 (Jelly Bean)
Lower API levels target more devices, but have fewer features available.
By targeting API 16 and later, your app will run on approximately **93,6%** of the devices that are active on the Google Play Store.
Help me choose

[ ] Wear
Minimum SDK: API 21: Android 5.0 (Lollipop)

[ ] TV
Minimum SDK: API 21: Android 5.0 (Lollipop)

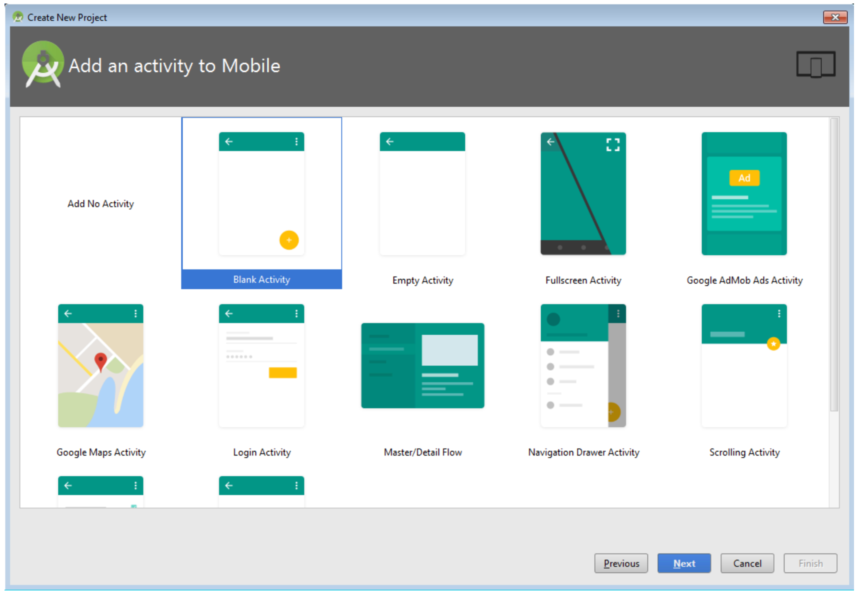
[ ] Android Auto

[ ] Glass
Minimum SDK: MNC: Android M (Preview)

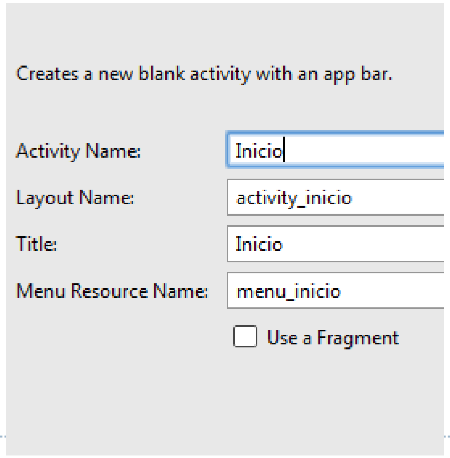
We choose the version on which we will run the application: Android 4.1 API 16 (Jelly Bean), Click on "Next"

Creating a Project

- Create a blank activity, we click "Next"



- We choose the name of our activity: Start and click on "Finish"

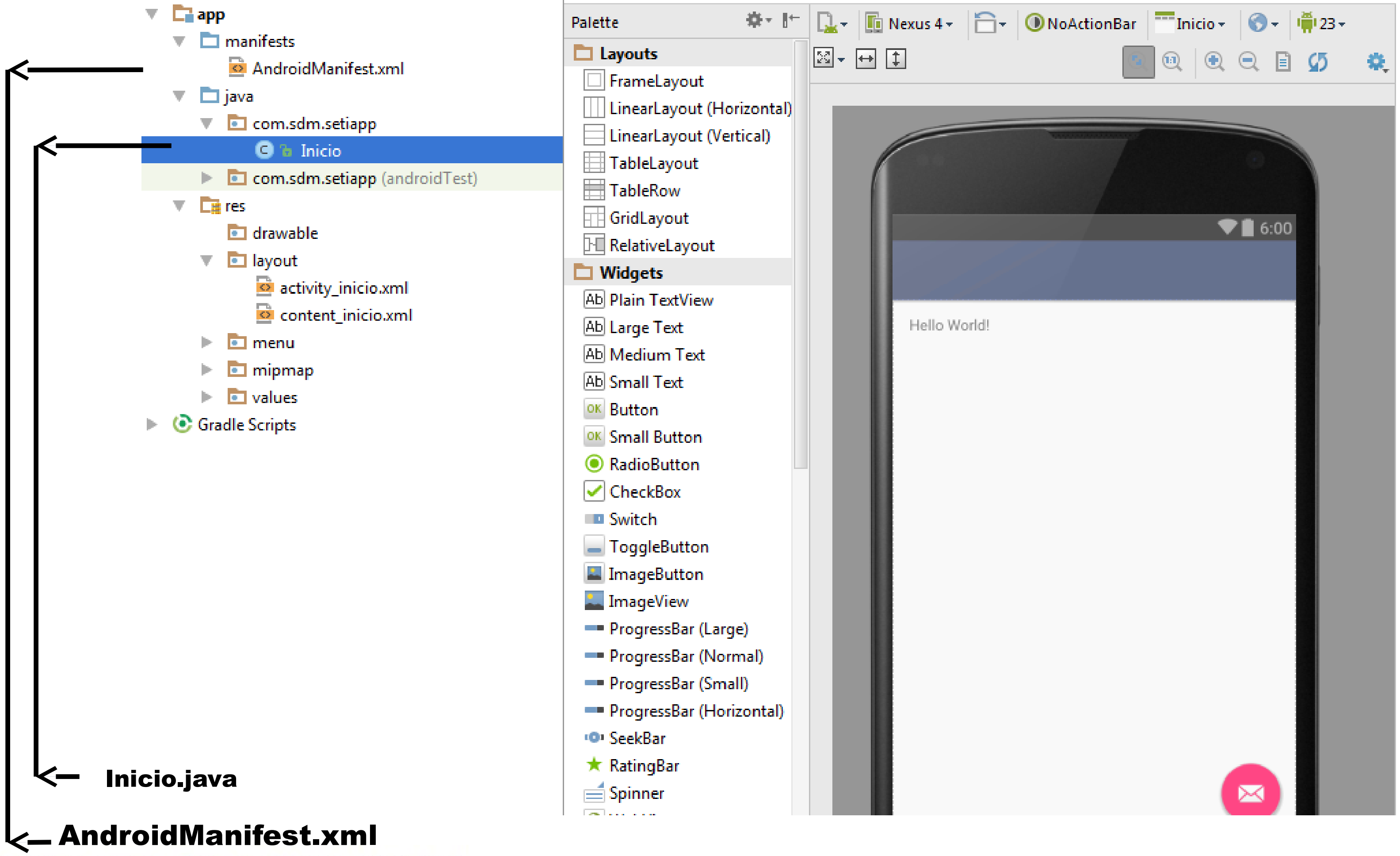


Creates a new blank activity with an app bar.

Activity Name:	Inicio
Layout Name:	activity_inicio
Title:	Inicio
Menu Resource Name:	menu_inicio
	☐ Use a Fragment

Creating a Project

- We have created our project:



app
manifests
AndroidManifest.xml
java
com.sdm.setiapp
Inicio
com.sdm.setiapp (androidTest)
res
drawable
layout
activity_inicio.xml
content_inicio.xml
menu
mipmap
values
Gradle Scripts

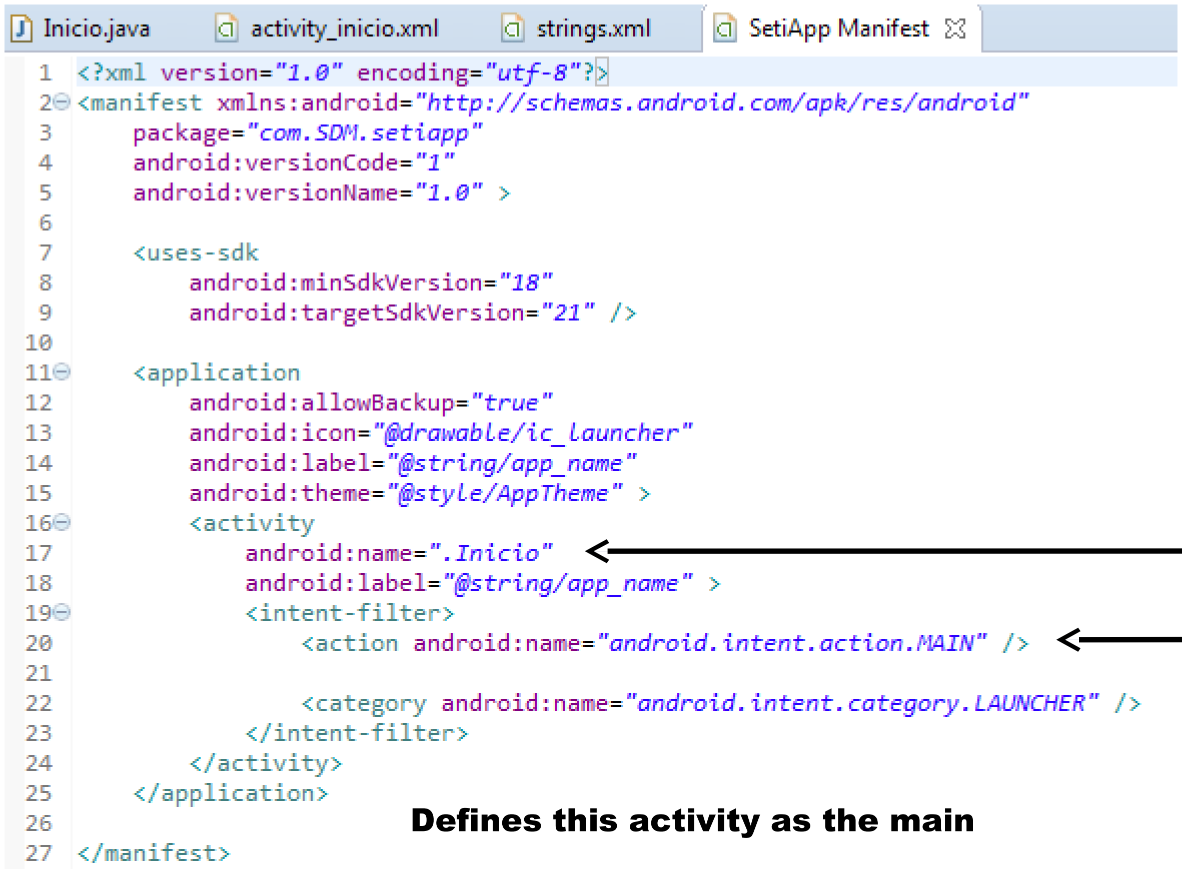
Palette
Layouts
FrameLayout
LinearLayout (Horizontal)
LinearLayout (Vertical)
TableLayout
TableRow
GridLayout
RelativeLayout
Widgets
Plain TextView
Large Text
Medium Text
Small Text
Button
Small Button
RadioButton
CheckBox
Switch
ToggleButton
ImageButton
ImageView
ProgressBar (Large)
ProgressBar (Normal)
ProgressBar (Small)
ProgressBar (Horizontal)
SeekBar
RatingBar
Spinner

Nexus 4 · NoActionBar · Inicio · 23

Inicio.java

AndroidManifest.xml

manifest.xml



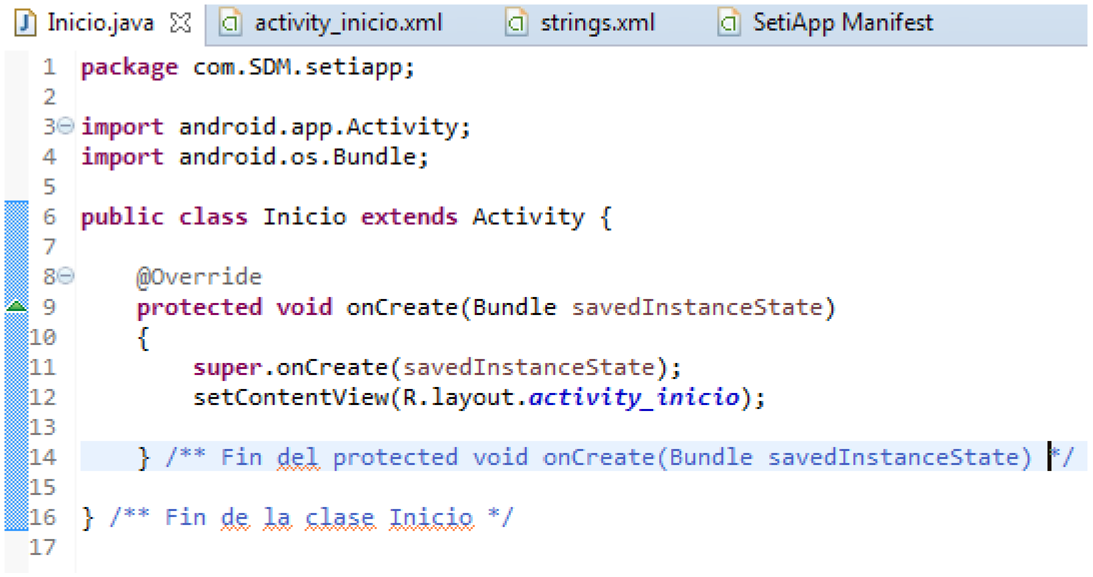
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.SDM.setiapp"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="18"
        android:targetSdkVersion="21" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".Inicio"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Defines this activity as the main

