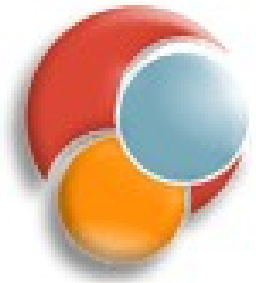


Desarrollo de Aplicaciones para Android

Sesión 1: Introducción a Android

Puntos a tratar

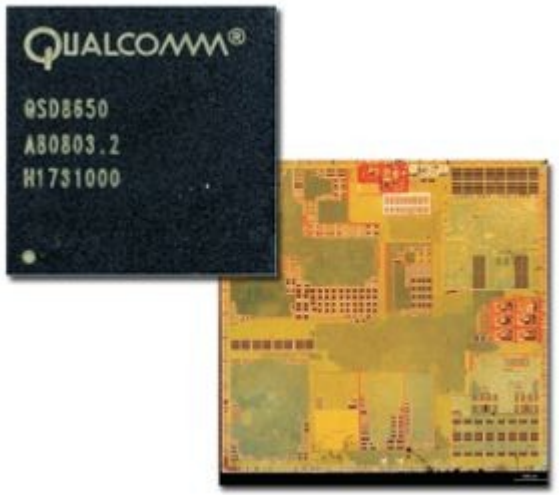
- Dispositivos móviles
- Historia de Android
- Desarrollo de aplicaciones
- Emulador
- AndroidManifest.xml
- Externalizar recursos
- Plug-in para Eclipse
- ¡Hola, Mundo!

Historia de los smartphones

- Primeros smartphones en los 90
  - Nokia communicator en 1996
- Primer smartphone con sistema operativo abierto en 2000:
  - Ericsson R380 con Symbian OS
- Smartphone actuales:
  - Pantalla táctil
  - Opcionalmente teclado físico
  - Sistema operativo con Market para aplicaciones
  - Sensores (GPS, equilibrio)
  - Cámaras y videoconferencia
  - Capacidad de almacenamiento y conectividad

Procesador

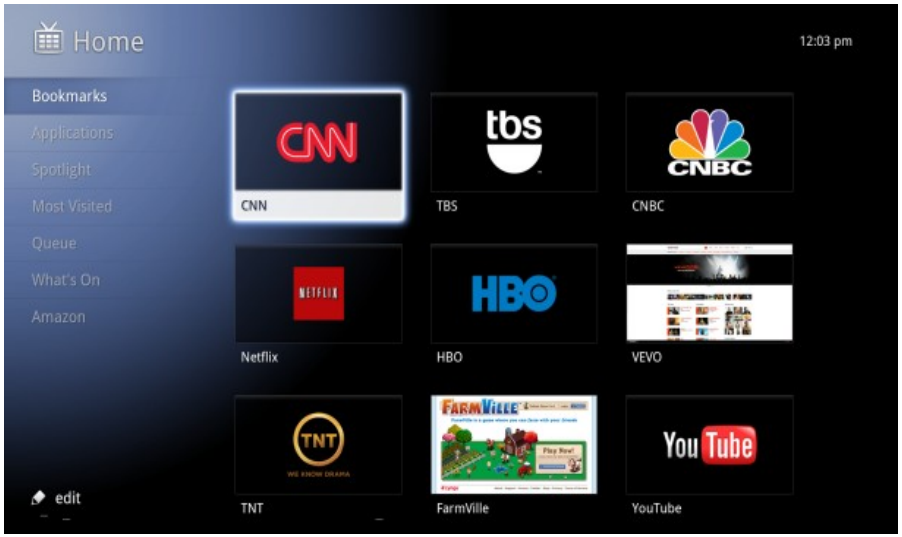
- Snapdragon de Qualcomm
  - Arquitectura ARM (en el 98% de los smartphones)
    - Tipo RISC: reduced instruction set computer
  - Instrucciones ARMv7
  - Plataforma “system on chip” que incluye:
    - Hasta dos CPU's de 1.5 G
    - HSPA+
    - GPS
    - Bluetooth
    - Video full definition
    - Wi-Fi
    - TV móvil



