

uc3m

Universidad **Carlos III** de Madrid

Grupo de investigación:
Computer Security Lab

Mobile Devices Security

Bachelor Degree in Informatics Engineering

2019

Agenda

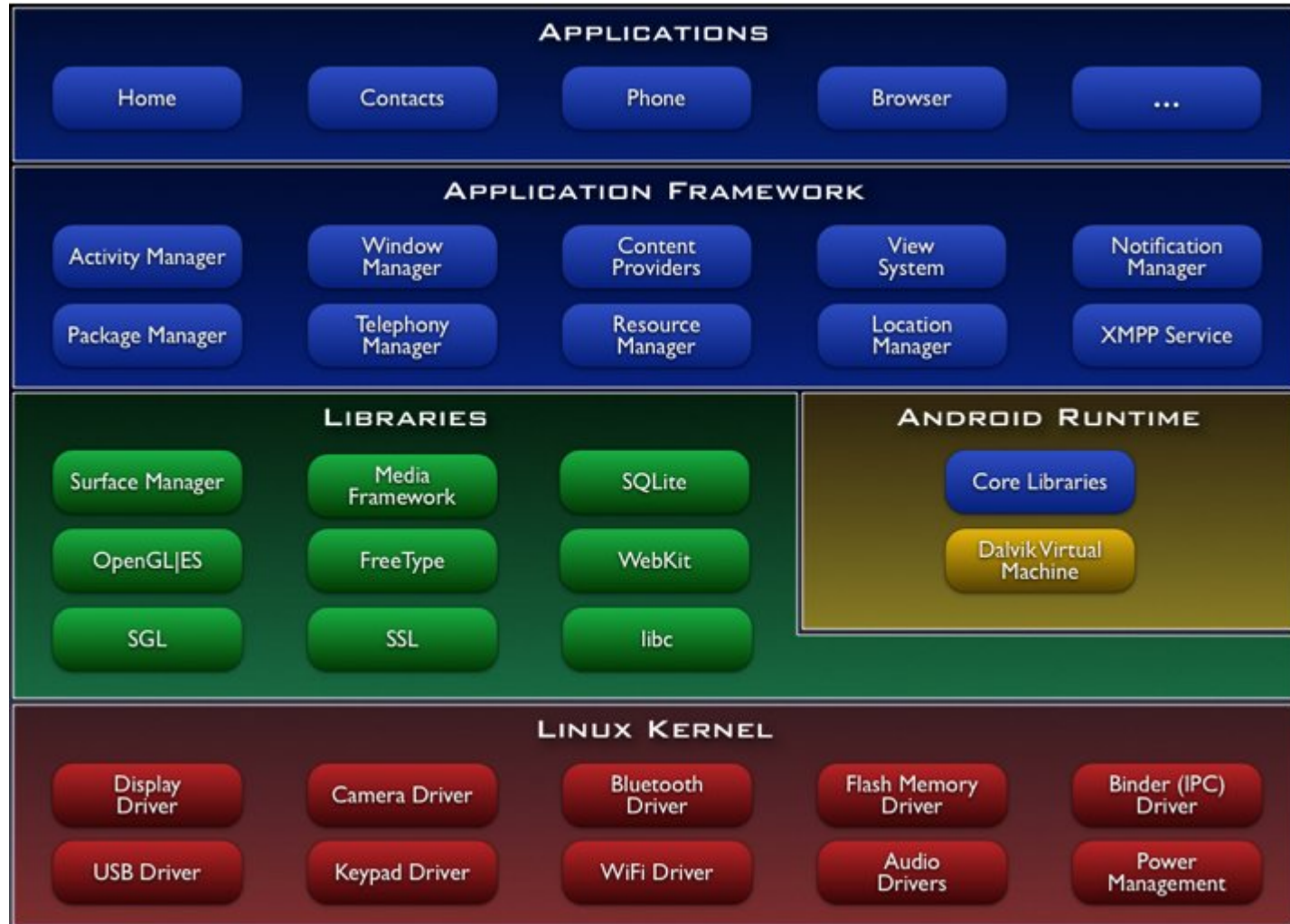
- ▶ Android architecture
- ▶ Components of an Android app
 - ▶ Project structure
 - ▶ Emulator
 - ▶ User interface
 - ▶ Activity
 - ▶ Intent

Android

Android is a software stack for mobile devices that includes an operating system, middleware and all kind of applications

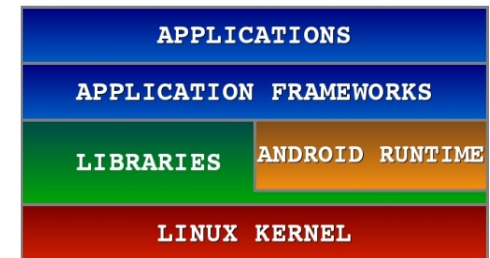
<http://developer.android.com/guide/index.html>