For each activity we define a class: ***public class Start extends*** **Activity {}**

Inicio.java

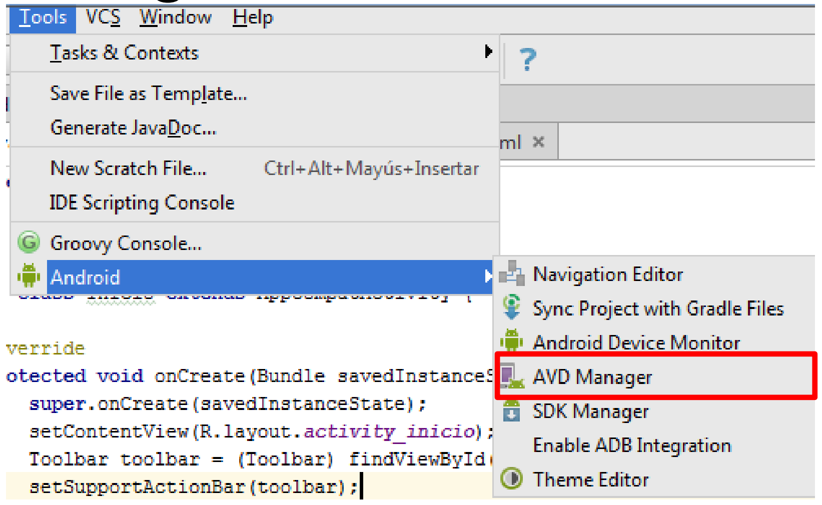
- In order to display a UI, Android Studio creates us a layout file type .xml which is what we see in mobile.
- It also creates a logical part by a java file



```java
package com.SDM.setiapp;

import android.app.Activity;
import android.os.Bundle;

public class Inicio extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inicio);

    } /** Fin del protected void onCreate(Bundle savedInstanceState) */

} /** Fin de la clase Inicio */
```

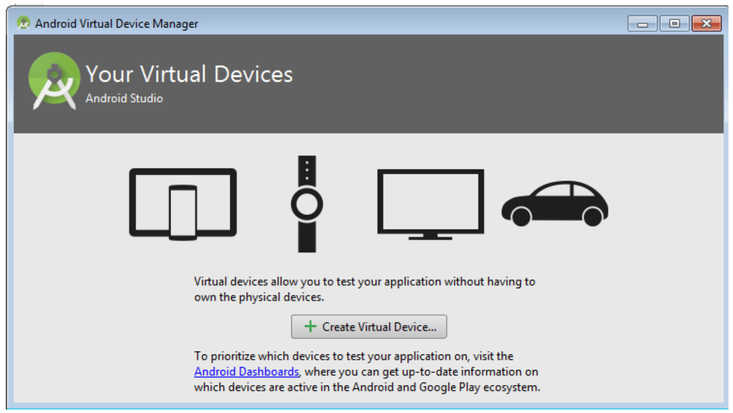
Emulator

- Tools> Android> AVD Manager



```
verride
otected void onCreate(Bundle savedInstanceS
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_inicio);
  Toolbar toolbar = (Toolbar) findViewById(
  setSupportActionBar(toolbar);
```

- Create a new AVD

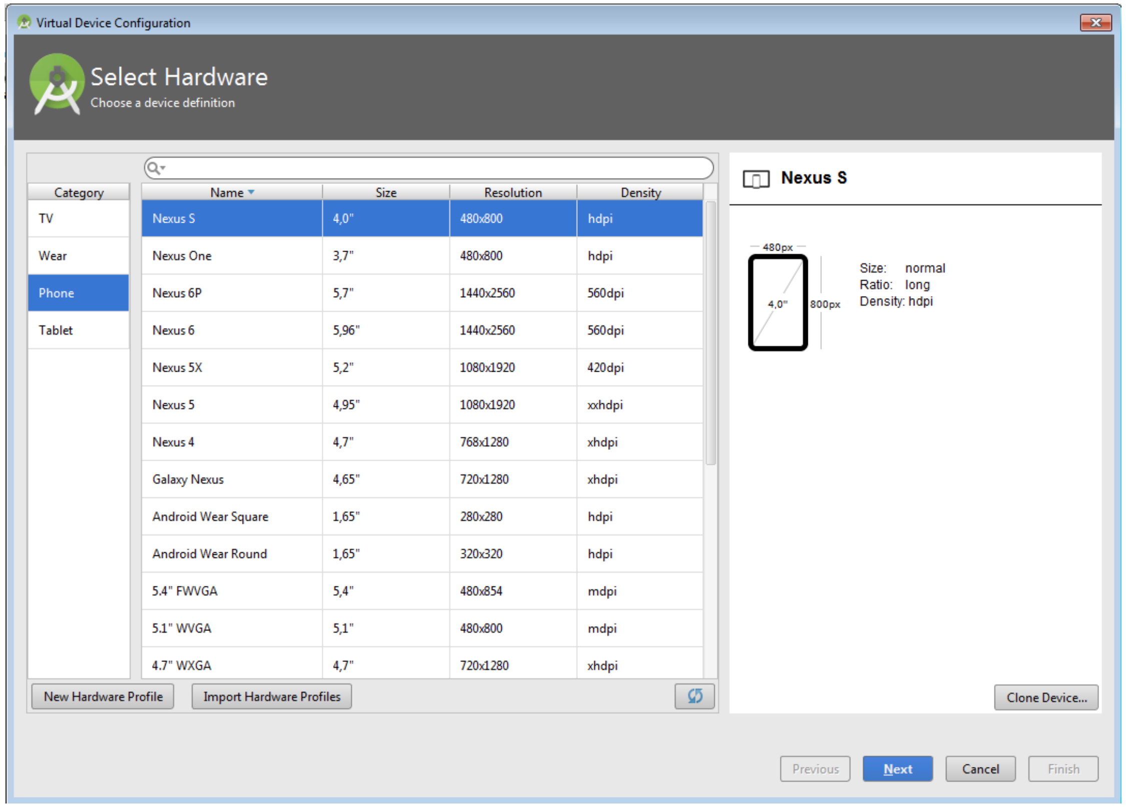


Virtual devices allow you to test your application without having to own the physical devices.

To prioritize which devices to test your application on, visit the Android Dashboards, where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.

Emulator

- Choose the type of phone that will hold the emulation



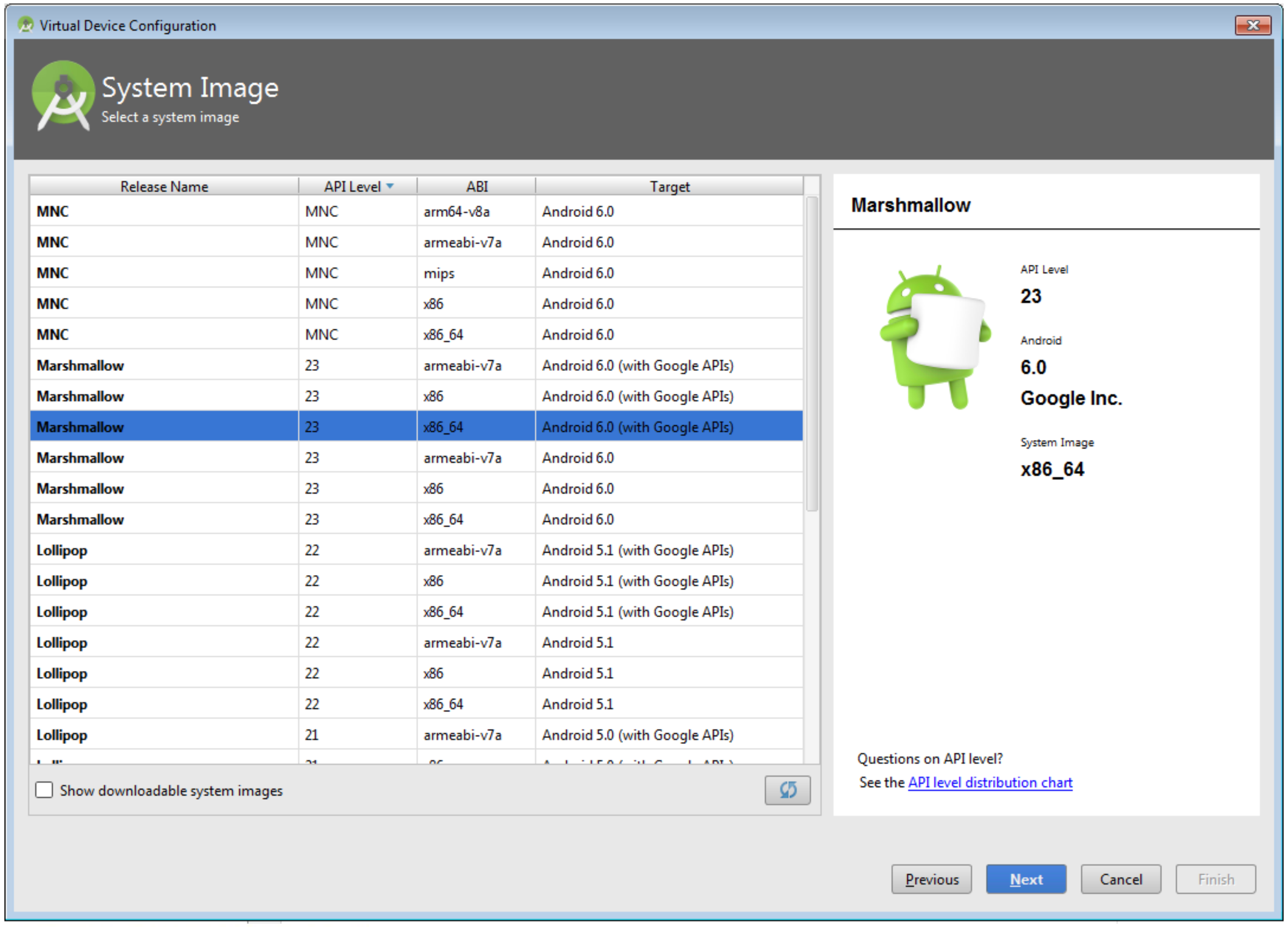
Category
TV
Wear
Phone
Tablet

Name	Size	Resolution	Density
Nexus S	4,0"	480x800	hdpi
Nexus One	3,7"	480x800	hdpi
Nexus 6P	5,7"	1440x2560	560dpi
Nexus 6	5,96"	1440x2560	560dpi
Nexus 5X	5,2"	1080x1920	420dpi
Nexus 5	4,95"	1080x1920	xxhdpi
Nexus 4	4,7"	768x1280	xhdpi
Galaxy Nexus	4,65"	720x1280	xhdpi
Android Wear Square	1,65"	280x280	hdpi
Android Wear Round	1,65"	320x320	hdpi
5.4" FWVGA	5,4"	480x854	mdpi
5.1" WVGA	5,1"	480x800	mdpi
4.7" WXGA	4,7"	720x1280	xhdpi

Size: normal
Ratio: long
Density: hdpi

Emulator

- Select the system image



Virtual Device Configuration

System Image
Select a system image

Release Name	API Level	ABI	Target
MNC	MNC	arm64-v8a	Android 6.0
MNC	MNC	armeabi-v7a	Android 6.0
MNC	MNC	mips	Android 6.0
MNC	MNC	x86	Android 6.0
MNC	MNC	x86_64	Android 6.0
Marshmallow	23	armeabi-v7a	Android 6.0 (with Google APIs)
Marshmallow	23	x86	Android 6.0 (with Google APIs)
Marshmallow	23	x86_64	Android 6.0 (with Google APIs)
Marshmallow	23	armeabi-v7a	Android 6.0
Marshmallow	23	x86	Android 6.0
Marshmallow	23	x86_64	Android 6.0
Lollipop	22	armeabi-v7a	Android 5.1 (with Google APIs)
Lollipop	22	x86	Android 5.1 (with Google APIs)
Lollipop	22	x86_64	Android 5.1 (with Google APIs)
Lollipop	22	armeabi-v7a	Android 5.1
Lollipop	22	x86	Android 5.1
Lollipop	22	x86_64	Android 5.1
Lollipop	21	armeabi-v7a	Android 5.0 (with Google APIs)