Procesador

- Apple A5
  - ARM Cortex-A9
  - 1 GHz ajustable
  - Fabricado por Samsung
  - Instrucciones ARMv7
  - Todos los componentes del “system on chip”

Dispositivos



Conectividad

- Los dispositivos deben conectarse para descargar las aplicaciones
  - Over The Air (OTA)
    - Conexión a Internet usando la red móvil (GSM, GPRS, UMTS)
  - Cable serie o USB
    - Conexión física
  - Infrarrojos
    - Los dispositivos deben tener contacto visual
  - Bluetooth
    - Ondas de radio (10 metros de alcance)
    - Alta velocidad (723kbit/s)

Redes de telefonía celular

- 1G: Red analógica
  - Sólo voz
  - Red TACS en España
  - Distintos países usan distintas redes
    - No permite itinerancia
- 2G: Red digital
  - Voz y datos
  - GSM (*Global System for Mobile communications*) en toda Europa
    - Permite itinerancia
  - Red no IP
    - Protocolos WAP (WSP)
    - Un gateway conecta la red móvil (WSP) a la red Internet (TCP/IP)
  - Conmutación de circuitos (*Circuit Switched Data*, CSD)
    - 9’6kbps
    - Se ocupa un canal de comunicación de forma permanente
    - Se cobra por tiempo de conexión

Redes de telefonía celular (2)

- 2,5G: GPRS (*General Packet Radio Service*)
  - Transmisión de paquetes
    - No ocupa un canal de forma permanente
    - Hasta 144kbps teóricamente (40kbps en la práctica)
    - Cobra por volumen de información transmitida
  - Se implementa sobre la misma red GSM
- 3G: Banda ancha
  - Red UMTS (*Universal Mobile Telephony System*)
    - Itinerancia global
  - Entre 384kbps y 2Mbps
  - Servicios multimedia
    - Videoconferencia, TV, música, etc
  - Transmisión de paquetes
  - Requiere nueva infraestructura

Redes de telefonía celular (3)

- 3.5G: HSDPA (High-Speed Downlink Packet Access)
  - mejora del scheduling de UTMS R99 y añade un nuevo canal downlink
  - Acceso a Internet con mayor ancho de banda y menor latencia: hasta 14Mbps
  - Variante HSPA+ hasta 84Mbps bajada y 22Mbps subida.
  - Terminales con HSDPA son compatibles con UTMS

Redes de telefonía celular (4)

- 4G LTE (Long term evolution)
  - Nuevo estándar de la norma 3GPP UMTS
  - Hasta 326,5Mbps de bajada y 86,5Mbps subida
  - Ancho de banda adaptativo entre 1.4 y 20 MHz
  - Celdas de 5Km óptimo, hasta 100Km
  - Alto rendimiento en movimiento hasta 15 Km/h. Conexión posible hasta 500Km/h
- 4G Basada completamente en protocolo IP
  - Abandonaría el acceso radio de UMTS, controlándolo por software
  - Implantada en Japón desde diciembre de 2011
- 5G: No está definido. Ideas: ubiquituous computing, conectado a la vez a diferentes fuentes de datos, smart-radio, servicio desde vehículos aéreos

Paradigmas de programación en móviles

- Documentos Web
  - Descarga documentos y los muestra en un navegador
  - Formato adecuado para móviles (WML, XHTML, ...)
  - Requiere conectar a red para descargar cada documento
  - Velocidad de descarga lenta
  - Documentos pobres (deben servir para todos los móviles)
- Aplicaciones locales
  - La aplicación se descarga en el móvil
  - Se ejecuta de forma local
  - Interfaz de usuario más flexible
  - Puede funcionar sin conexión (minimiza el tráfico)

Documentos Web

- WML (Wireless Markup Language)
  - Forma parte de los protocolos WAP (Capa de aplicación, WAE)
  - Lenguaje de marcado dirigido a móviles
  - Requiere aprender un nuevo lenguaje diferente a HTML
  - Documentos muy pobres
- iMode
  - Documentos escritos en cHTML (HTML compacto)
    - Subconjunto de HTML
    - Propietario de NTT DoCoMo
  - Sobre la red japonesa PDC-P (extensión de la red japonesa PDC, similar a GSM, para transmisión de paquetes)
    - En Europa se lanza sobre GPRS
- XHTML MP
  - Versión reducida de XHTML dirigido a móviles
  - A diferencia de cHTML, se desarrolla como estándar