Android architecture



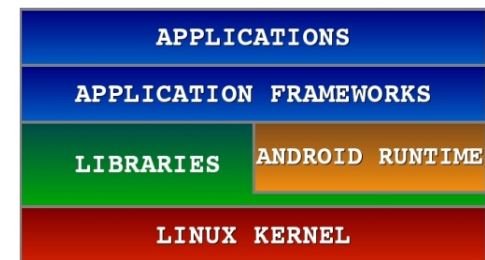
Linux Kernel

- ▶ Based on the Linux Kernel which implements the core system services
 - ▶ Memory and process management
 - ▶ Network Stack
 - ▶ Driver Model
 - ▶ Security
- ▶ Provides an abstraction layer between the HW and the SW stack



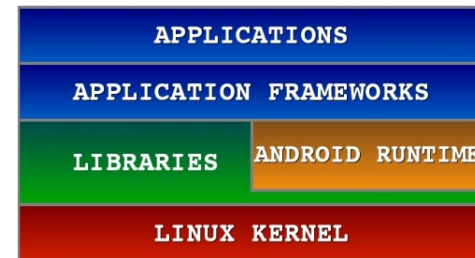
Libraries

- ▶ Includes a set of C / C++ libraries used by Android system *components*
- ▶ Made available to developers through the Android application platform



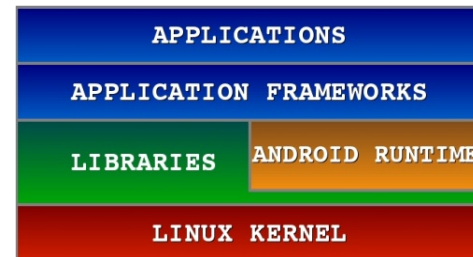
Android Runtime

- ▶ Core Libraries: Provide access to most of the features available in the core libraries in Java:
 - ▶ API (Application Programming Interface)
 - ▶ Data structures
 - ▶ Utilities
 - ▶ File access
 - ▶ Network access
 - ▶ Graphics, etc.



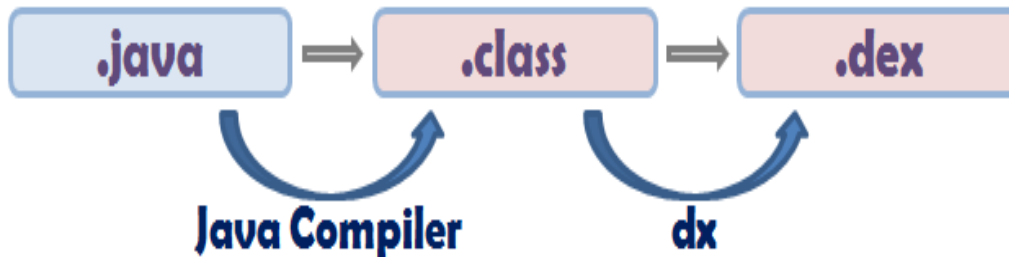
Dalvik Virtual Machine(DVM)

- ▶ Providing the environment in which each application is deployed
 - ▶ An Android application runs in its own process, with its own instance of the Dalvik Virtual Machine (DVM).
 - ▶ Dalvik has been created so that a device can run multiple VMs efficiently
- ▶ Virtual machine based on registries



Dalvik Virtual Machine (DVM)

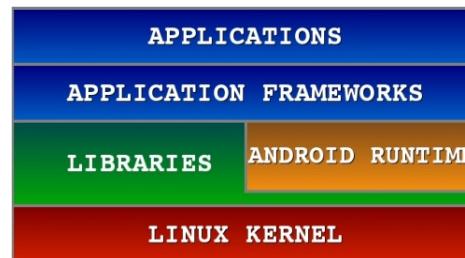
- ▶ DVM executes *.dex* files
 - ▶ Files with *.dex* (**d**alvik **e**xecutable) format are optimized for a minimum memory consumption
 - ▶ For an more efficient compilation



- ▶ DVM is based on the Linux Kernel for:
 - ▶ Multi-thread programming
 - ▶ Low level memory management

Android platform

- ▶ Simplifies and enables the reuse of components
 - ▶ Developers have full access to the same platform APIs used by core applications
 - ▶ Users can replace components



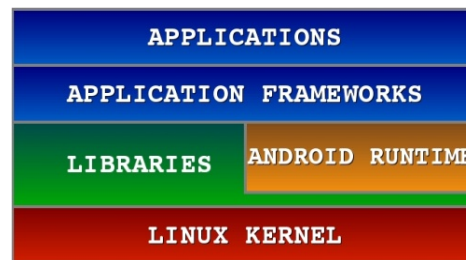
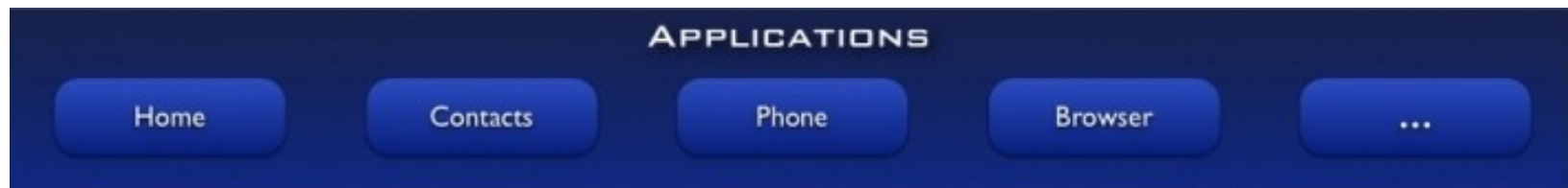
Android platform