☐ Show downloadable system images

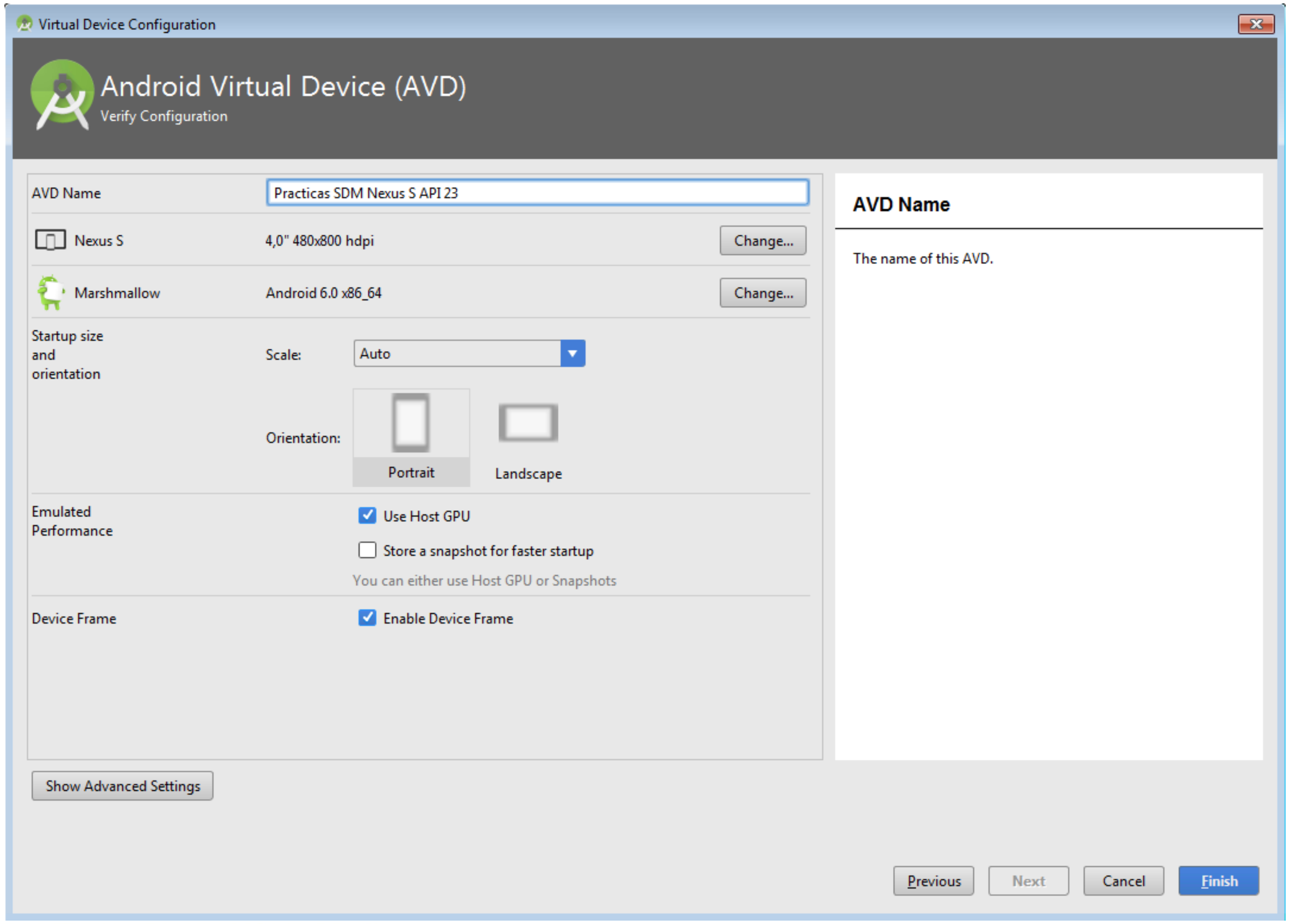
Marshmallow

API Level
23
Android
6.0
Google Inc.
System Image
x86_64

Questions on API level?
See the API level distribution chart

Emulator

- We completed the configuration assigning a name to the device



AVD Name: Practicas SDM Nexus S API 23

Nexus S — 4,0" 480x800 hdpi — Change...

Marshmallow — Android 6.0 x86_64 — Change...

Startup size and orientation — Scale: Auto — Orientation: Portrait / Landscape

Emulated Performance — [✓] Use Host GPU — [ ] Store a snapshot for faster startup

You can either use Host GPU or Snapshots

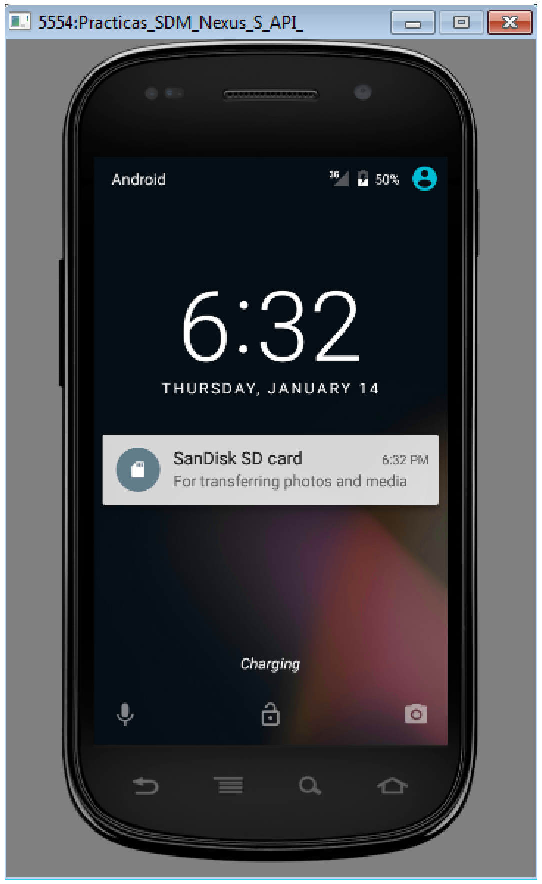
Device Frame — [✓] Enable Device Frame

AVD Name

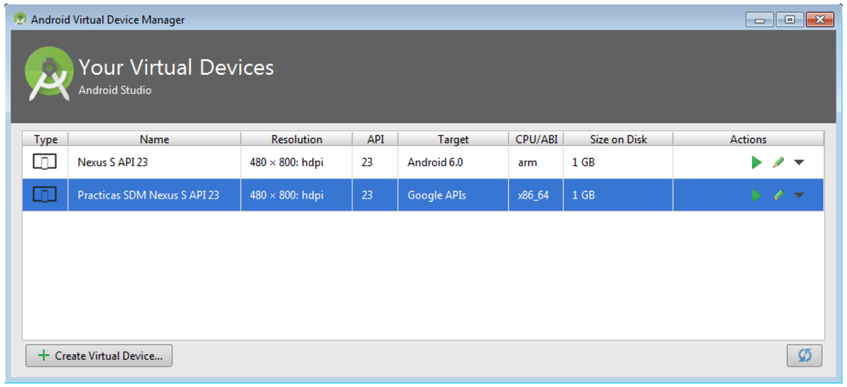
The name of this AVD.

Emulator

- We started the AVD



SanDisk SD card 6:32 PM
For transferring photos and media



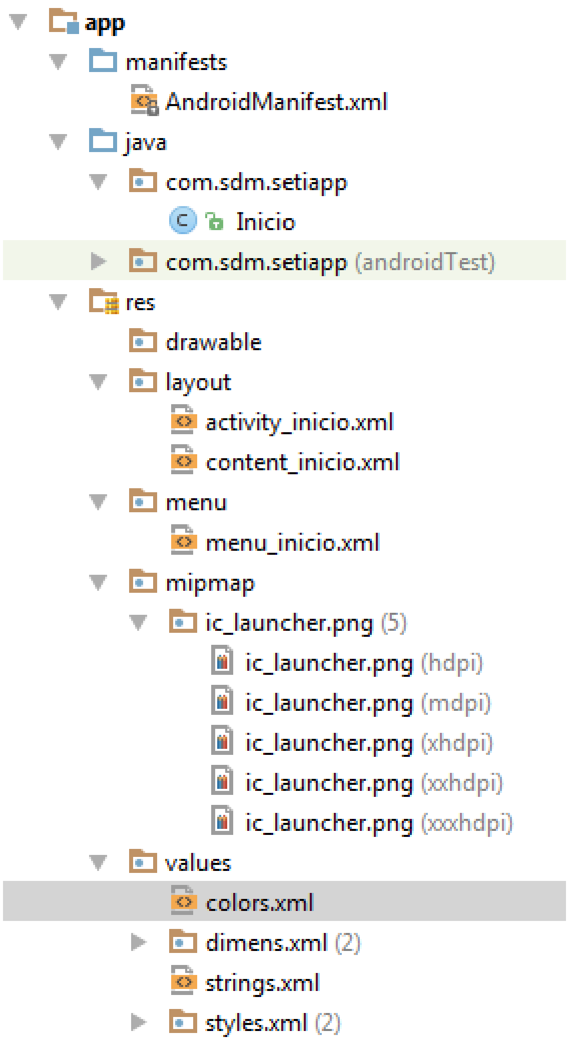
Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus S API 23	480 × 800: hdpi	23	Android 6.0	arm	1 GB	
	Practicas SDM Nexus S API 23	480 × 800: hdpi	23	Google APIs	x86_64	1 GB	

We can emulate all the AVDs we want !!

We are ready for a project !!

Project structure

- Project Explorer
  - Java
    - com.sdm.setiapp
- Folder res: Where our resources are saved
  - layout
    - activity_inicio.xml
  - mipmap
    - App icons
  - values
    - Used to define constant values used by the app

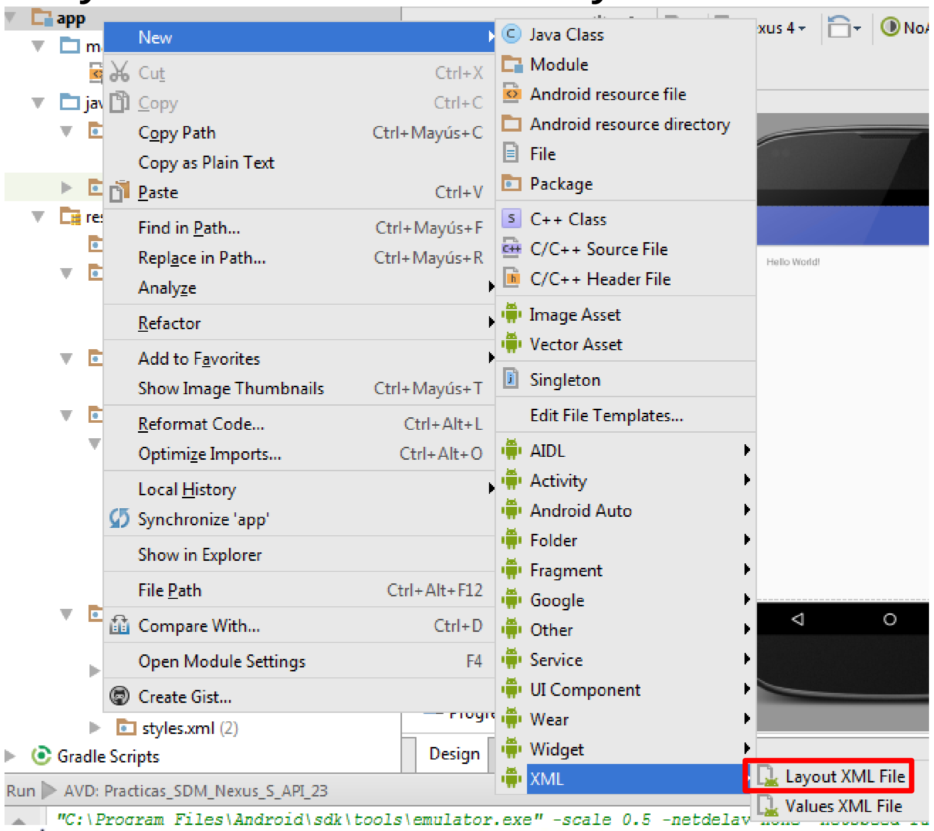


```
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▼ com.sdm.setiapp
        Inicio
    ▶ com.sdm.setiapp (androidTest)
  ▼ res
      drawable
    ▼ layout
        activity_inicio.xml
        content_inicio.xml
    ▼ menu
        menu_inicio.xml
    ▼ mipmap
      ▼ ic_launcher.png (5)
          ic_launcher.png (hdpi)
          ic_launcher.png (mdpi)
          ic_launcher.png (xhdpi)
          ic_launcher.png (xxhdpi)
          ic_launcher.png (xxxhdpi)
    ▼ values
        colors.xml
      ▶ dimens.xml (2)
        strings.xml
      ▶ styles.xml (2)
```

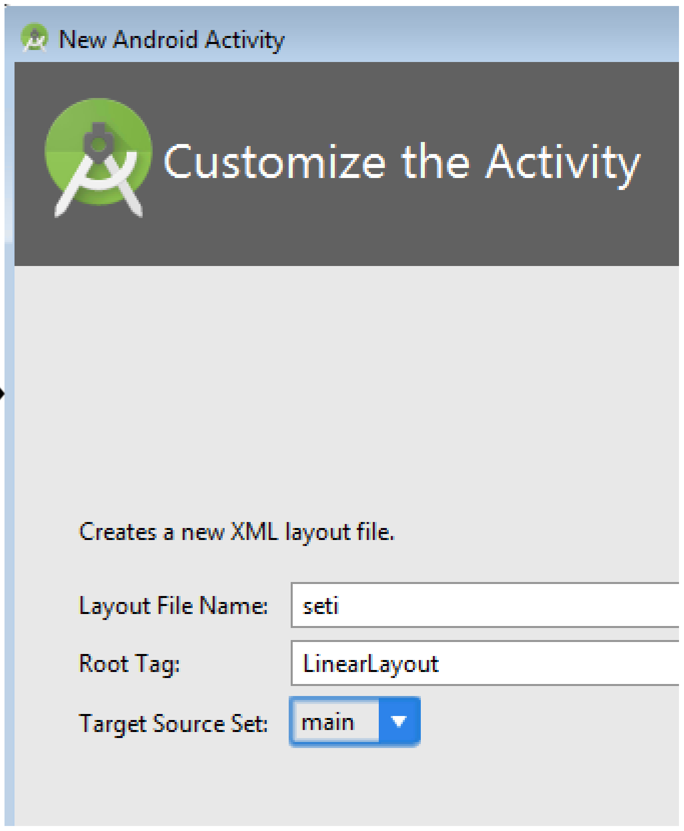
Creating a layout

We modified our project "SetiApp"We change the background displayed in the application, then:

layout > New> XML>layout XML File



```
"C:\Program Files\Android\sdk\tools\emulator.exe" -scale 0.5 -netdelay
```



Creates a new XML layout file.