Aplicaciones locales

- Sistema operativo
  - Symbian OS, Palm OS, Windows Pocket PC, Windows Mobile, Android, iOS, etc
  - Poco portable
  - Requiere aprender nuevas APIs
- Runtime Environments
  - BREW
    - Soportado por pocos dispositivos
    - Requiere aprender una nueva API
  - Java ME (J2ME)
    - Soportado por gran cantidad de dispositivos
    - Existe una gran comunidad de desarrolladores Java

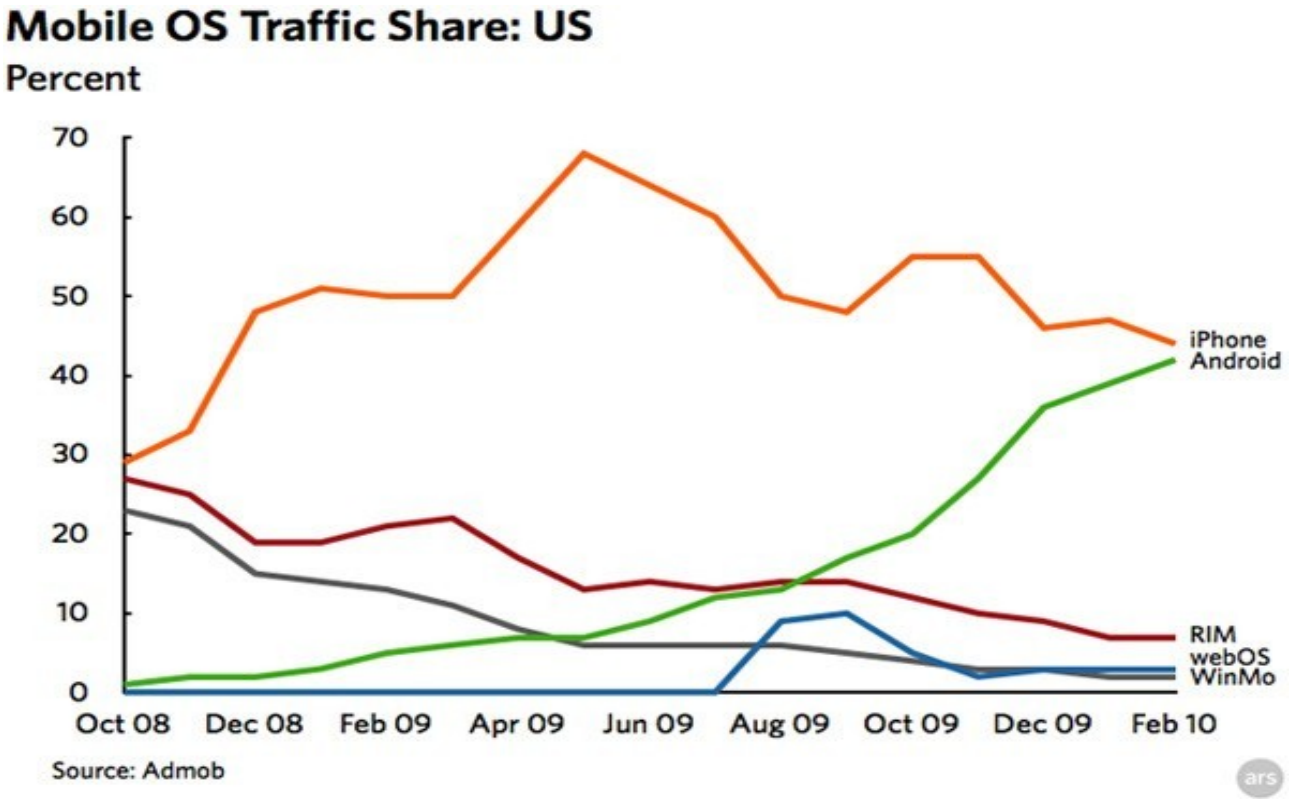
¿Sistema operativo o entorno de ejecución?