► Characteristics

Característica	Rol
View System	Used to build an application, including lists, tables, text fields, buttons, embedded
Content Provider	Allows an application to access data from another application or share you own data
Resource Manager	Provides access to uncoded resources (strings, graphics and payout files)
Notification Manager	It enables all applications to display user alerts in the status bar
Activity Manager	Manages the lifecycle of applications and provides a common navigation order in which activities have been executed

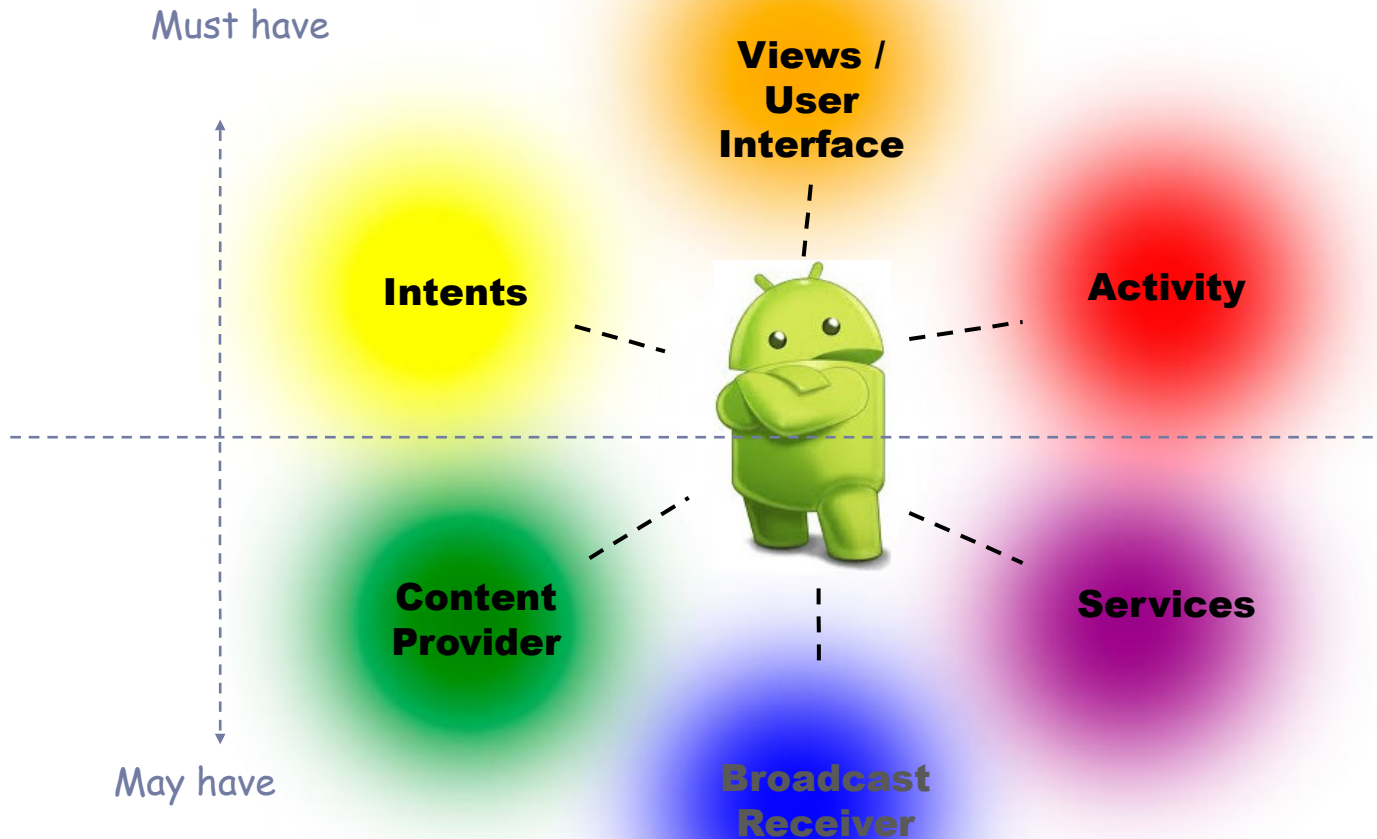
Applications

- ▶ Android provides a set of basic applications:
 - ▶ Email client
 - ▶ SMS
 - ▶ Calendar
 - ▶ Maps, browser, contacts, etc...
- ▶ All applications are written using the Java language



Android App components

Each component type has a different purpose and has a different lifecycle that defines how the component is created and destroyed



Android App components

▶ Activity

- ▶ Single display for an user interface
- ▶ Each activity is completely isolated from one another
- ▶ An application may invoke another application's activity if it allowed to do so