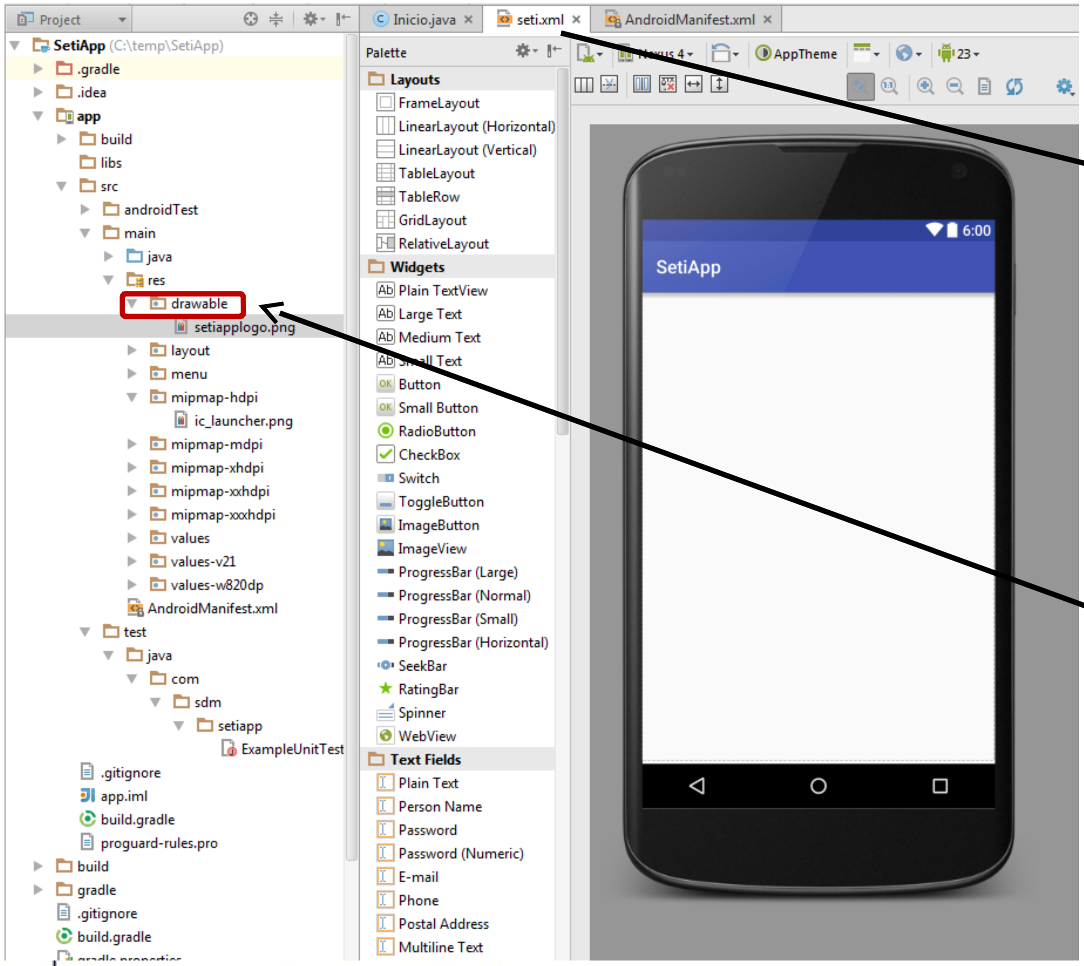
Layout File Name: seti

Root Tag: LinearLayout

Target Source Set: main

Linear layout

- To the new layout we put a logo SeTiApp of welcoming

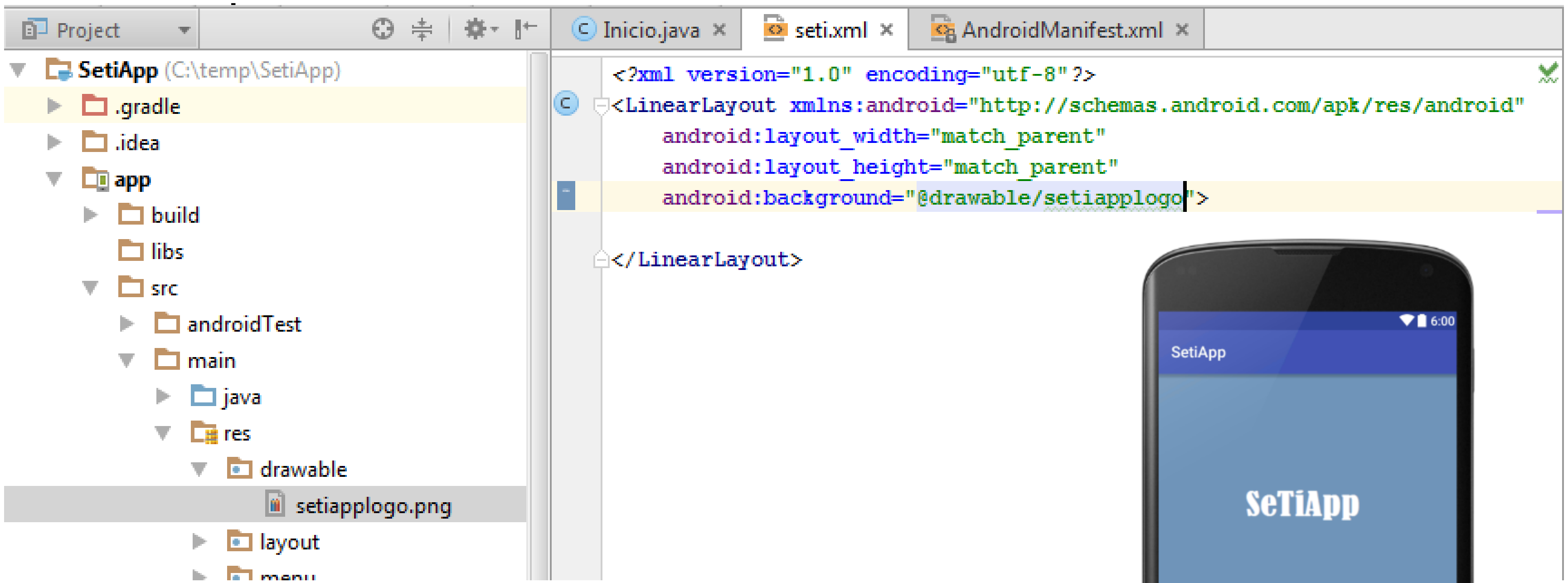


```
<?xml version="1.0" encoding="Utf-8"?>
<LinearLayout xmlns: android =
"http://schemas.android.com/apk/beef/an
droid"
    android: layout_width="match_parent"
    android: layout_height="match_parent"
    android: orientation="vertical" >

</LinearLayout>
```

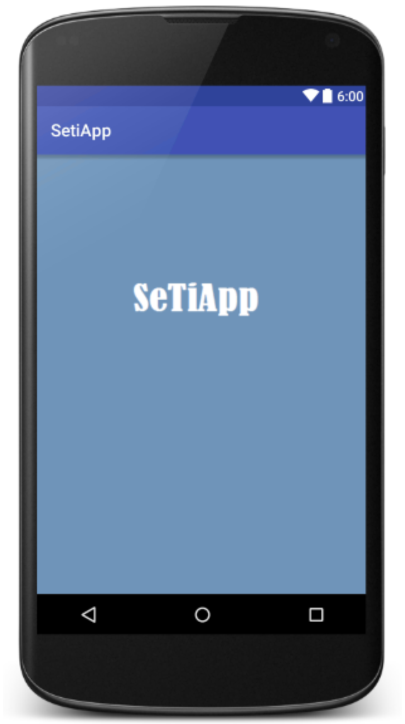
We keep.png in res/drawable

linear layout

- Seti.xml defined in the background *@drawable*: Lowercase without



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/setiapplogo">

</LinearLayout>
```



Saved and checked

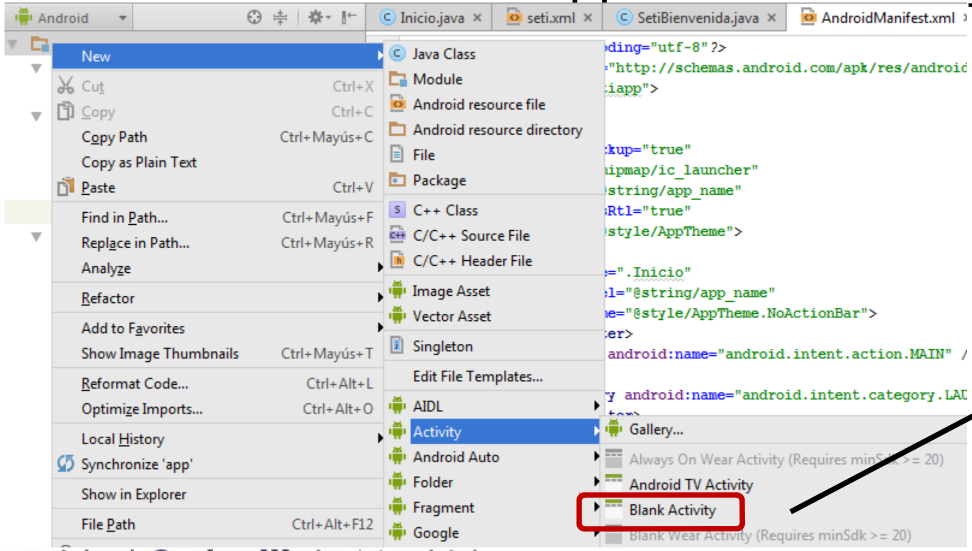
- **png (400 x 600 px)**

uc3m | Universidad Carlos III de Madrid
Grupo de investigación: Computer Security Lab

We expand our application. To do this we'll add an Activity. This activity called "SetiBienvenida" will show a welcome interface with a logo background for SetiApp. For this purpose we need to create a layout that we will call "seti.xml" and also a java class we call "SetiBienvenida.java". Finally we add music to this new presentation

Activity

- It is like a window - UI component, a screen that the user sees
- You can have more than one activity. In the current interface your device is shown only one
- The activities They are shaped by two parts:
  - logical part - .java file is the class that is created to manipulate, interact and place the code that activity
  - graphic part - XML that has all the elements we are seeing a screen
- We create a new class: app> New> Activity

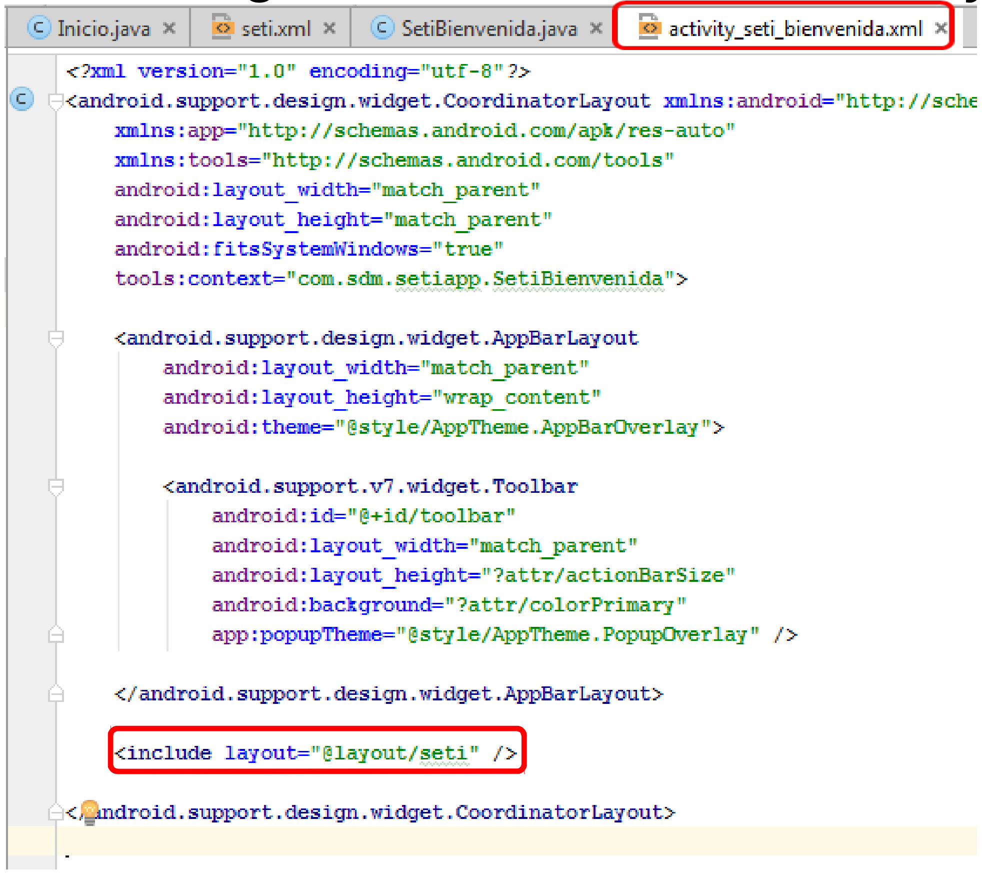


