

UNIDAD 0

INTRODUCCIÓN A ANDROID



Diseño de Interfaces Web



UD0. Introducción a Android

1. Introducción
2. La plataforma Android
3. Arquitectura de Android
4. Las versiones de Android y niveles de API
5. Creación de un primer programa
6. Ejecución del programa
7. Elementos de un proyecto de Android
8. Componentes de una aplicación



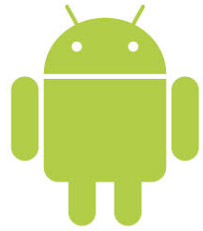
UD0. Introducción a Android

- 1. Introducción**
2. La plataforma Android
3. Arquitectura de Android
4. Las versiones de Android y niveles de API
5. Creación de un primer programa
6. Ejecución del programa
7. Elementos de un proyecto de Android
8. Componentes de una aplicación

1. Introducción

- Instalar y trabajar con el entorno de desarrollo (Android Studio+ Android SDK)
- Describir la estructura de un proyecto Android





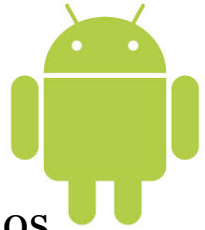
UD0. Introducción a Android

1. Introducción
2. **La plataforma Android**
3. Arquitectura de Android
4. Las versiones de Android y niveles de API
5. Creación de un primer programa
6. Ejecución del programa
7. Elementos de un proyecto de Android
8. Componentes de una aplicación



2.1. ¿Qué hace a Android tan especial?

- **Plataforma realmente abierta.** Basada en Linux y de código abierto. Se puede usar y customizar el sistema sin pagar royalties.
- **Adaptable a cualquier tipo de hardware.** No sólo teléfonos y tabletas: relojes, cámaras, electrodomésticos y sistemas empotrados. Suponer un esfuerzo adicional al programador. La aplicación ha de funcionar correctamente en dispositivos con gran variedad de tipos de entrada, pantalla, memoria, etc.
- **Portabilidad asegurada.** Java nos asegura que podrán ser ejecutadas en cualquier tipo de CPU, tanto presente como futuro.
- **Arquitectura basada en componentes inspirados en Internet.** Diseño de la interfaz de usuario se hace en xml (móvil de pantalla reducida o en un TV).
- **Filosofía de dispositivo siempre conectado a Internet.**
- **Gran cantidad de servicios incorporados.** Localización basada tanto en GPS como en redes, bases de datos con SQL, reconocimiento y síntesis de voz, navegador, multimedia.



2.1. ¿Qué hace a Android tan especial?

- **Aceptable nivel de seguridad.** Los programas se encuentran aislados unos de y dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a Internet, etc.)
- **Optimizado para baja potencia y poca memoria.** Uso máquina virtual de Java optimizada para dispositivos móviles.
- **Alta calidad de gráficos y sonido.** Gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en 3 dimensiones basados en OpenGL. Incorpora codecs estándar más comunes de audio y vídeo, incluyendo H.264 (AVC), MP3, AAC, etc.



2.2. Los orígenes

- Google adquiere Android Inc. en el año **2005**. Se crea **máquina virtual Java** optimizada para móviles (Dalvik VM).
- En el año **2007** se crea el consorcio **Handset Alliance** con el objetivo de **desarrollar estándares abiertos para móviles**. Está formado por Google, Intel, Texas Instruments, Motorola, T-Mobile, Samsung, Ericson, Toshiba, Vodafone, NTT DoCoMo, Sprint Nextel y otros
- En noviembre del **2007** se lanza una **primera versión del Android SDK**.
- **2008** primer móvil con Android (T-Mobile G1). En octubre Google libera el código fuente de Android y se abre **Android Market**.
- 2009 Google lanza la versión 1.5 del SDK (teclado en pantalla) y la versión 2.0.
- **2010** versiones 2.1, 2.2 y 2.3. **Android se consolida** como uno de los sistemas operativos para móviles más utilizados.



2.2. Los orígenes



- **2011** versión 3.0, 3.1 y 3.2 (**tabletas**) y la 4.0 (móviles/tabletas). Se consolida como la plataforma para móviles más importante alcanzando una **cuota de mercado superior al 50%**.
- En **2012** Google cambia su estrategia en su tienda de descargas online, reemplazando Android Market por **Google Play Store**. Donde en un solo portal **unifica** tanto la descarga de **aplicaciones** como de **contenidos**. En este año aparecen las versiones 4.1 y 4.2 del SDK. Android mantiene su espectacular crecimiento, alcanzando a finales de año una **cuota de mercado del 75%**.
- En abril de **2013** se hizo público que Android alcanzó el **92%** en ventas de nuevos smartphones para el trimestre comprendido entre diciembre 2012 y febrero 2013 en España.