▶ Intents

- ▶ Asynchronous messages
- ▶ Each application runs with separate file permissions to restrict permissions to other application processes, an application can not directly access components from another application

▶ Views

- ▶ Building blocks for the user interface
- ▶ Activity user interface is built with widgets (term used to refer to a discrete object)

▶ Broadcast receivers

- ▶ They are used for alerts
- ▶ May use the status bar for notifications
- ▶ No user interface

▶ Services

- ▶ Run in the background
- ▶ Perform long-running operations
- ▶ Perform remote processes
- ▶ No user interface

▶ Content providers (CP)

- ▶ Manages a shared set of application data
- ▶ May store data in the filesystem, database(SQLite), web, etc... wherever the application has access to
- ▶ Through the CP other applications may request to the above if the CP allows

Project structure

- ▶ Project Explorer

- ▶ java

- ▶ com.sdm.setiapp

- ▶ Res folder: where all our resources are placed

- ▶ layout

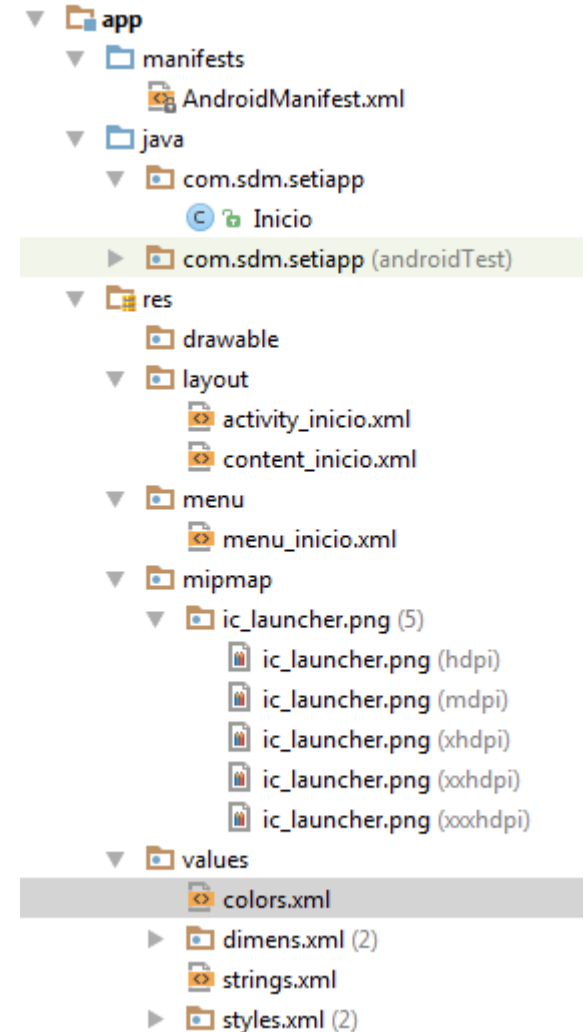
- ▶ activity_inicio.xml

- ▶ mipmap

- ▶ Iconos de la app

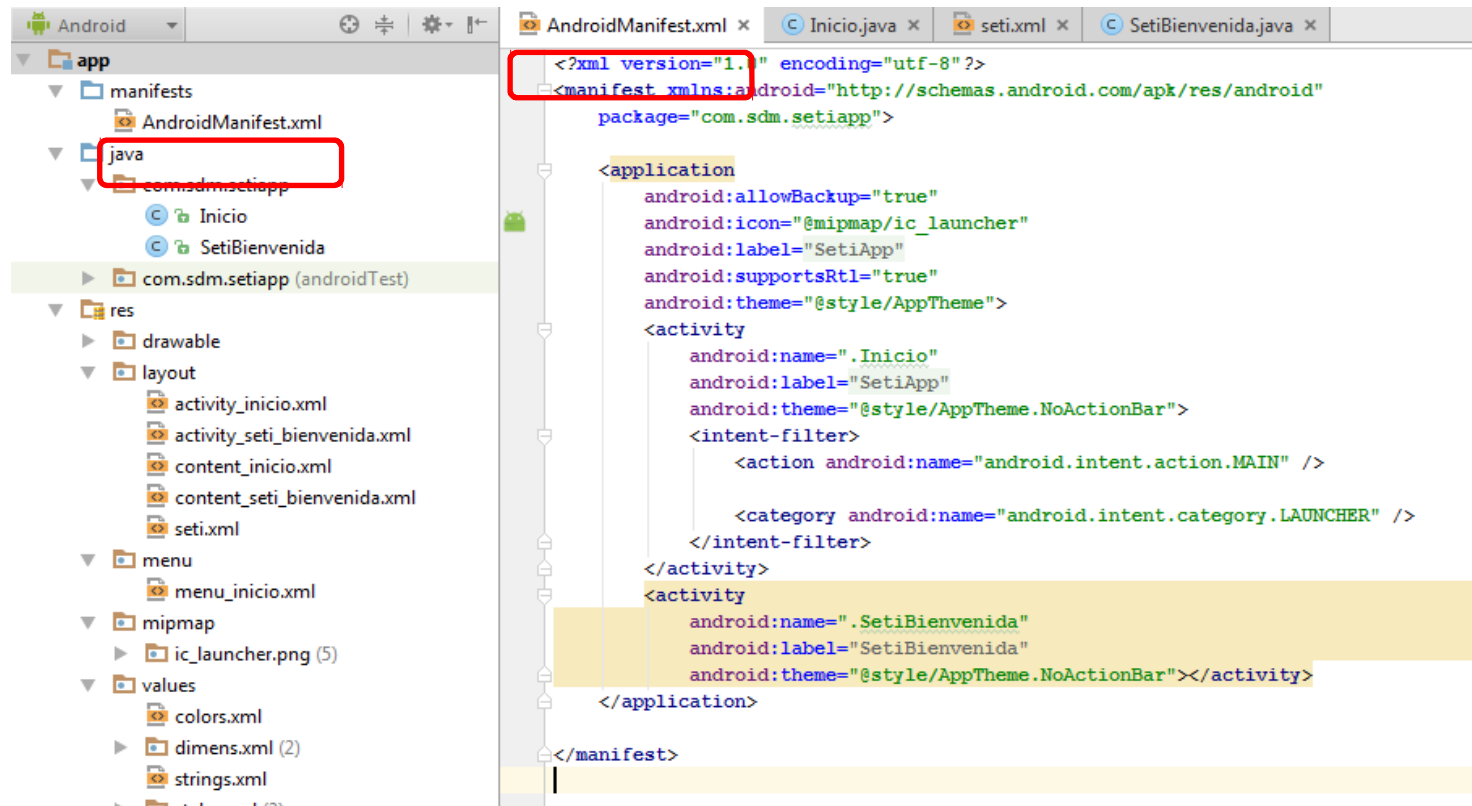
- ▶ values

- ▶ Used to define constant values used by the app



AndroidManifest.xml

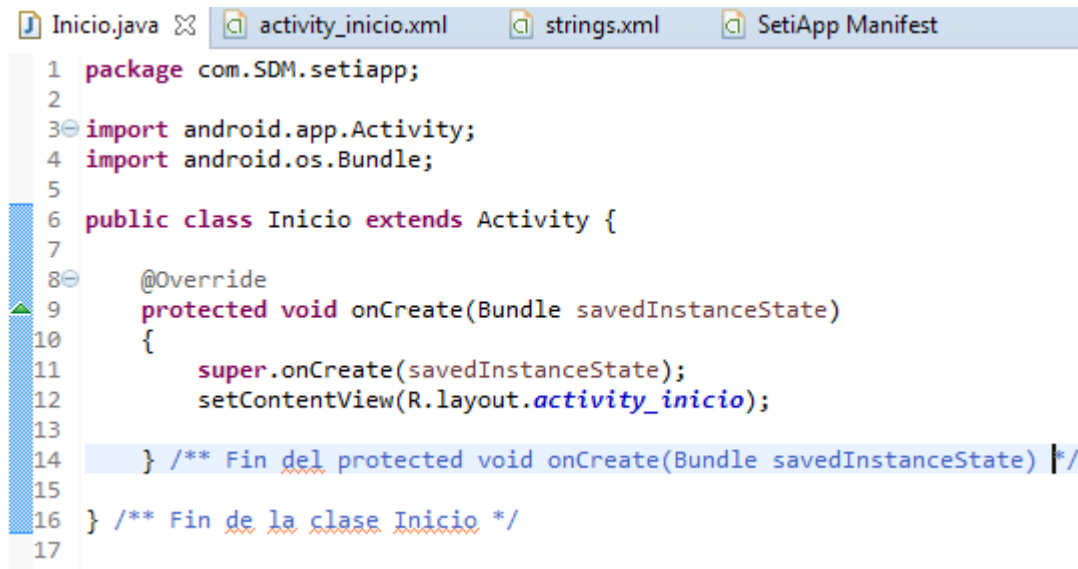
- Configuration file with the basic app configuration



- To view the xml file contents, click on the tab

Inicio.java

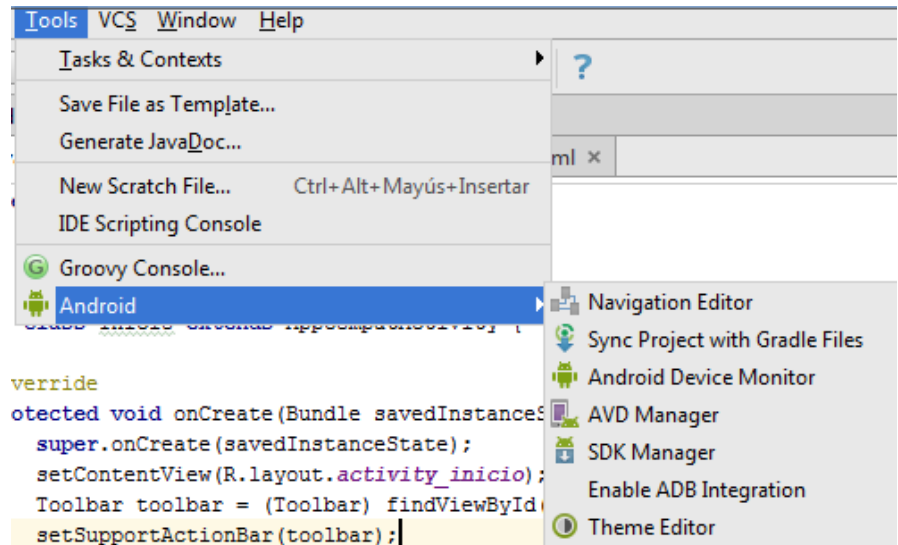
- ▶ In order to show a UI, Android Studio created an xml describing the graphical layout, which is the screen seen in the phone
- ▶ It also creates the logical Java code part