```
ding="utf-8"?>
="http://schemas.android.com/apk/res/androic
iapp">

kup="true"
ipmap/ic_launcher"
string/app_name"
Rtl="true"
style/AppTheme">

=".Inicio"
l="@string/app_name"
e="@style/AppTheme.NoActionBar">
er>
 android:name="android.intent.action.MAIN" /
y android:name="android.intent.category.LAU
```

We choose the name of the class: SetiBienvenida

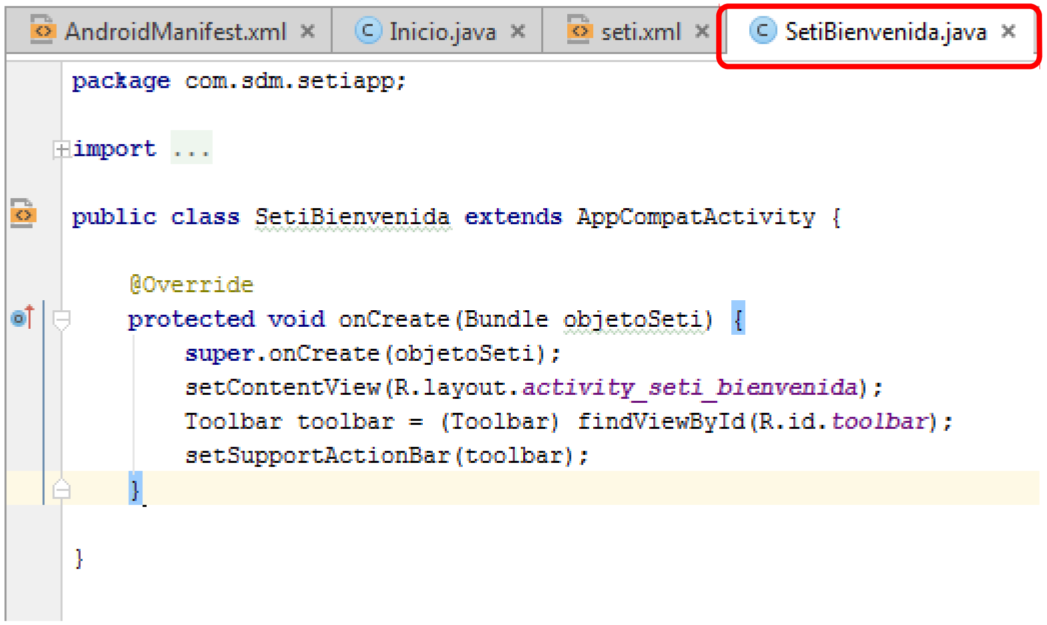
Activity

- assign the layout previously created (seti.xml) and remove the floating button that creates us by default



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://sche
    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"
    android:fitsSystemWindows="true"
    tools:context="com.sdm.setiapp.SetiBienvenida">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/seti" />

</android.support.design.widget.CoordinatorLayout>
```

Activity

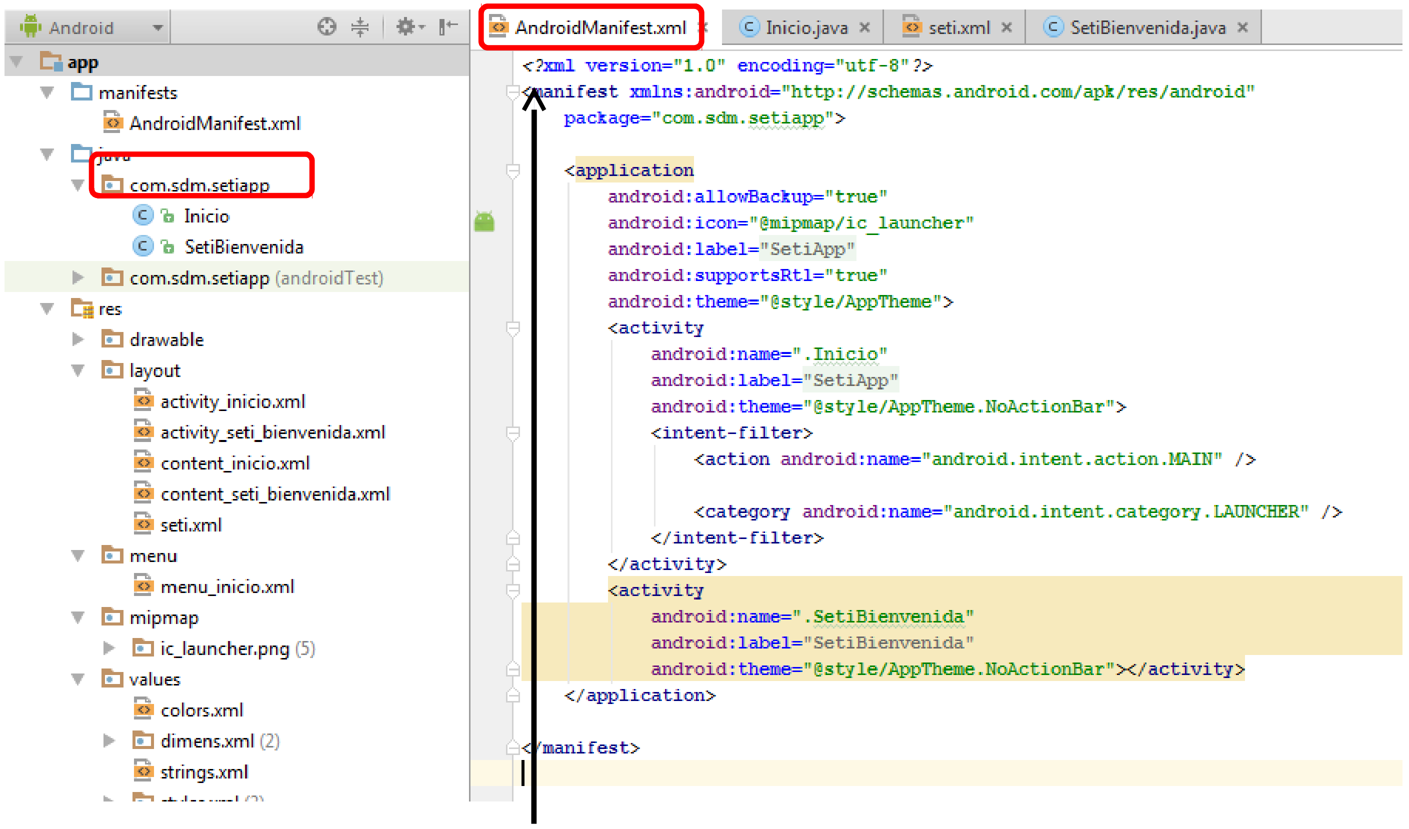


```
package com.sdm.setiapp;

import ...

public class SetiBienvenida extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle objetoSeti) {
        super.onCreate(objetoSeti);
        setContentView(R.layout.activity_seti_bienvenida);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }

}
```

- Then we configure the AndroidManifest.xml so that the activity "SetiBienvenida" is the first to execute when the application is executed

AndroidManisfest.xml

- Configuration file where you can apply the basic settings of our app

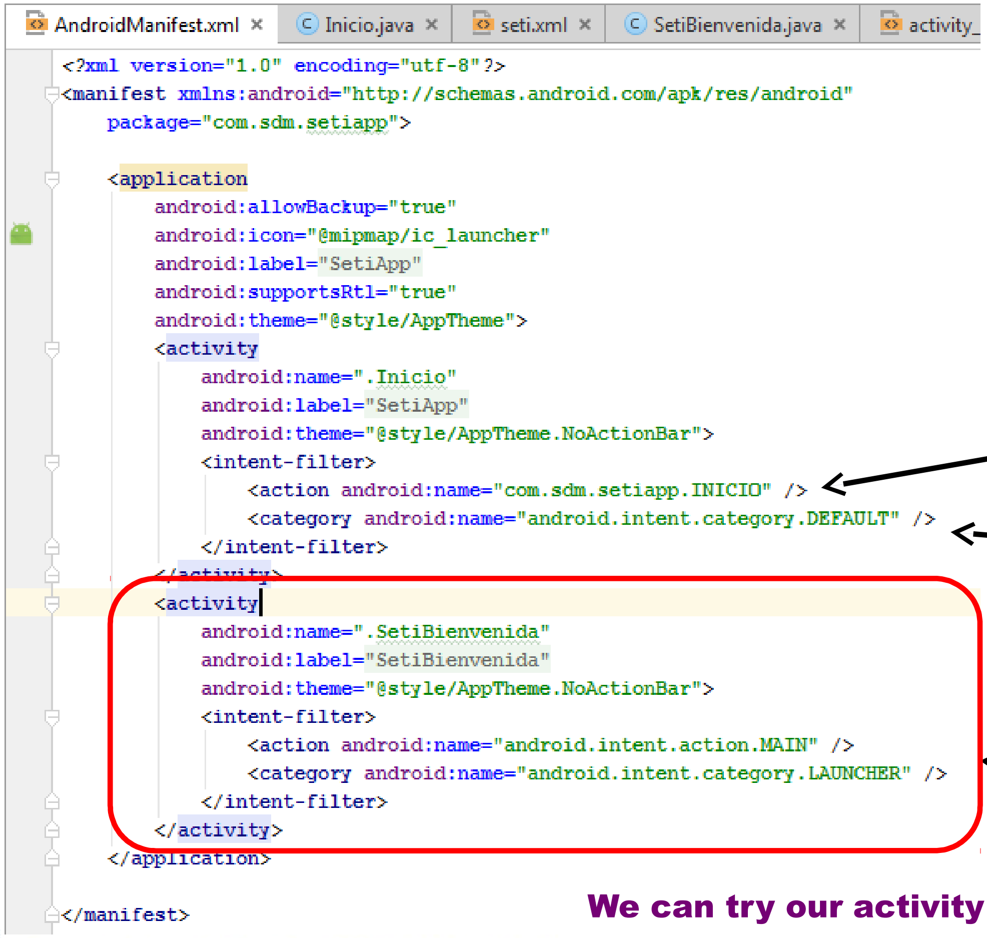


```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sdm.setiapp">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SetiApp"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".Inicio"
            android:label="SetiApp"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".SetiBienvenida"
            android:label="SetiBienvenida"
            android:theme="@style/AppTheme.NoActionBar"></activity>
    </application>

</manifest>
```

- To view the file.xml click on the tab

AndroidManisfest.xml

- add our Activity "SetiBienvenida"



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.sdm.setiapp">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SetiApp"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".Inicio"
            android:label="SetiApp"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="com.sdm.setiapp.INICIO" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity
            android:name=".SetiBienvenida"
            android:label="SetiBienvenida"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Rename the action to the package name: com.sdm.setiapp.INICIO

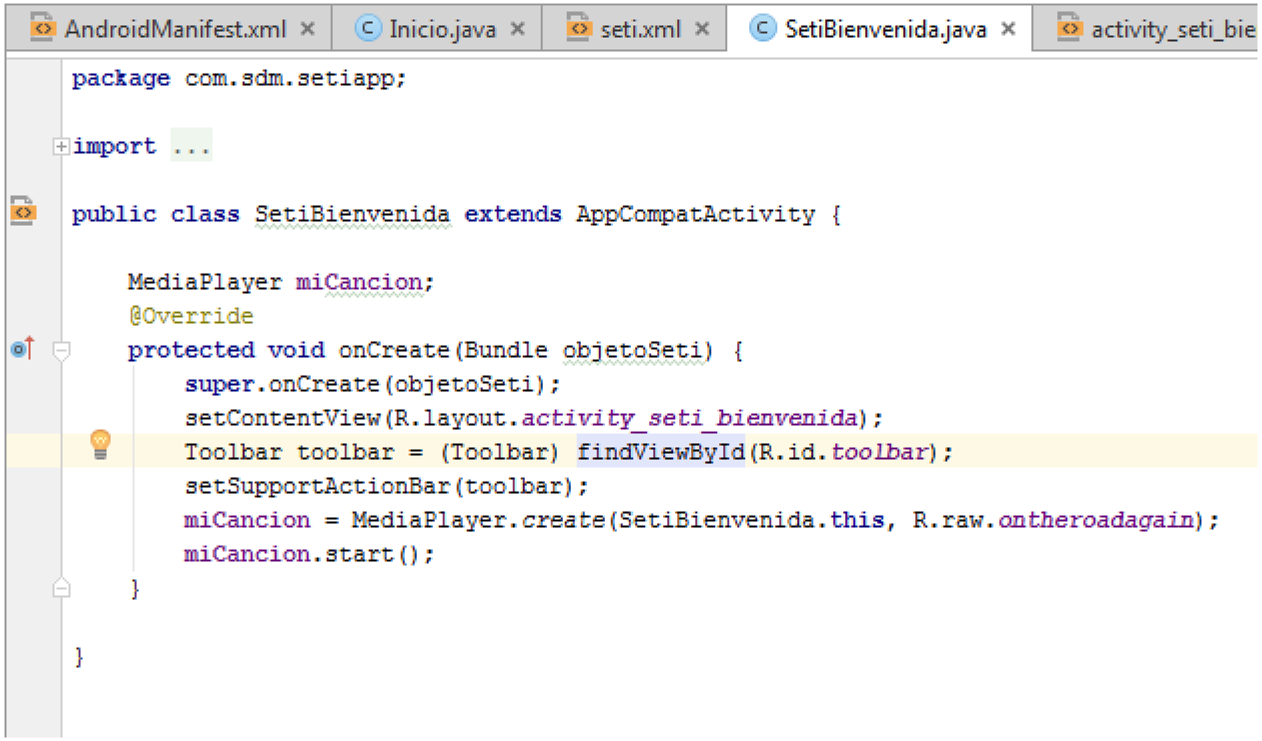
We change the category: DEFAULT

created Activity: SetiBienvenida

We can try our activity !!!