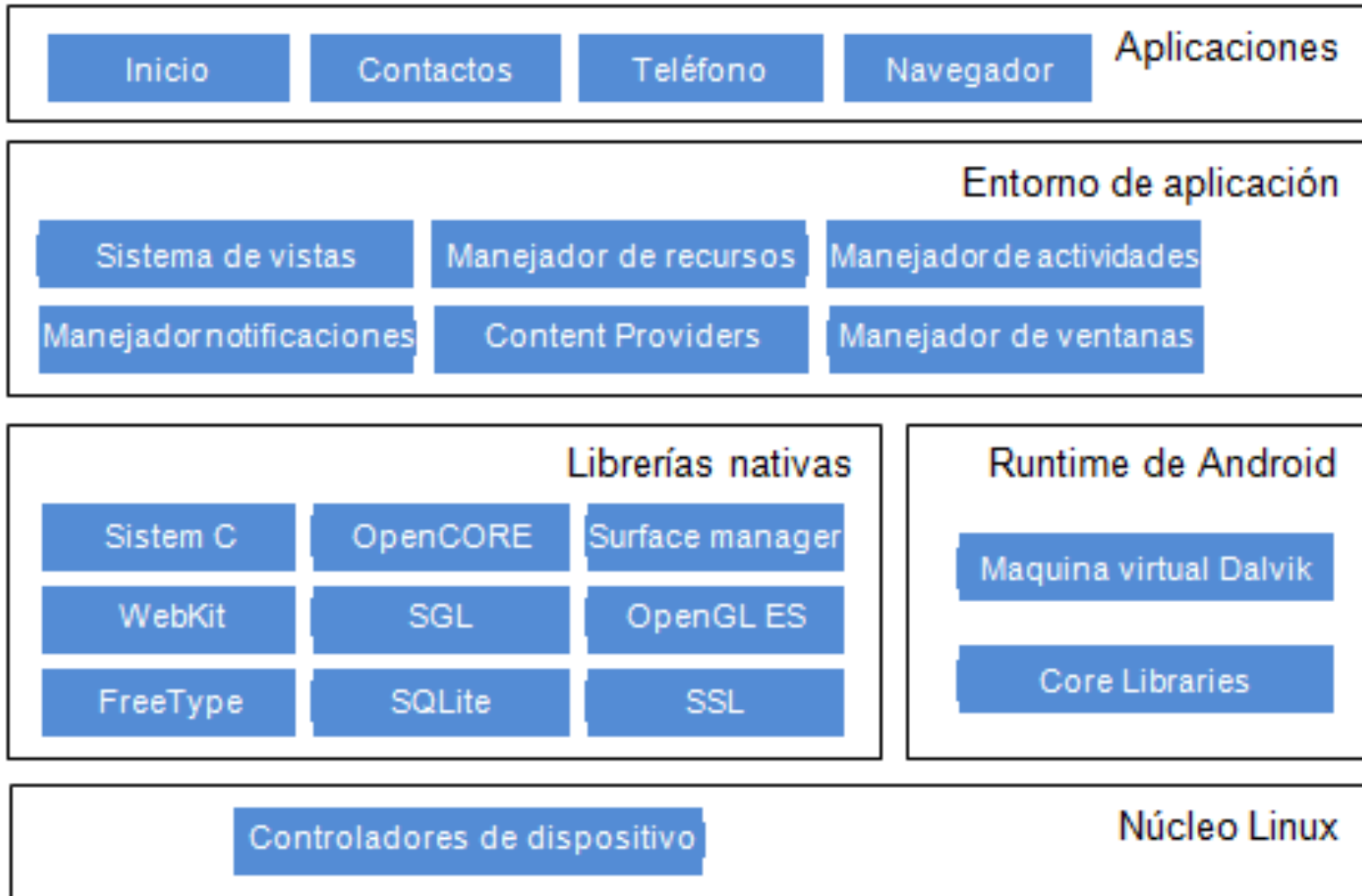


UD0. Introducción a Android

1. Introducción
2. La plataforma Android
- 3. Arquitectura de Android**
4. Las versiones de Android y niveles de API
5. Creación de un primer programa
6. Ejecución del programa
7. Elementos de un proyecto de Android
8. Componentes de una aplicación



3. Arquitectura de Android





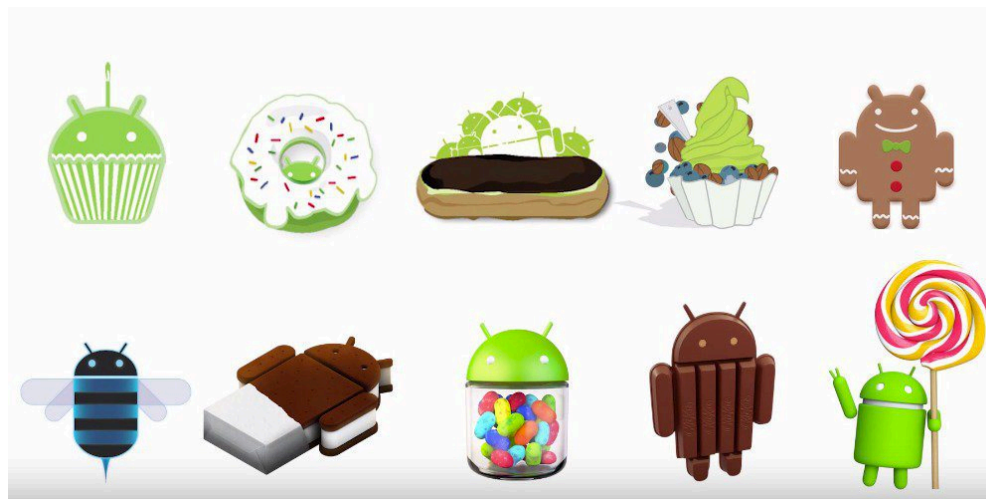
UD0. Introducción a Android

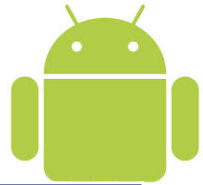
1. Introducción
2. La plataforma Android
3. Arquitectura de Android
4. **Las versiones de Android y niveles de API**
5. Creación de un primer programa
6. Ejecución del programa
7. Elementos de un proyecto de Android
8. Componentes de una aplicación