- Los SO aprovechan mejor el dispositivo y ofrecen mejor acceso al hardware y periféricos
- Los smartphones usan SO
- Los SO más extendidos entre los smartphones son iOS y Android
  - El usuario es libre de usar cualquiera de ellos
  - El desarrollador se ve obligado a desarrollar para ambas plataformas (y otras) para llegar a todos
- Plataforma de ejecución
  - no dependemos del sistema operativo
  - hay más de 2 mil millones de dispositivos con JavaME

Sistemas operativos

- Android (open source)
- BlackBerry OS de RIM (propietario)
- iOS de Apple (propietario)
- Symbian OS (open source)
- Windows Phone (propietario)
- webOS de HP (algunas partes abiertas)
- QNX de RIM (propietario)
- SHR (basado en linux)
- bada de Samsung (propietario)
- Brew de Qualcomm

Sistemas operativos antes de 2010

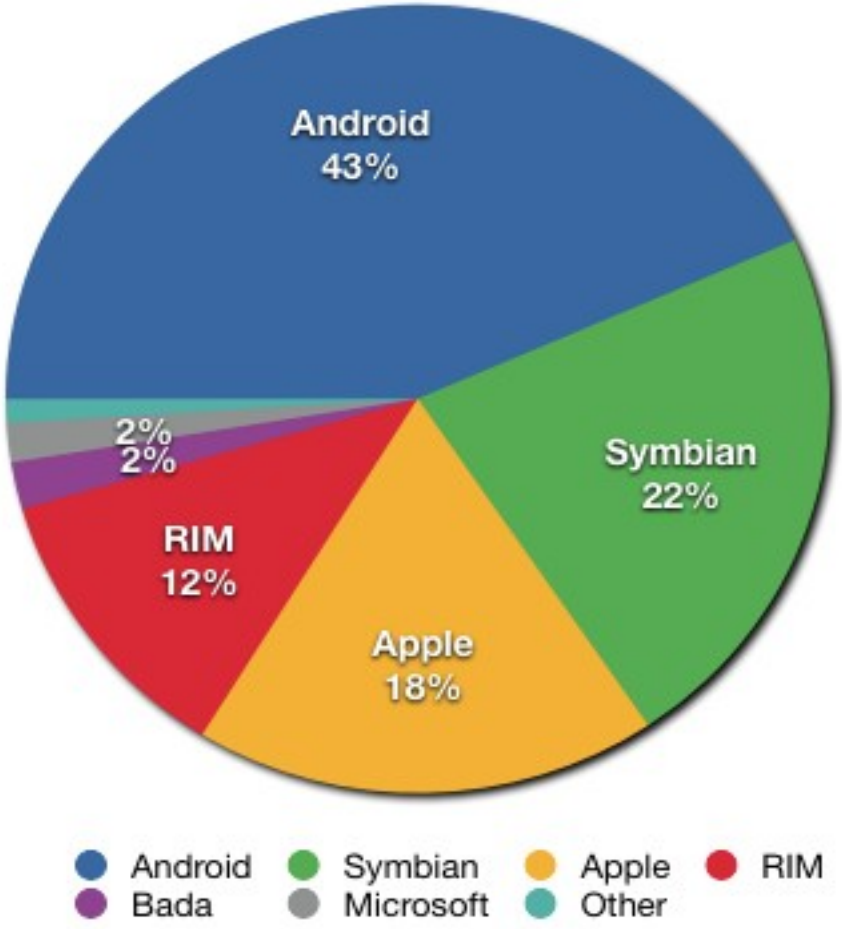


Sistemas operativos en el mercado actual

- Estadísticas de uso según Gartner

Year	Symbian	Android	RIM	iOS	Microsoft	Other OSs
2011 Q1	27.4%	36.0%	12.9%	16.8%	3.6%	3.3%
2010	37.6%	22.7%	16.0%	15.7%	4.2%	3.8%
2009	46.9%	3.9%	19.9%	14.4%	8.7%	6.1%
2008	52.4%	0.5%	16.6%	8.2%	11.8%	10.5%
2007	63.5%	N/A	9.6%	2.7%	12.0%	12.1%