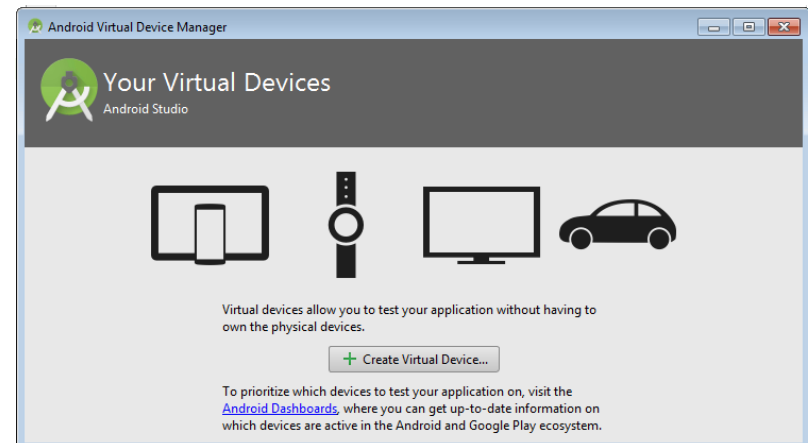
```
1 package com.SDM.setiapp;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5
6 public class Inicio extends Activity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState)
10     {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_inicio);
13
14     } /** Fin del protected void onCreate(Bundle savedInstanceState) */
15
16 } /** Fin de la clase Inicio */
17
```

Emulator

- Tools > Android > AVD Manager



- Create a new AVD

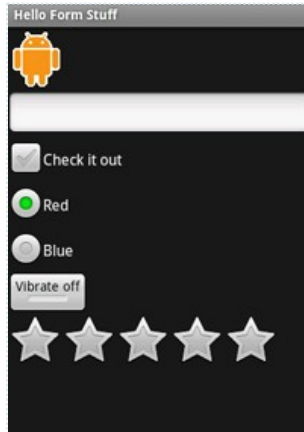


Interface de Usuario IU

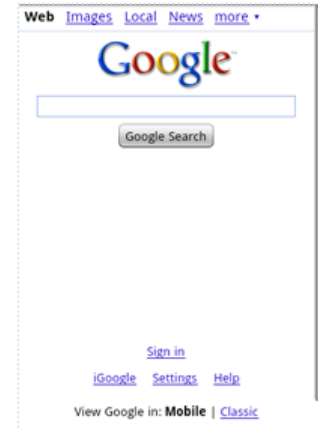
▶ Views

▶ Basic UI components:

- ▶ Button
- ▶ ImageButton
- ▶ EditText
- ▶ CheckBox
- ▶ RadioButton
- ▶ ToggleButton
- ▶ RatingBar



- ▶ DatePicker
- ▶ TimePicker
- ▶ Spinner
- ▶ AutoComplete
- ▶ Gallery
- ▶ MapView
- ▶ WebView



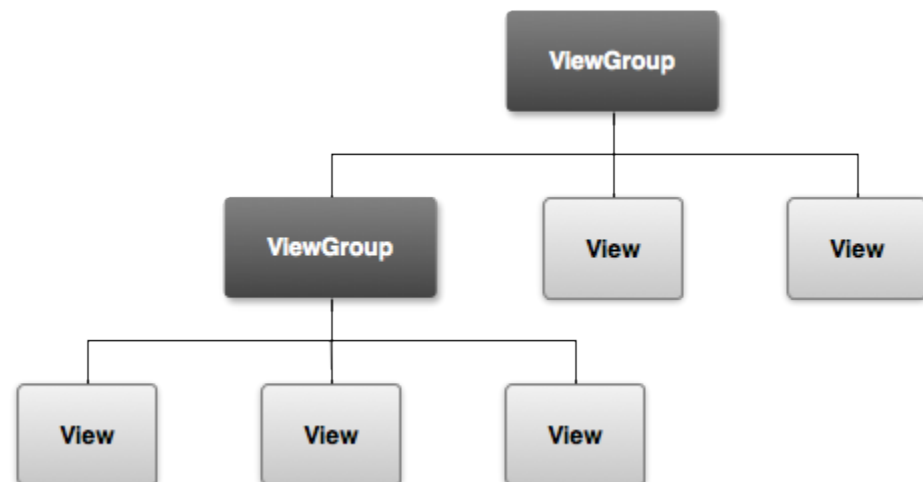
▶ ViewGroup

- ▶ Invisible container containing other views
- ▶ Defines how views are placed in the container
- ▶ LinearLayout, RelativeLayout, GridView, ListView