4. Las versiones de Android



- Antes de empezar a proyecto en Android hay que elegir la versión del sistema para la que deseamos realizar la aplicación.
- Es muy importante observar que hay clases y métodos que están disponibles a partir de una versión, si las vamos a usar hemos de conocer la versión mínima necesaria.
- Cuando se ha lanzado una nueva plataforma siempre ha sido compatible con las versiones anteriores. Es decir, solo se añaden nuevas funcionalidades y en el caso de modificar alguna funcionalidad no se elimina, se etiquetan como obsoletas pero se pueden continuar utilizando.

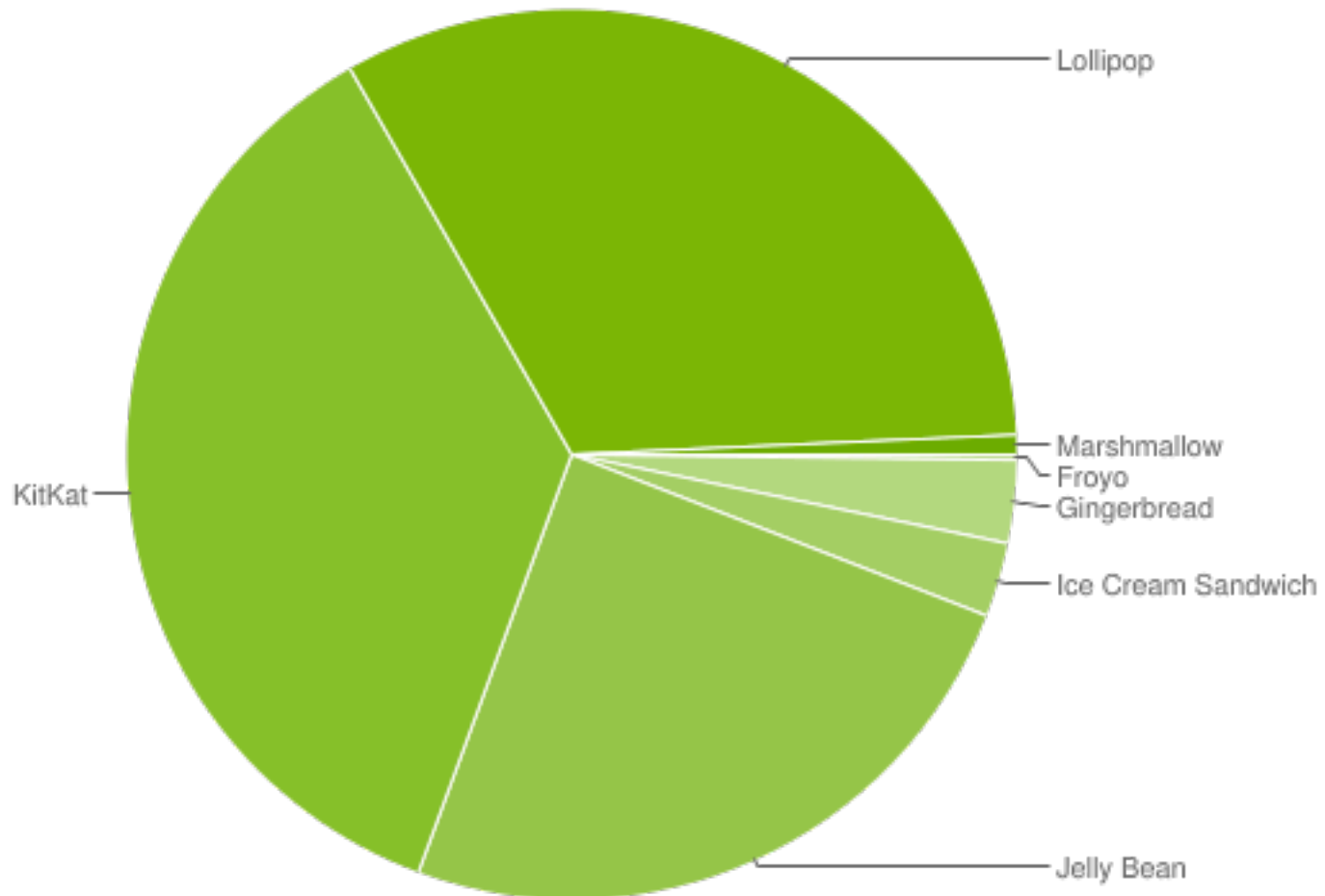




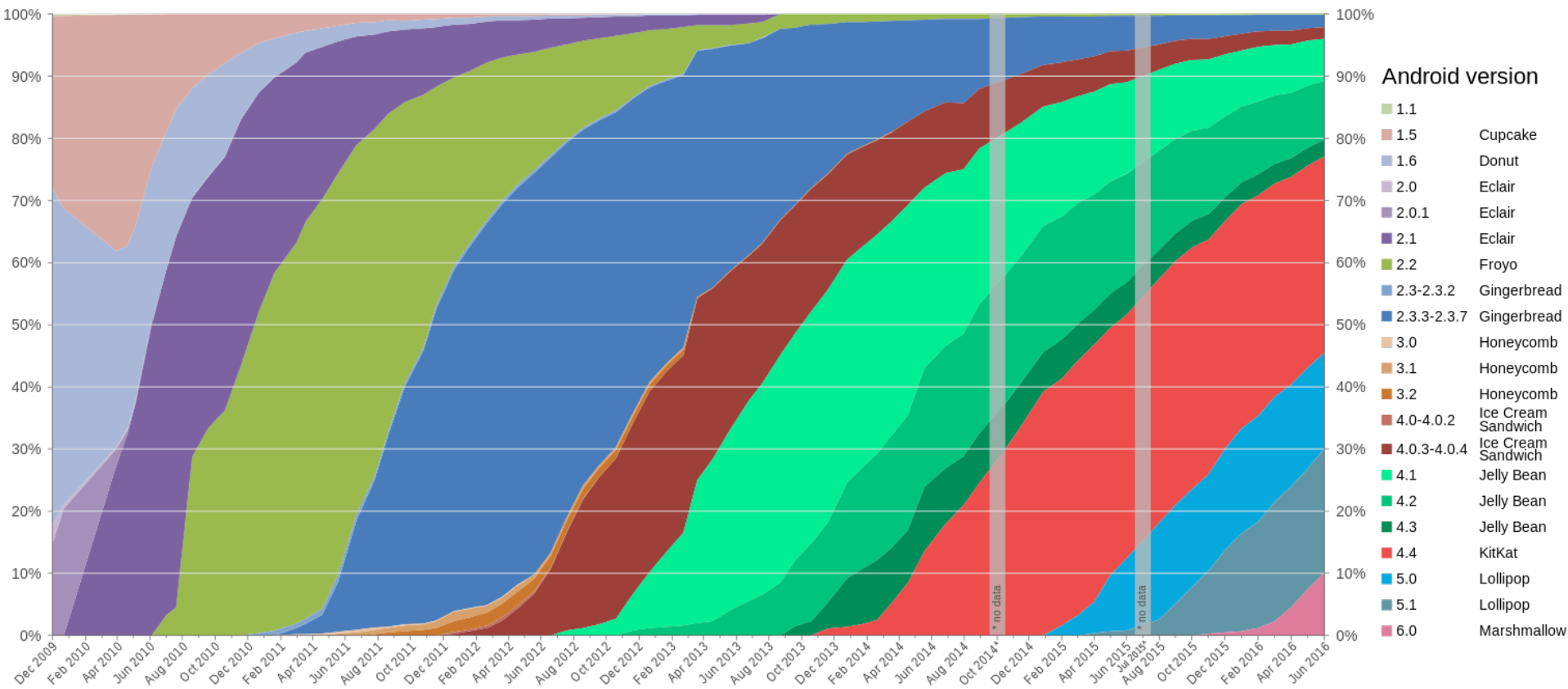
4. Las versiones de Android

Nombre código	Número de versión	Fecha de lanzamiento	Nivel de API
Apple Pie	1.0	23 de septiembre de 2008	1
Banana Bread	1.1	9 de febrero de 2009	2
Cupcake	1.5	27 de abril de 2009	3