Sistemas operativos en el mercado actual



Android

- Sistema operativo para dispositivos móviles
- Núcleo basado en el de Linux
- Programación de aplicaciones en Java

Historia: motivación

- Los dispositivos embebidos se programaban a bajo nivel: necesidad de entender todo su hardware
- Sistemas operativos: abstracción del hardware
  - Ej: Symbian. Código en C/C++. Nivel medio/bajo, bibliotecas propietarias, complicaciones con hardware específico.
- Java ME: abstrae del HW y del SO. Máquina virtual limita el acceso a hardware.

Historia: android

- Android 1.1 se publica en febrero de 2009 (coincide con la proliferación de smartphones táctiles).
- Siguientes versiones:
  - 1.5 Cupcake (basada en el núcleo de Linux 2.6.27)
  - …
  - 2.2 Froyo (basada en el núcleo de Linux 2.6.32)

Nombres basados en repostería

Nombres basados en repostería

- 2.3 Gingerbread

Nombres basados en repostería

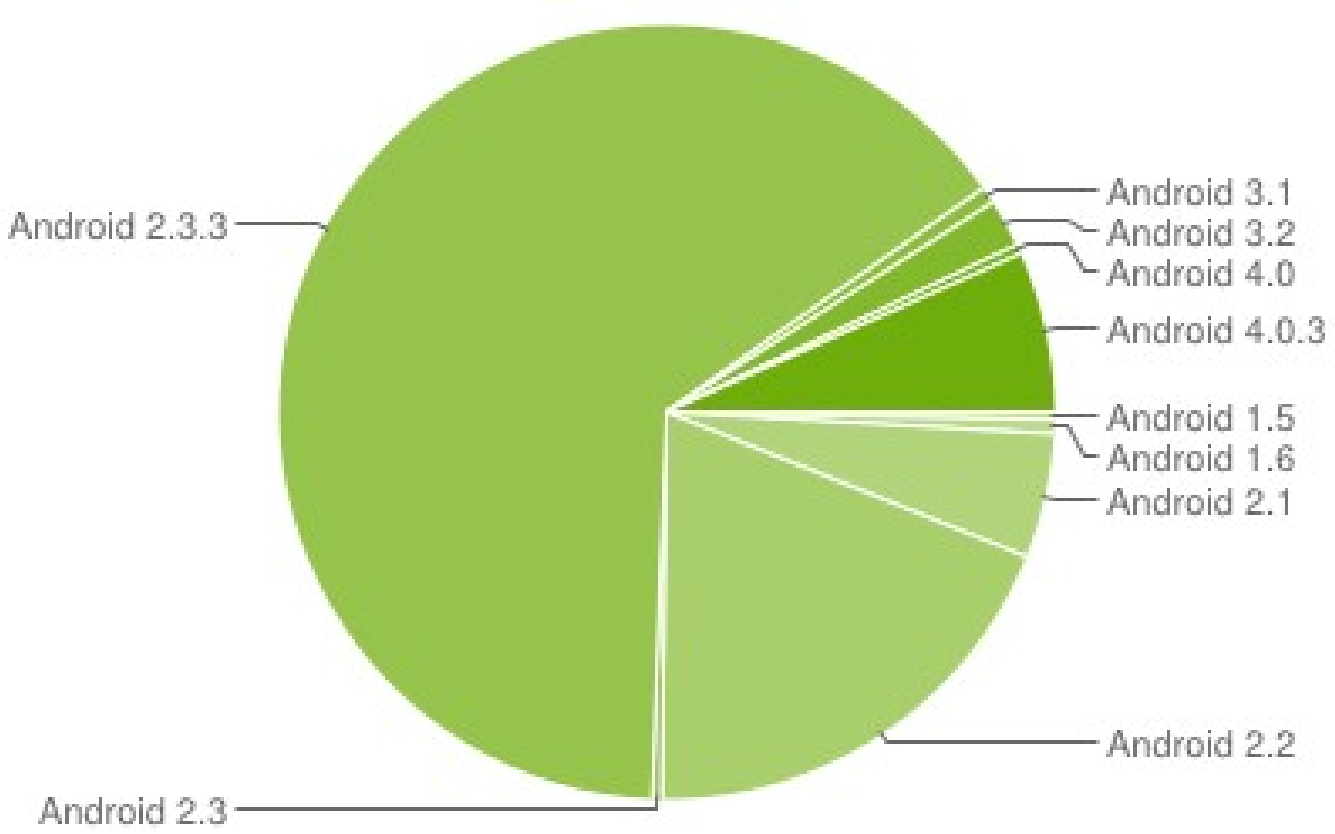
- 3.0 / 3.1 Honeycomb (orientado a tablets)