View, ViewGroup

In an application, all user interfaces (UI) are created using View and ViewGroup objects

- ▶ View – drawable object which a user may interact with
- ▶ ViewGroup – view object containing other views (View or ViewGroups) to define where to place them in the screen
- ▶ ViewGroup is a generic structure for defining Views or other ViewGroups



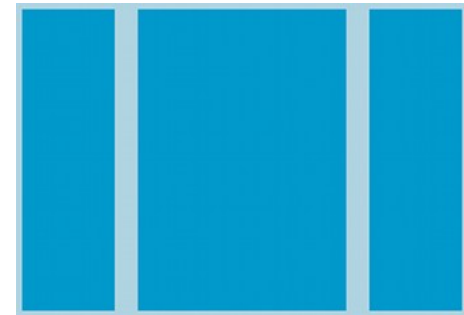
Layout

- Defines a visual structure for a user interface, such as an UI for an activity

Relative Layout



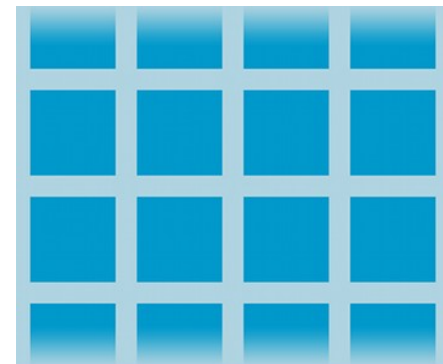
Linear Layout



List View



Grid View



Relative Layout

Layout are defined in an xml file

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```

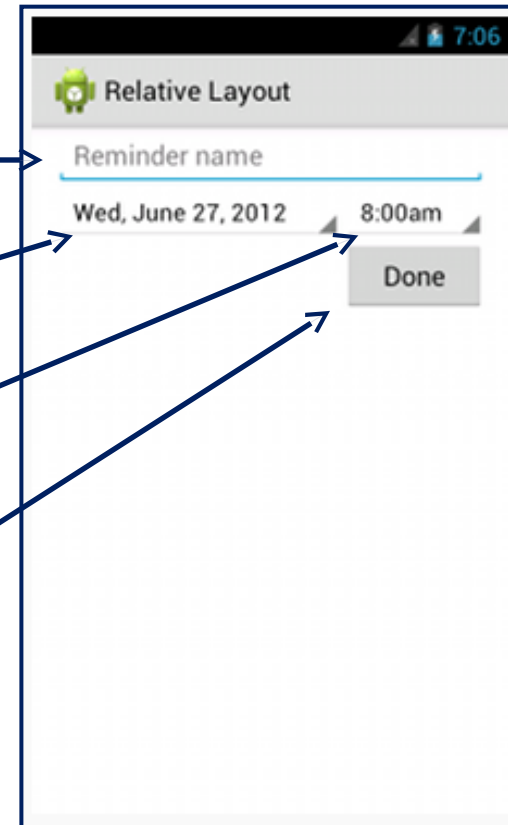


Table Layout

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="1" >

    <TableRow>
        <EditText
            android:layout_column="1"
            android:padding="3dip"
            android:text="Abrir..."
            android:textSize="24sp" />
        <EditText
            android:gravity="right"
            android:padding="3dip"
            android:text="Ctrl -O"
            android:textSize="24sp" />
    </TableRow>
    <TableRow>
        <EditText
            android:layout_column="1"
            android:padding="3dip"
            android:text="Guardar..."
            android:textSize="24sp" />
        <EditText
            android:gravity="right"
            android:padding="3dip"
            android:text="Ctrl -G"
            android:textSize="24sp" />
    </TableRow>
    .....
    .....
</ TableLayout >
```



Grid Layout

```
<?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:orientation="vertical" >
```

```
<GridView
```

```
    android:id="@+id/gridview"  
    android:layout_width="match_parent"  
    android:layout_height="match_content"  
    android:columnWidth="90dp"  
    android:gravity="center"  
    android:horizontalSpacing="10dp"  
    android:numColumns="auto_fit"  
    android:stretchMode="columnWidth"  
    android:verticalSpacing="10dp" />
```