Donut	1.6	15 de septiembre de 2009	4
Eclair	2.0–2.1	26 de octubre de 2009	5–7
Froyo	2.2–2.2.3	<u>20 de mayo</u> de 2010	8
Gingerbread	2.3–2.3.7	6 de diciembre de 2010	9–10
Honeycomb ¹	3.0–3.2.6	22 de febrero de 2011	11–13
Ice Cream Sandwich	4.0–4.0.4	18 de octubre de 2011	14–15
Jelly Bean	4.1–4.3.1	9 de julio de 2012	16–18
KitKat	4.4–4.4.4, 4.4W–4.4W.2	31 de octubre de 2013	19–20
Lollipop	5.0–5.1.1	12 de noviembre de 2014	21–22
Marshmallow	6.0–6.0.1	5 de octubre de 2015	23
Nougat	7.0	15 de junio de 2016	24

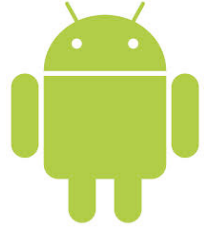
4. Las versiones de Android



4. Las versiones de Android



Junio 2016



UD0. Introducción a Android

1. Introducción
2. La plataforma Android
3. Arquitectura de Android
4. Las versiones de Android y niveles de API
5. Creación de un primer programa
6. Ejecución del programa
7. **Elementos de un proyecto de Android**
8. Componentes de una aplicación