Adding music

- We want to add sound to open the app:
  - Create a folder within the call res raw
  - Save the file in the folder mp3 raw
  - Add an object MediaPlayer code



```
package com.sdm.setiapp;

import ...

public class SetiBienvenida extends AppCompatActivity {

    MediaPlayer miCancion;
    @Override
    protected void onCreate(Bundle objetoSeti) {
        super.onCreate(objetoSeti);
        setContentView(R.layout.activity_seti_bienvenida);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        miCancion = MediaPlayer.create(SetiBienvenida.this, R.raw.ontheroadagain);
        miCancion.start();
    }

}
```

Our app

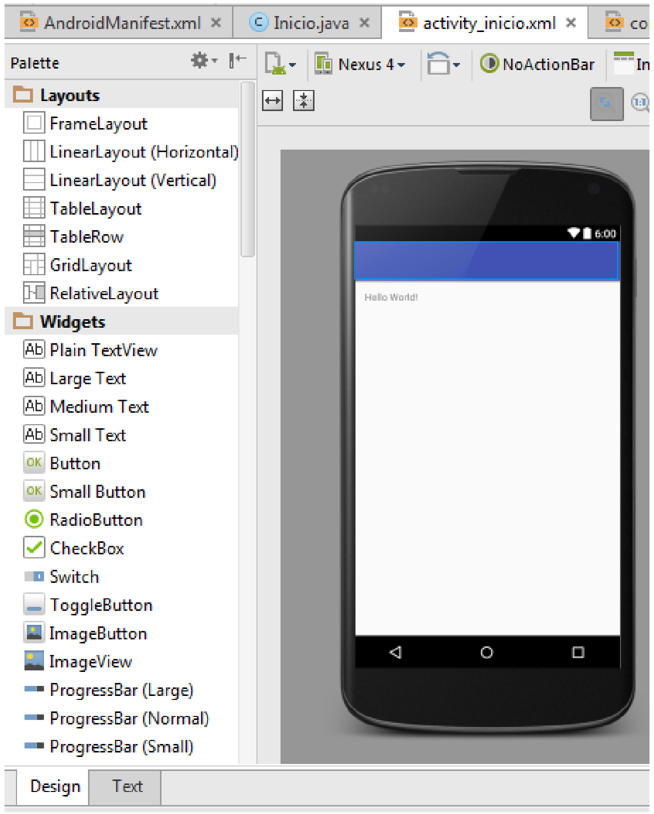
Thoughts:

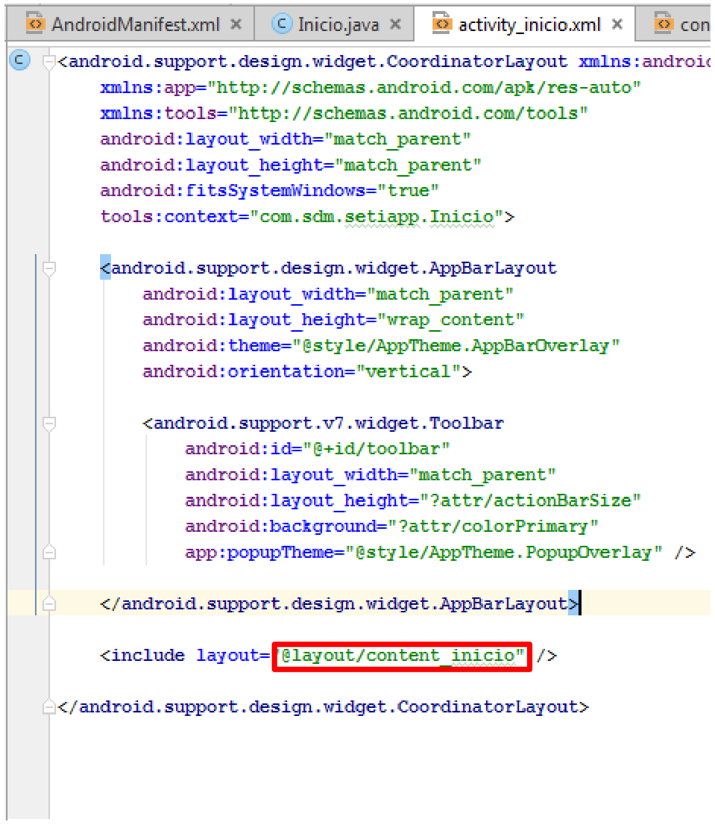
1. The initial IU (Inicio.java) not used
2. What about the song: so we want to happen?
3. How much memory use right now?
4. What if at this time someone calls you?
5. Modify the application to use the initial activity

The application starts with the song and logo SetiApp.After 9 seconds will pass a second interface.
This interface is a text that will take a count. Two buttons are displayed. One counter will increase by one, the other one decreases. Where do we start?

activity_inicio.xml

- Package Explorer > activity_inicio.xml
  - **Graphical layout**
  - **.xml layout**

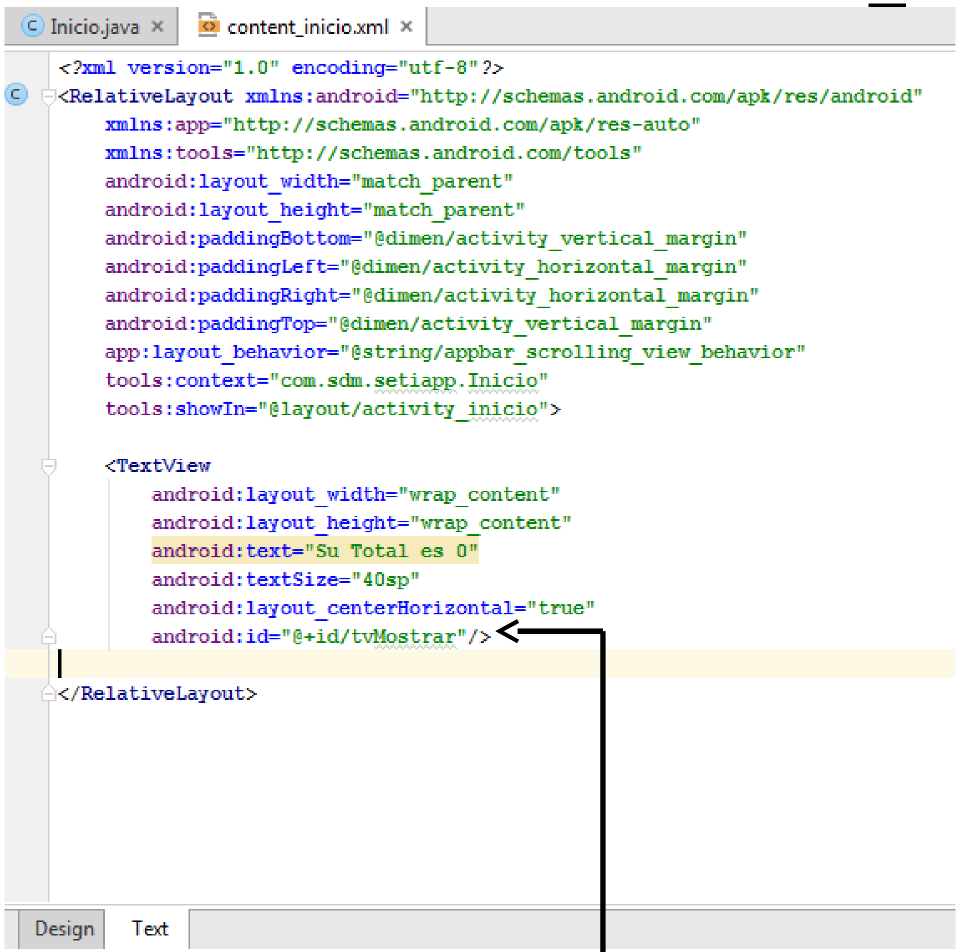




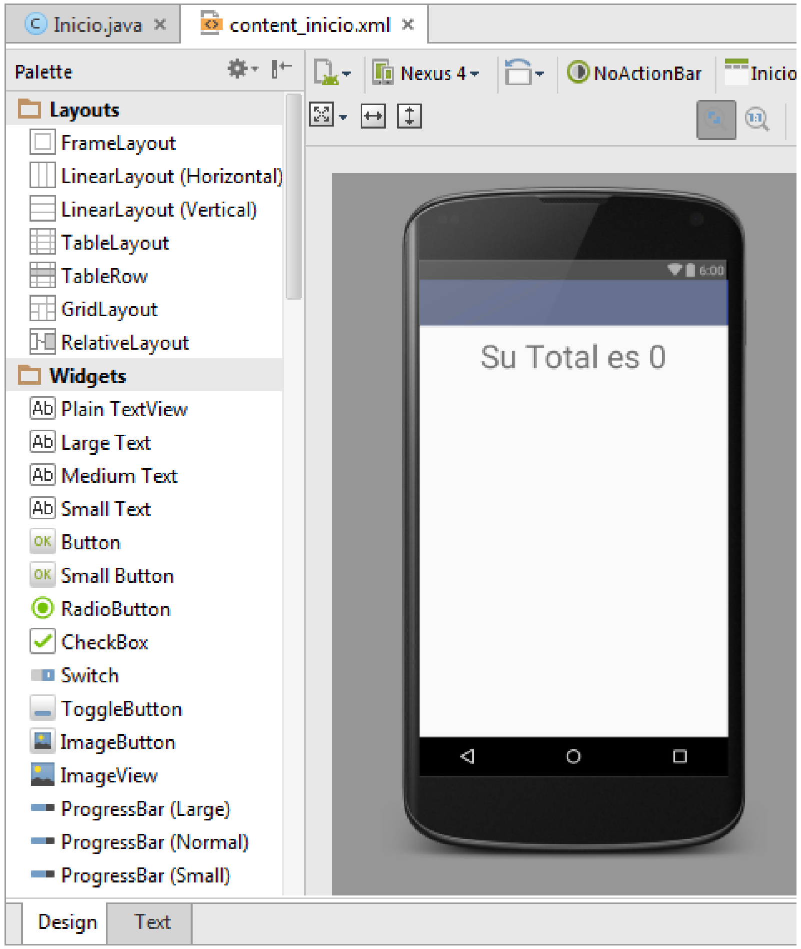
```xml
<android.support.design.widget.CoordinatorLayout xmlns:androic
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="com.sdm.setiapp.Inicio">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay"
        android:orientation="vertical">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_inicio" />

</android.support.design.widget.CoordinatorLayout>
```

Changing the app

- added a TextView in content_inicio.xml:



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.sdm.setiapp.Inicio"
    tools:showIn="@layout/activity_inicio">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Su Total es 0"
        android:textSize="40sp"
        android:layout_centerHorizontal="true"
        android:id="@+id/tvMostrar"/>