```
</ LinearLayout >
```



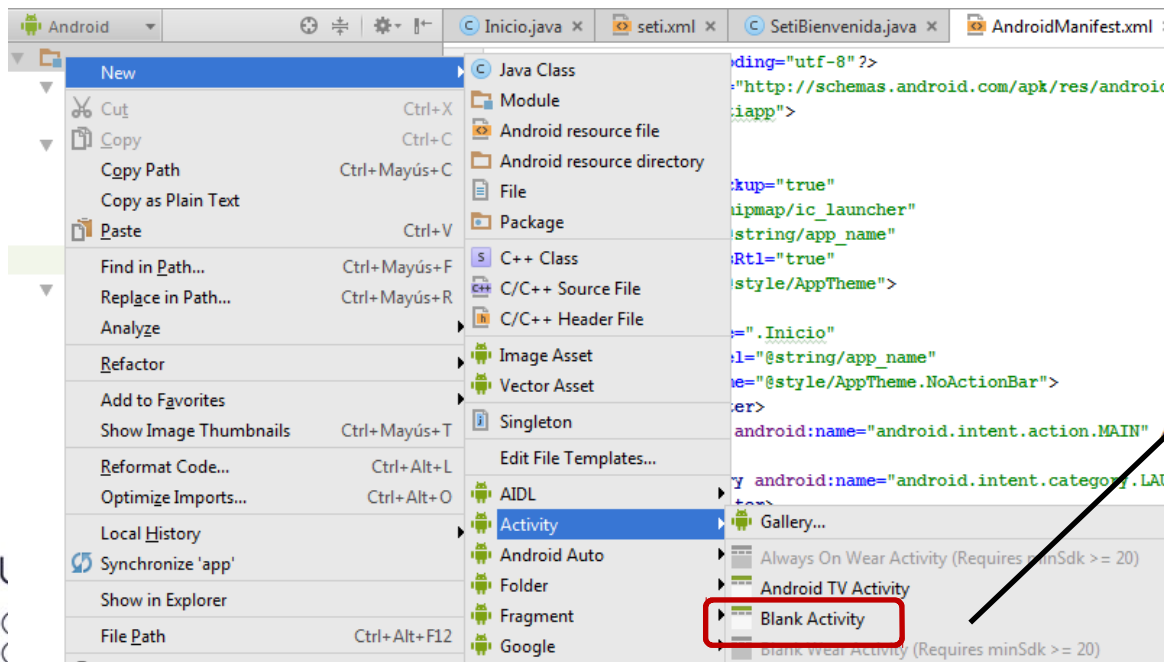
Interactive components

Interactive UI components

Component type	Description	Classes
Button	A button may be clicked to trigger an action	Button
Text field	An editable text field. AutoCompleteTextView allows for suggestions and complete	EditText, AutoCompleteTextView
Checkbox	An on/off button.	CheckBox
Radio button	Only one button from a group may be activated	RadioGroup RadioButton
Toggle button	An on/off button	ToggleButton
Spinner	A drop down list which allows the user to select a value	Spinner
Pickers	A dialog which allows users to select a set of values using buttons. Use an DatePicker for selecting dates (year, month, day). TimePicker for selecting time values (hour, minute, AM/PM), which will be formatted according to the regional user configuration	DatePicker, TimePicker

Activity

- ▶ Similar to a window – UI component, a screen shown to the user
- ▶ It is possible to have multiple activities, but only one at a time gets displayed
- ▶ Activities are made from two parts:
 - ▶ Logical – java file which controls the UI components
 - ▶ Graphical – XML declaring the elements are defining how they will be displayed
- ▶ Create a new class: app > New > Activity



**Choose the class
name:
SetiBienvenida**

Activity

► An activity may be triggered in two ways:

1. **Explicit** – if the Activity to launch is known Ex.: the default activity declared in the manifest.xml
2. **Implicit** – only the desired action is known. Ex.: open a browser, send an SMS, take a photo, etc...



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.SDM.setichat"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk
8         android:minSdkVersion="18"
9         android:targetSdkVersion="21" />
10
11     <application
12         android:allowBackup="true"
13         android:icon="@drawable/ic_launcher"
14         android:label="@string/app_name"
15         android:theme="@style/AppTheme" >
16         <activity
17             android:name=".Inicio"
18             android:label="@string/app_name" >
19             <intent-filter>
20                 <action android:name="android.intent.action.MAIN" />
21
22                 <category android:name="android.intent.category.LAUNCHER" />
23             </intent-filter>
24         </activity>
25     </application>
26
27 </manifest>
28
```

Intents

- ▶ Messages structure used to activate components:
 - ▶ Exists specifically for the component activation
 - ▶ Describes an operation/event of interest
 - ▶ It is not related with the Android's generic class 'Message'
- ▶ Two main purposes
 - ▶ Describe an operation to perform (the most common)
 - ▶ Describe something that has occurred
- ▶ Intents have three main uses:
 - ▶ Starting an activity
 - ▶ Starting a service
 - ▶ Delivering a broadcast

Intents

- ▶ Is an abstract representation of:
 - ▶ An operation that will be triggered
 - ▶ An event that has occur that other components need to be aware of
- ▶ Mechanism to invoke components
- ▶ Allow to call other external applications, trigger events or another apps that are listening and could respond, execute alarms, etc...
- ▶ Messages from the OS
- ▶ The way different components communicate among them
- ▶ OS is in charge to control if an intent communicates with another activity
- ▶ Intents may be sent from one activity to another
- ▶ Intents may carry data

uc3m

Universidad **Carlos III** de Madrid

Grupo de investigación:
Computer Security Lab

Mobile Devices Security

Bachelor Degree in Informatics Engineering

2019