7. Elementos de un proyecto Android

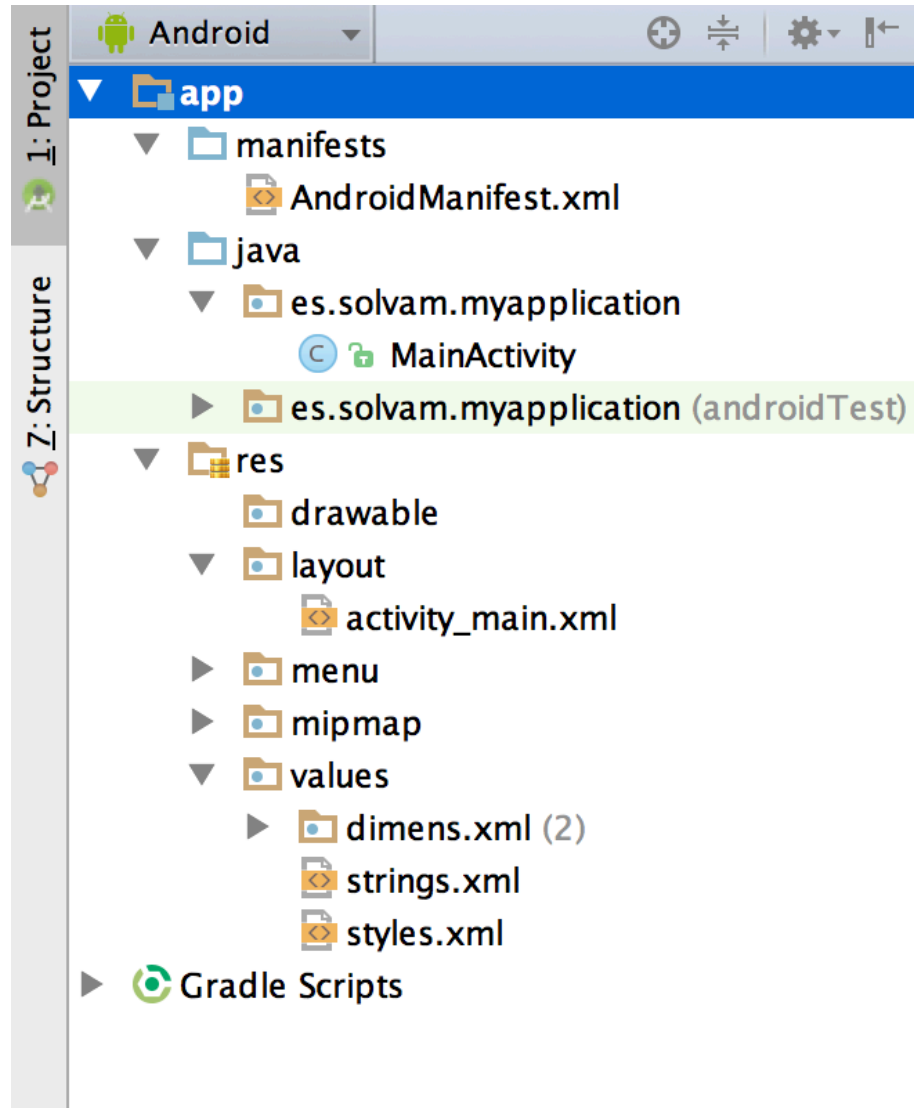
Partes del proyecto

Propiedad	Nombre
Nombre de proyecto	Nombre del proyecto
Nombre de la aplicación	Nombre de la aplicación.
Nombre del paquete	Nombre del paquete. Debemos utilizar un nombre DNS
Primera actividad (pantalla)	Nombre de la pantalla principal del entrada a la aplicación
Versión mínima de SDK	Versión mínima del SDK Android

- "Activity": ventana que contiene la interfaz de usuario de nuestra aplicación. Una aplicación puede tener varias pantallas o puede no tener ninguna (un servicio que esté ejecutando tareas de fondo)

7. Elementos de un proyecto Android

Carpetas de un proyecto





7. Elementos de un proyecto Android

Partes del proyecto

The screenshot shows the Android Studio interface. On the left, the 'Project' and 'Structure' toolbars are visible. The 'Project' view shows the hierarchy: 'app' > 'java' > 'es.solvam.myapplication' > 'MainActivity'. The 'Structure' view shows the layout: 'res' > 'drawable' > 'layout' > 'activity_main.xml'. The main editor displays the 'MainActivity.java' file. The code is as follows:

```
package es.solvam.myapplication;

import ...

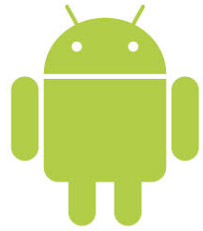
public class MainActivity extends Activity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
}
```

A red box highlights 'MainActivity' in the Project view, and another red box highlights 'MainActivity' in the code. A red arrow points from the first box to the second.