</RelativeLayout>
```

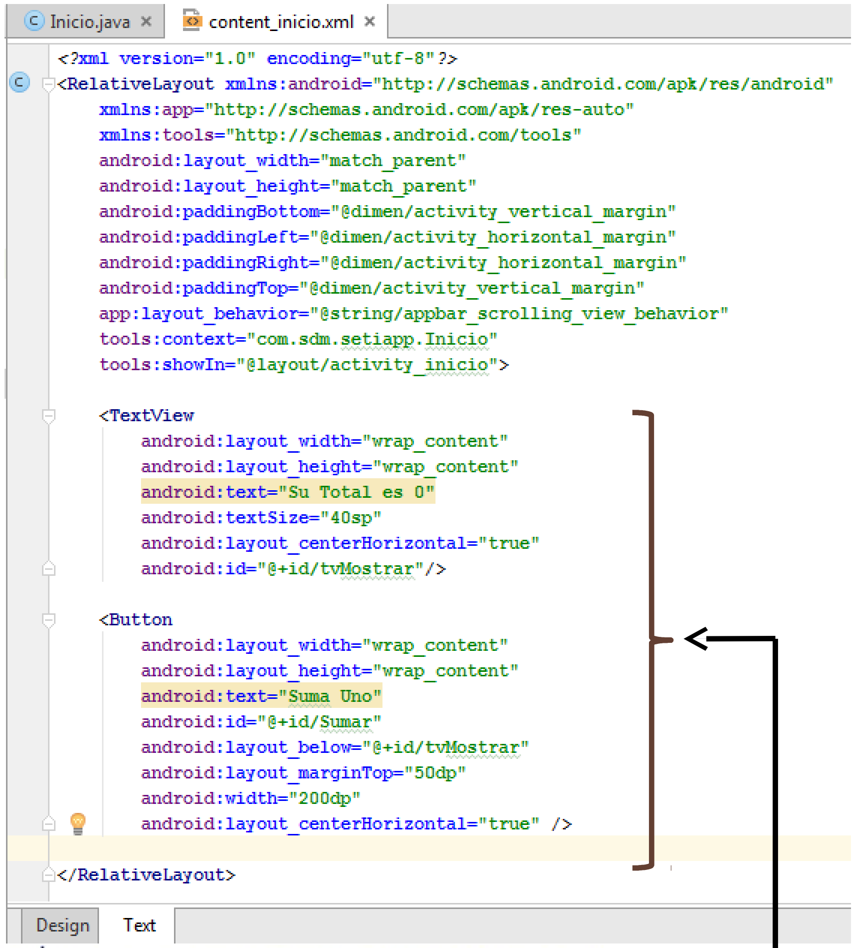


Reference for use in java

uc3m Universidad Carlos III de Madrid
Grupo de investigación: Computer Security Lab

Changing the app

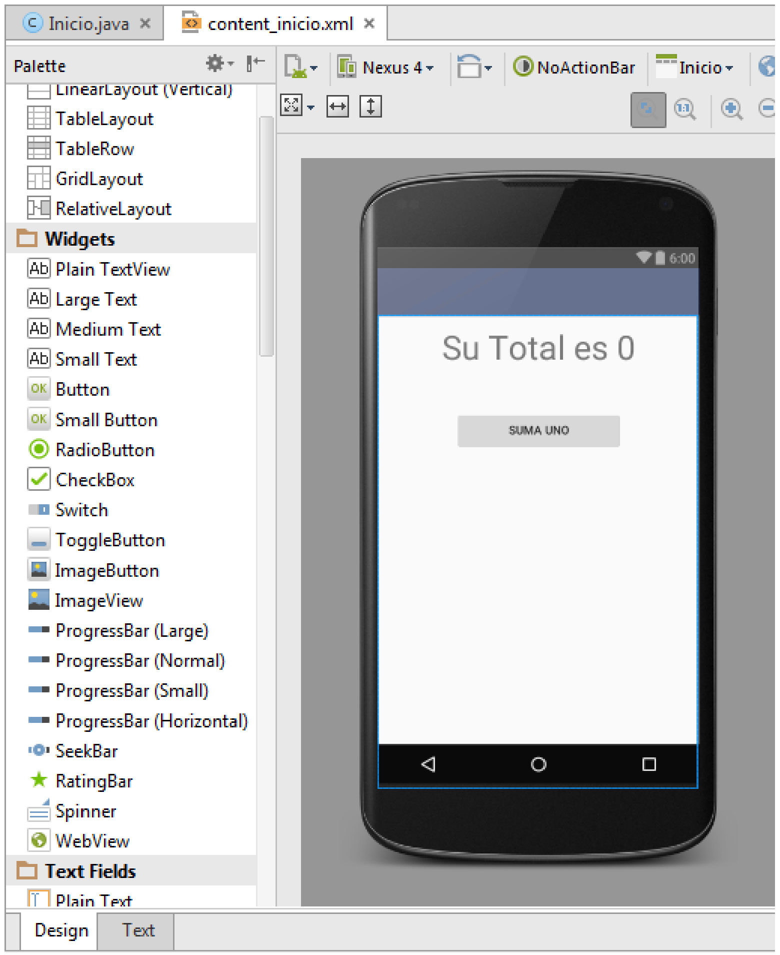
- We add a button:



Inicio.java × content_inicio.xml ×

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.sdm.setiapp.Inicio"
    tools:showIn="@layout/activity_inicio">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Su Total es 0"
        android:textSize="40sp"
        android:layout_centerHorizontal="true"
        android:id="@+id/tvMostrar"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Suma Uno"
        android:id="@+id/Sumar"
        android:layout_below="@+id/tvMostrar"
        android:layout_marginTop="50dp"
        android:width="200dp"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

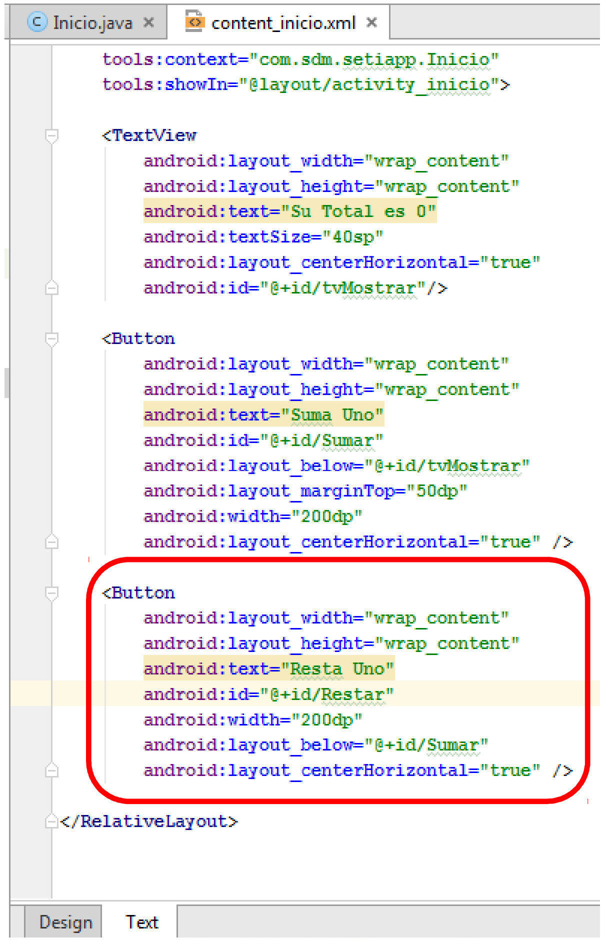
Design Text



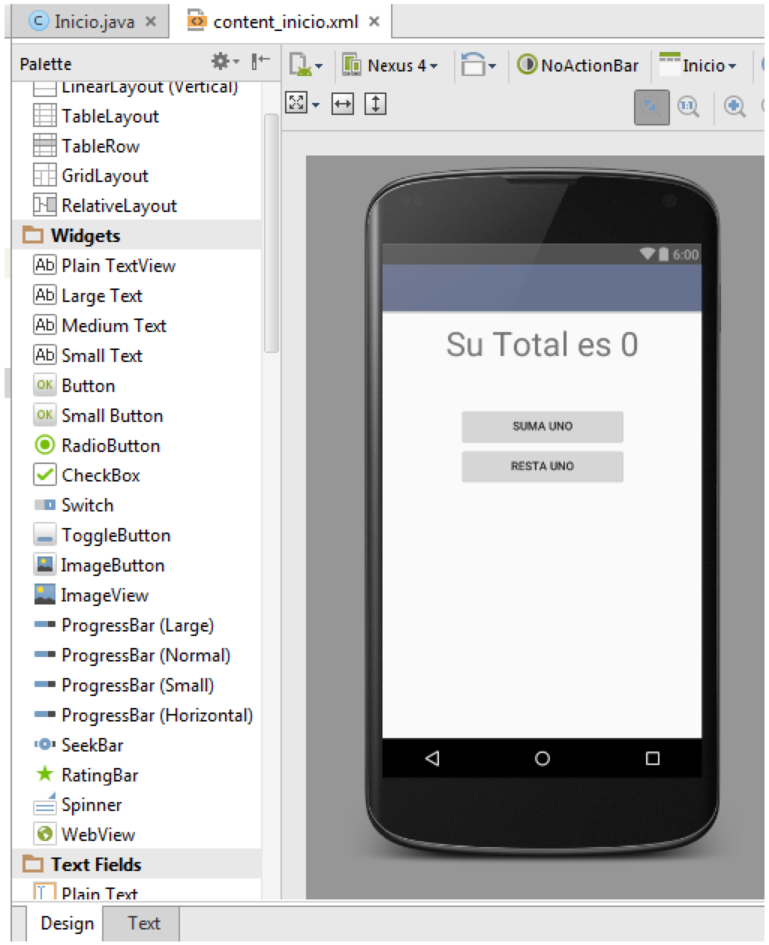
Everything written in the xml They are considered views

Changing the app

- Add another button:

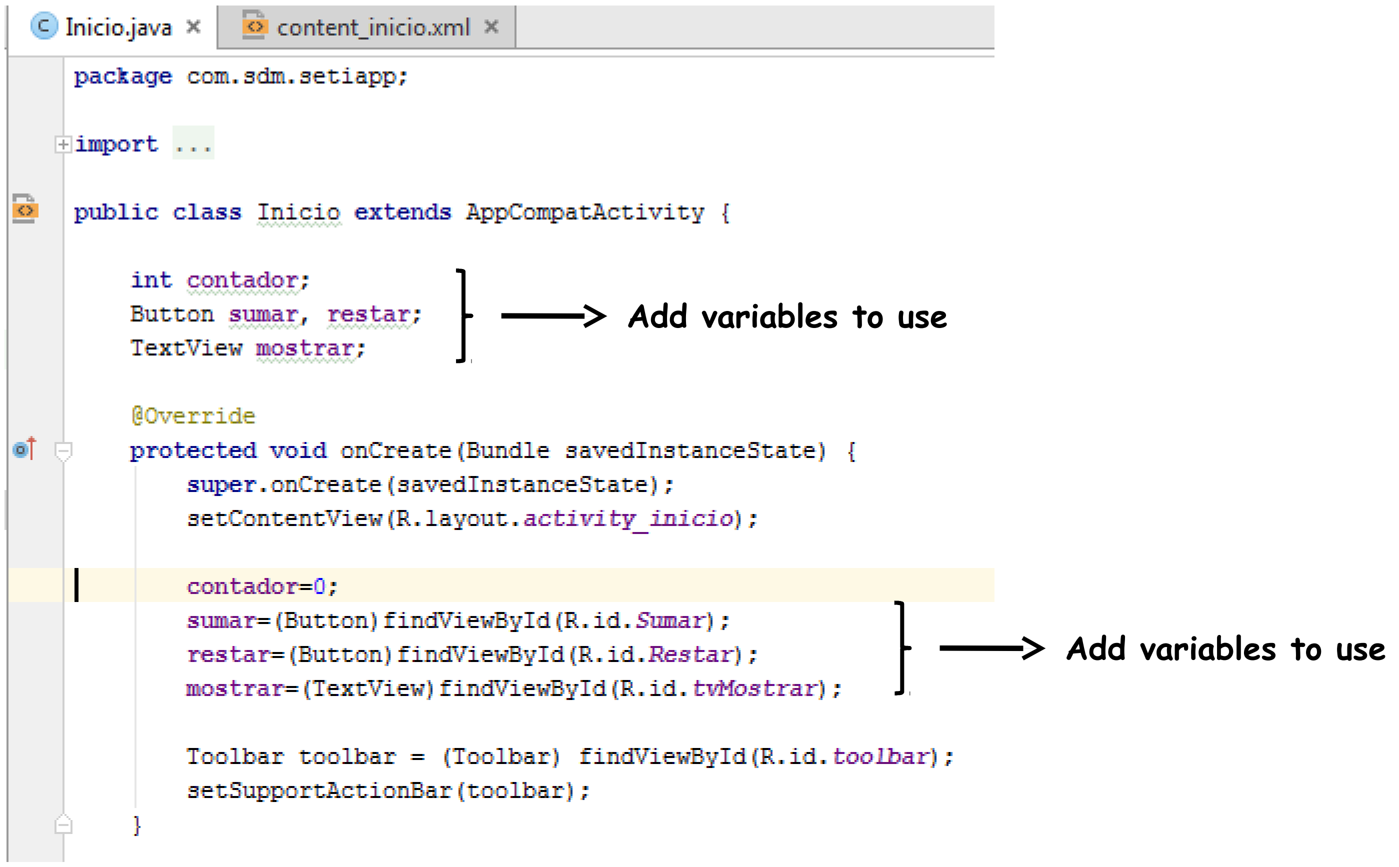


```
    tools:context="com.sdm.setiapp.Inicio"
    tools:showIn="@layout/activity_inicio">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Su Total es 0"
        android:textSize="40sp"
        android:layout_centerHorizontal="true"
        android:id="@+id/tvMostrar"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Suma Uno"
        android:id="@+id/Sumar"
        android:layout_below="@+id/tvMostrar"
        android:layout_marginTop="50dp"
        android:width="200dp"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Resta Uno"
        android:id="@+id/Restar"
        android:width="200dp"
        android:layout_below="@+id/Sumar"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```



Changing the app

- We modify the code in the Startup type:

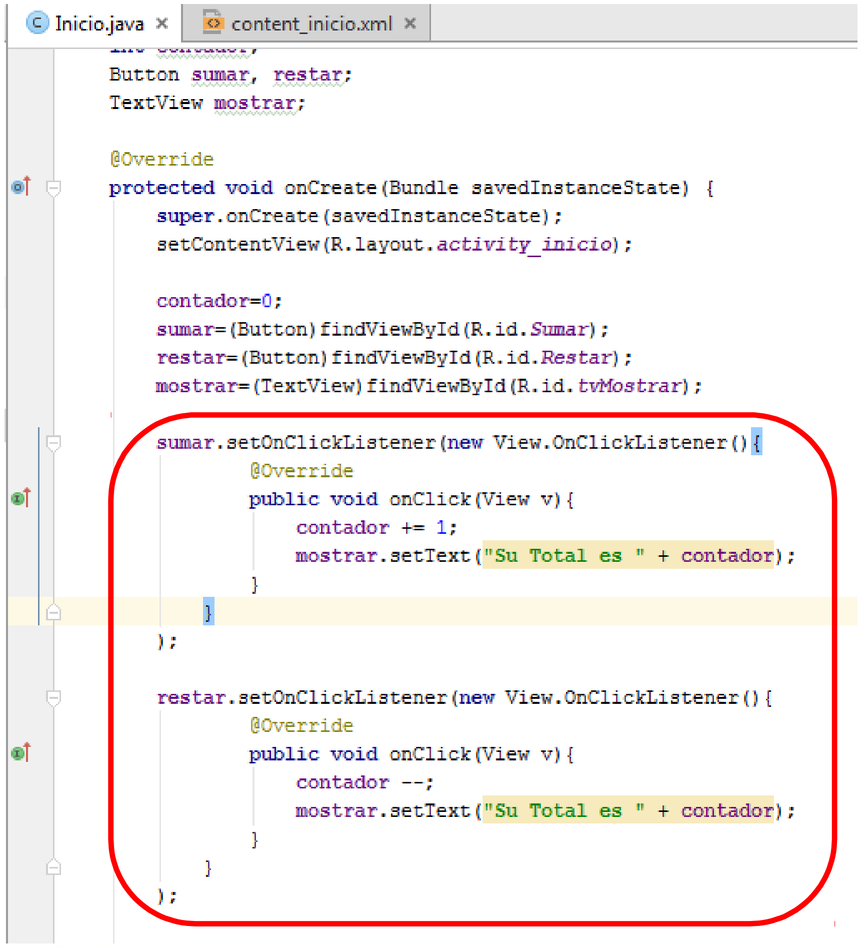


Inicio.java × | content_inicio.xml ×

```
package com.sdm.setiapp;

import ...

public class Inicio extends AppCompatActivity {

    int contador;
    Button sumar, restar;
    TextView mostrar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inicio);

        contador=0;
        sumar=(Button)findViewById(R.id.Sumar);
        restar=(Button)findViewById(R.id.Restar);
        mostrar=(TextView)findViewById(R.id.tvMostrar);

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }
```

Changing the app

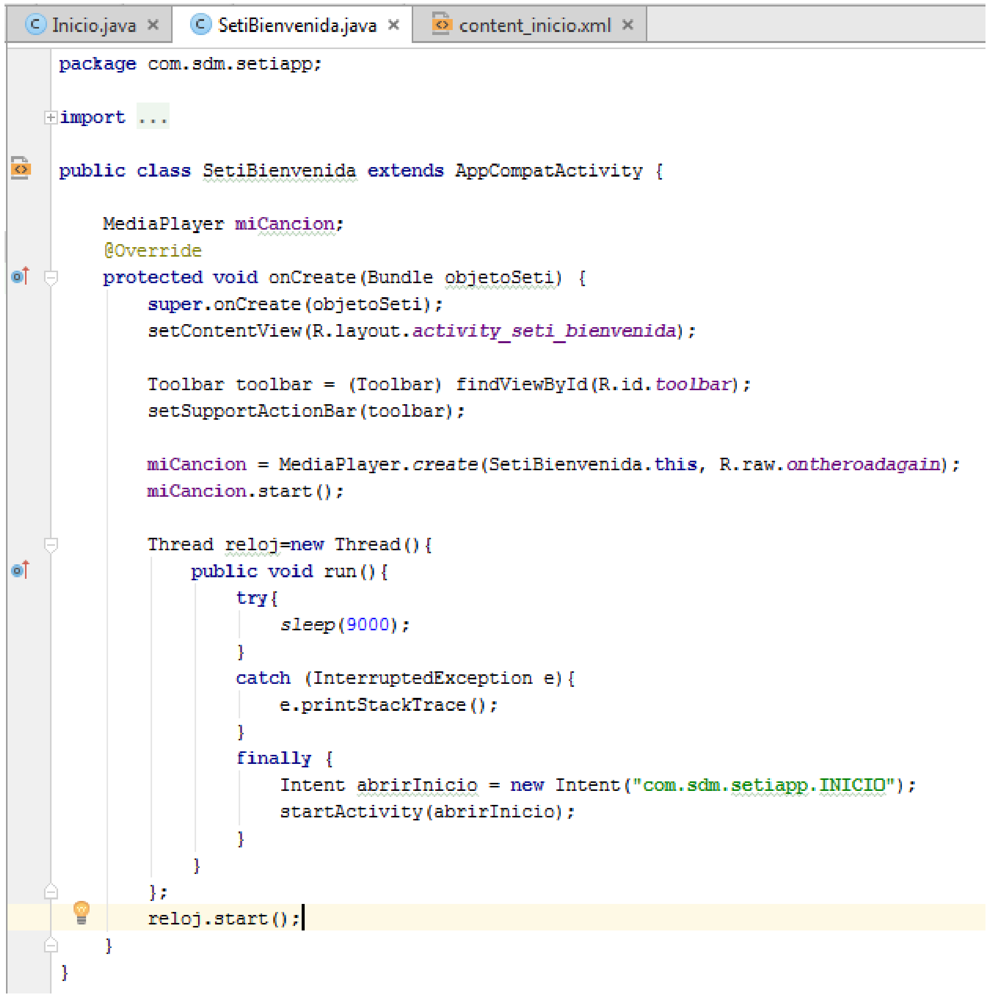
- We use the methods of the class Button:



Inicio.java × | content_inicio.xml ×

```
    Button sumar, restar;
    TextView mostrar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inicio);

        contador=0;
        sumar=(Button)findViewById(R.id.Sumar);
        restar=(Button)findViewById(R.id.Restar);
        mostrar=(TextView)findViewById(R.id.tvMostrar);

        sumar.setOnClickListener(new View.OnClickListener(){
                @Override
                public void onClick(View v){
                    contador += 1;
                    mostrar.setText("Su Total es " + contador);
                }
            }
        );

        restar.setOnClickListener(new View.OnClickListener(){
                @Override
                public void onClick(View v){
                    contador --;
                    mostrar.setText("Su Total es " + contador);
                }
            }
        );
```

Adding 9 seg.



Inicio.java | SetiBienvenida.java | content_inicio.xml

```java
package com.sdm.setiapp;

import ...

public class SetiBienvenida extends AppCompatActivity {

    MediaPlayer miCancion;
    @Override
    protected void onCreate(Bundle objetoSeti) {
        super.onCreate(objetoSeti);
        setContentView(R.layout.activity_seti_bienvenida);

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        miCancion = MediaPlayer.create(SetiBienvenida.this, R.raw.ontheroadagain);
        miCancion.start();

        Thread reloj=new Thread(){
            public void run(){
                try{
                    sleep(9000);
                }
                catch (InterruptedException e){
                    e.printStackTrace();
                }
                finally {
                    Intent abrirInicio = new Intent("com.sdm.setiapp.INICIO");
                    startActivity(abrirInicio);
                }
            }
        };
        reloj.start();
    }
}
```

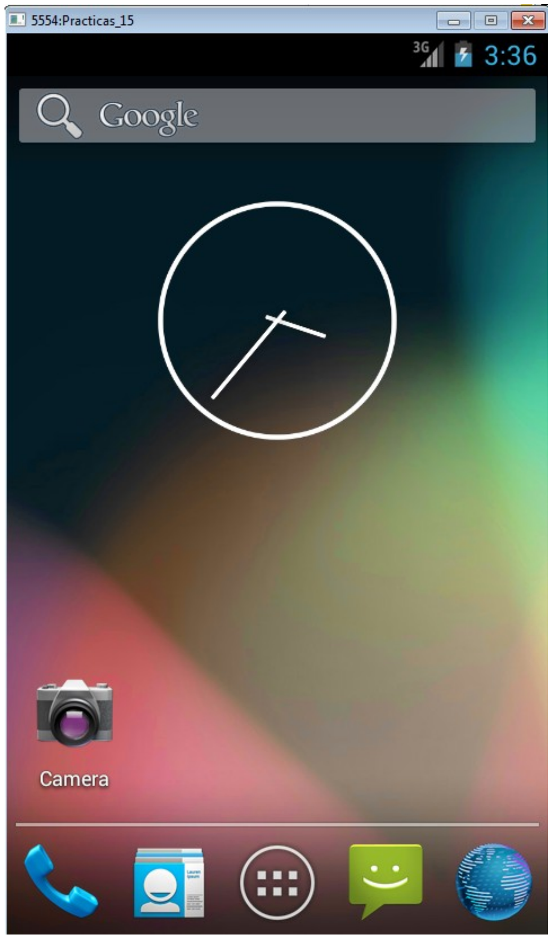
We wanted to add sound, so the application now pause for 9 seconds:

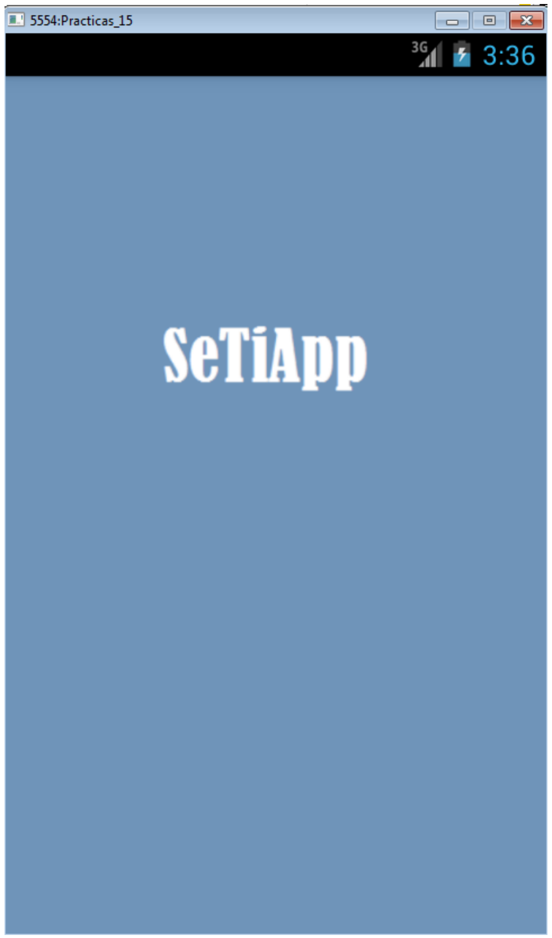
1. **Welcome SeTiApp music and after 9 seconds, the application switches to another Activity**
2. **Song .mp3 file format or .ogg already in the folder beef/raw**
3. **We have an object MediaPlayer which is instantiated with the class and location of the folder that contains the file name and mp3 (lowercase)**
4. **Now we add a thread application sleeping for 9 seg. Then he will start the second activity. Here we add an Intent charging the other activity and initiates the second**

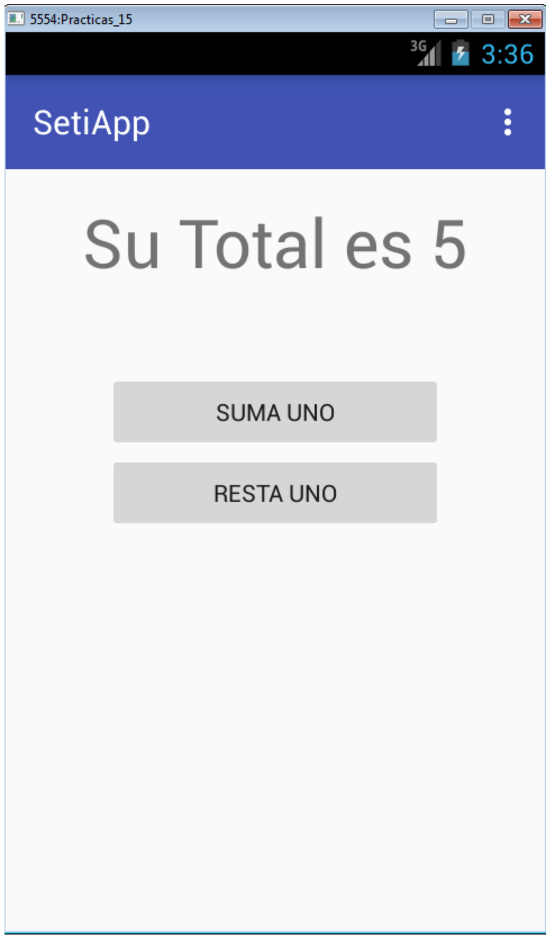
we can test the application

Testing the app

- We start the emulator if we have not done:







Life Cycle

- It happens at the bottom of this activity:
  - The music is still heard, do we want to it be stopped?
  - What about the activity "SeTiBienvenida"?
  - How much memory do we use right now?
  - What if at this time someone calls you?

uc3m | Universidad **Carlos III** de Madrid

Grupo de investigación:
Computer Security Lab

Mobile Devices Security

Degree in Computer Engineering

2019