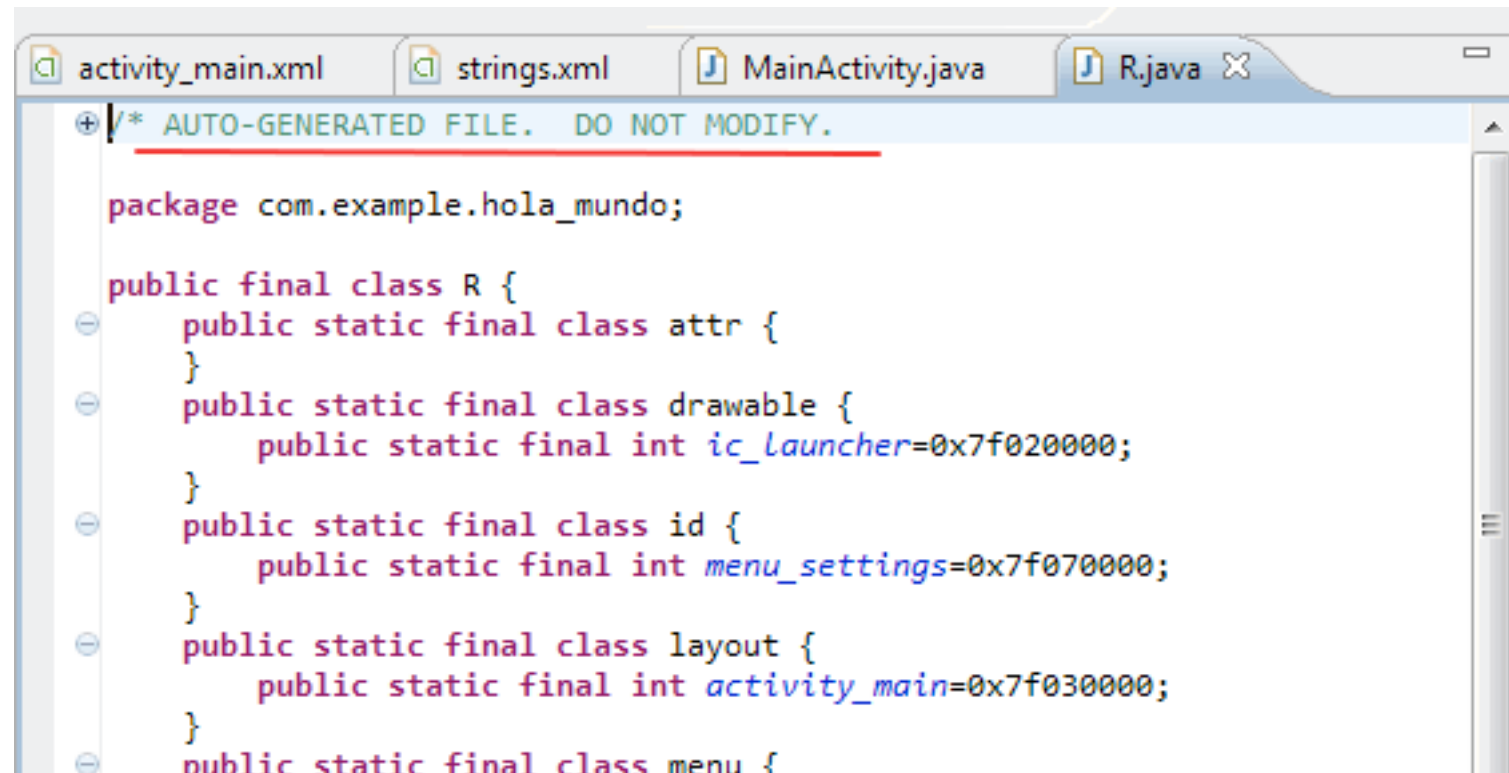
- "java" es la carpeta donde escribiremos nuestro código Java para la aplicación Android.



7. Elementos de un proyecto Android

Partes del proyecto

- El fichero que genera, llamado "R" es el que contiene los recursos del programa: pantallas, gráficos, etc.



```
activity_main.xml  strings.xml  MainActivity.java  R.java X
+ /* AUTO-GENERATED FILE. DO NOT MODIFY.
package com.example.hola_mundo;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class id {
        public static final int menu_settings=0x7f070000;
    }
    public static final class layout {
        public static final int activity_main=0x7f030000;
    }
    public static final class menu {
```

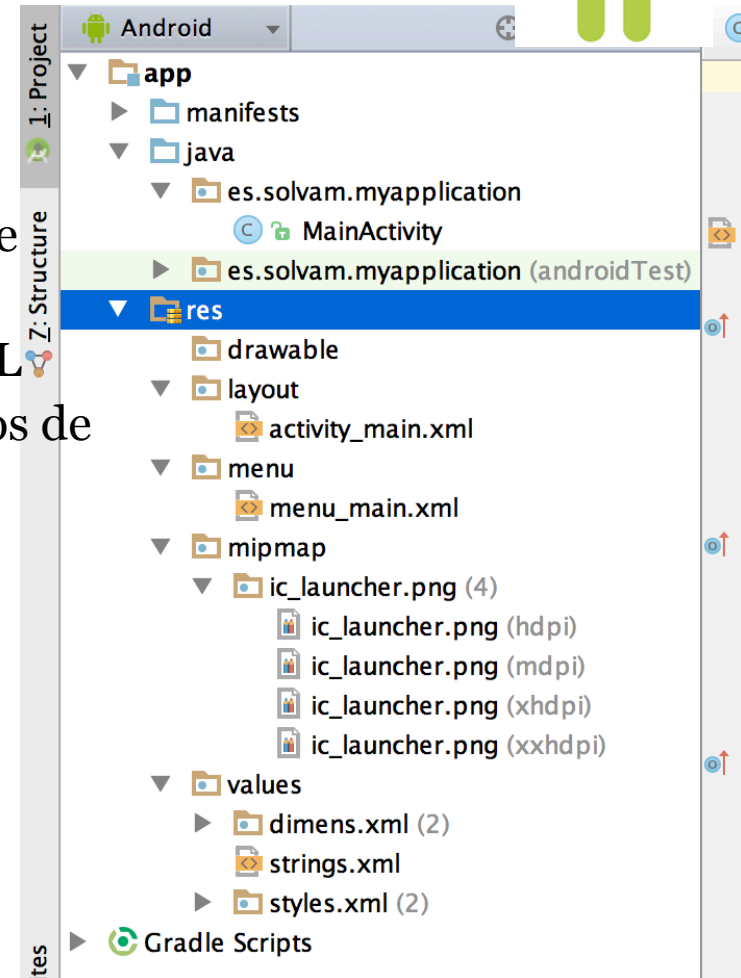


7. Elementos de un proyecto Android

Partes del proyecto

La carpeta de recursos /res:

- res/drawable: para imágenes (JPEG o PNG)
- res/layout: para especificaciones de formato de pantallas basadas en XML
- res/menu: para definir menús en formato XML
- res/raw: de propósito general (vídeos o ficheros de texto)
- res/values: para cadenas de caracteres
- res/xml: para otro tipo de ficheros XML que queremos empaquetar con el proyecto





7. Elementos de un proyecto Android

Partes del proyecto

- La carpeta de recursos /res:

The screenshot shows the Android Studio interface. On the left, the 'Project' view displays the project structure. The 'res' folder is expanded, showing subfolders like 'drawable', 'layout', 'menu', 'mipmap', and 'values'. The 'strings.xml' file is highlighted in the 'values' folder. A red box is drawn around the 'res' folder and its subfolders. Another red box is drawn around the 'strings.xml' file in the 'values' folder. A red arrow points from the 'strings.xml' file in the project structure to the 'strings.xml' file in the editor. The editor shows the 'strings.xml' file with the following XML code:

```
<resources>
  <string name="app_name">My Application</string>
  <string name="hello_world">Hello world!</string>
  <string name="action_settings">Settings</string>
</resources>
```

A green box highlights the XML code for two TextViews in the editor:

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello" />

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Y la he hecho yo solo, Jose" />
```



UD0. Introducción a Android

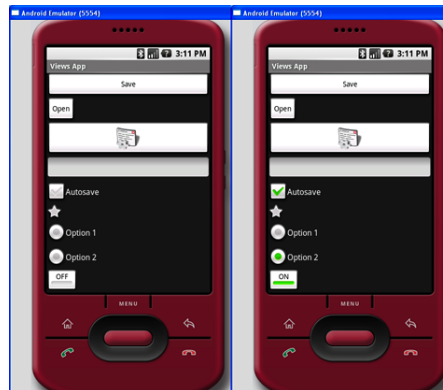
1. Introducción
2. La plataforma Android
3. Arquitectura de Android
4. Las versiones de Android y niveles de API
5. Creación de un primer programa
6. Ejecución del programa
7. Elementos de un proyecto de Android
8. **Componentes de una aplicación**

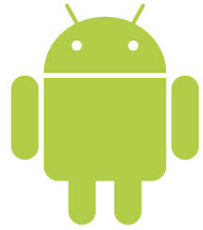


8. Componentes de Android

Vista (*View*)

- Las *vistas* son los elementos que componen **la interfaz de usuario** de una aplicación.
- Son por ejemplo, un botón, una entrada de texto,...
- Todas las vistas van a ser objetos descendientes de la clase ***View***, y por tanto, pueden ser definidos utilizando código **Java**.
- Sin embargo, lo habitual va a ser definir las vistas utilizando un fichero **XML** y **dejar que el sistema cree los objetos por nosotros** a partir de este fichero.
- Esta forma de trabajar es muy similar a la definición de una página web utilizando código HTML.

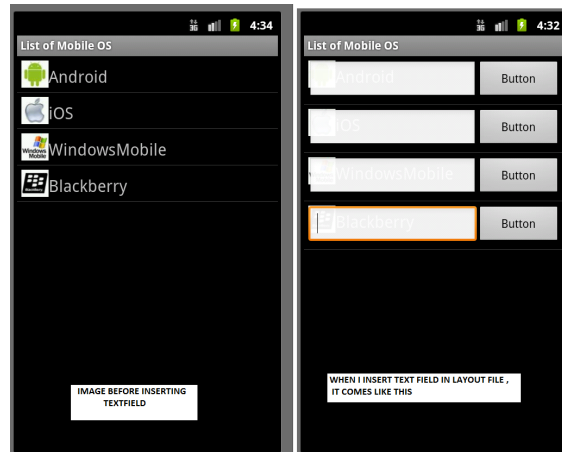




8. Componentes de Android

Layout

- Un Layout es un **conjunto de vistas** agrupadas de una determinada forma.
- Vamos a disponer de diferentes tipos de Layouts para organizar las vistas de forma lineal, en cuadrícula o indicando la posición absoluta de cada vista.
- Los Layouts también son objetos descendientes de la clase **View**.
- Igual que las vistas los Layouts pueden ser definidos en código, aunque la forma habitual de definirlos es utilizando código **XML**.

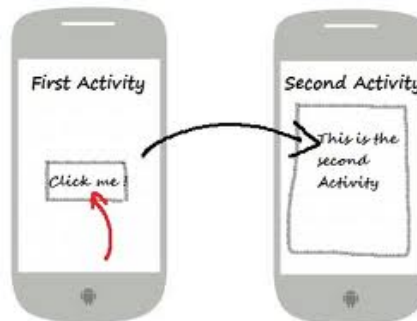




8. Componentes de Android

Actividad (*Activity*)

- Una aplicación en Android va a estar formada por un conjunto de elementos básicos de visualización, coloquialmente conocidos como pantallas de la aplicación.
- En Android cada uno de estos **elementos, o pantallas**, se conoce como ***actividad***.
- Su función principal es la **creación del interfaz de usuario**.
- Una aplicación suelen necesitar varias *actividades* para crear el interfaz de usuario.
- Las diferentes *actividades* creadas serán **independientes entre sí**, aunque todas trabajarán para un objetivo común.
- Toda actividad ha de pertenecer a una clase descendiente de ***Activity***.

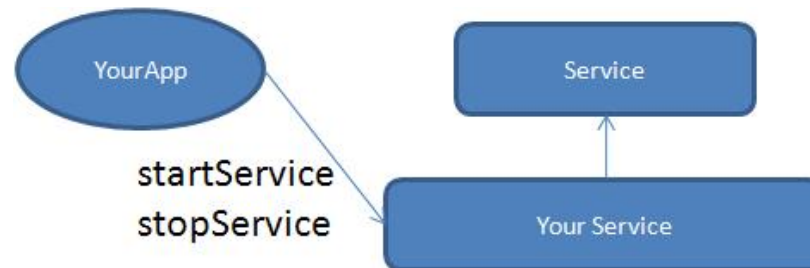


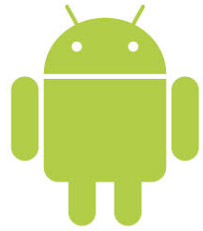


8. Componentes de Android

Servicio (*Service*)

- Un *servicio* es un proceso que **se ejecuta “detrás”**, sin la necesidad de una interacción con el usuario.
- Es algo parecido a un *demonio* en Unix o a un *servicio* en Windows.
- En Android disponemos de dos tipos de servicios:
 - **servicios locales**, que pueden ser utilizados por aplicaciones del mismo terminal
 - **servicios remotos**, que pueden ser utilizados desde otros terminales.





8. Componentes de Android

Intención (*Intent*)

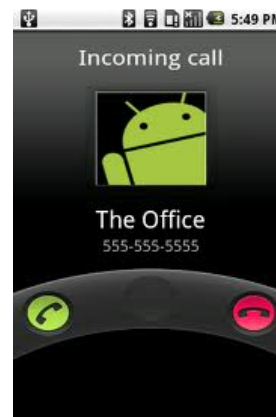
- Una *intención* representa la **voluntad** de realizar alguna **acción**; como realizar una llamada de teléfono, visualizar una página web.
- Se utiliza cada vez que queramos:
 - lanzar una *actividad*
 - lanzar un *servicio*
 - lanzar un *anuncio de tipo broadcast*
- Comunicarnos con un *servicio*
- Los componentes lanzados pueden ser internos o externos a nuestra aplicación. También utilizaremos las *intenciones* para el intercambio de información entre estos componentes.
- En muchas ocasiones una *intención* no será inicializada por la aplicación, si no por el sistema, por ejemplo, cuando pedimos visualizar una página web. En otras ocasiones será necesario que la aplicación inicialice su propia *intención*. Para ello se creará un objeto de la clase ***Intent***.



8. Componentes de Android

Receptor de anuncios (Broadcast receiver)

- Un *receptor de anuncios* recibe y reacciona ante anuncios de tipo broadcast.
- Existen muchos originados por el sistema; como por ejemplo *Batería baja*, *llamada entrante*,... Aunque, las aplicaciones también puede lanzar un *anuncio broadcast*.
- No tienen interfaz de usuario, aunque pueden iniciar una actividad para atender a un anuncio.





8. Componentes de Android

Proveedores de Contenido (*Content Provider*)

- La compartición de **información entre teléfonos móviles** resulta un tema vital.
- Android define un mecanismo estándar para que las aplicaciones puedan compartir datos sin necesidad de comprometer la seguridad del sistema de ficheros.
- Con este mecanismo podremos acceder a datos de otras aplicaciones, como la lista de contactos, o proporcionar datos a otras aplicaciones.