Nombres basados en repostería

- 4.0 Ice Cream Sandwich (tablets / móviles)

Estadísticas de uso de versiones

- Junio de 2012



Licencia

- Android
  - Sistema operativo (Licencia Apache)
  - Plataforma de desarrollo (Licencia Apache)
- Licencia Apache
  - Open Source
  - Permite a los fabricantes añadir extensiones propietarias sin ponerlas en manos de la comunidad del software libre

Open Source

- El open source hace posible:
  - Una comunidad de desarrollo, gracias a sus completas APIs y documentación ofrecida.
  - Desarrollo desde cualquier plataforma (Linux, Mac, Windows, etc).
  - Un sistema operativo para cualquier tipo de dispositivo móvil, al no estar diseñado para un sólo tipo de móvil.
  - Posibilidad para cualquier fabricante de diseñar un dispositivo que trabaje con Android, y la posibilidad de abrir el sistema operativo y adaptarlo o extenderlo para su dispositivo.
  - Valor añadido para los fabricantes de dispositivos: las empresas se ahorran el coste de desarrollar un sistema operativo completo para sus dispositivos.
  - Valor añadido para los desarrolladores: los desarrolladores se ahorran tener que programar APIs, entornos gráficos, aprender acceso a dispositivos hardware particulares, etc.

¿De qué está hecho?

- Núcleo (branch) basado en linux (memoria, procesos, hardware)
- Bibliotecas open source (SQLite, WebKit, OpenGL, manejador de medios, etc).
- Entorno de ejecución Dalvik
- Framework de desarrollo: pone a disposición de las aplicaciones los servicios del sistema
- SKD: herramientas, plug-in para Eclipse, emulador, ejemplos, doc.
- Interfaz de usuario para pantalla, dispositivos de entrada, etc.
- Aplicaciones preinstaladas (destacamos Flash Player)
- Android Market

Desarrollan Android:

- Open Handset Alliance
  - Trata de definir estándares abiertos para dispositivos móviles
  - Consorcio de decenas de compañías (entre ellas está Google):
    - Operadores de telefonía móvil
    - Fabricantes de dispositivos
    - Fabricantes de procesadores y microelectrónica
    - Compañías de software
    - Compañías de comercialización

Cuestiones éticas

- Aspectos positivos
  - Código abierto
    - Valor añadido para todos
    - Mantenibilidad
    - Seguridad informática
    - Transparencia del uso de sensores
  - Servicios gratuitos de Google

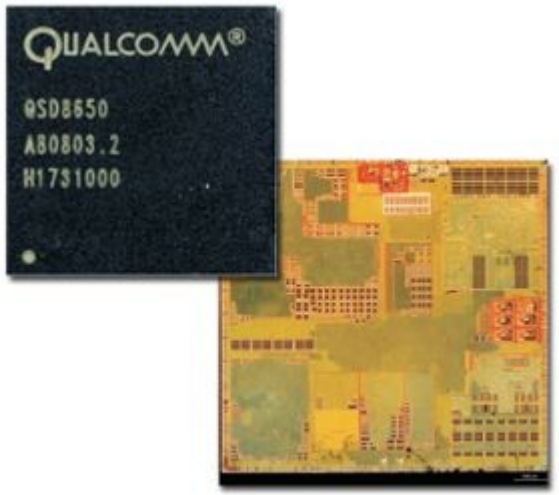
Cuestiones éticas

- Aspectos negativos
  - Obligatorio log-in con el ID de Google (dependencia)
  - Constante intercambio de datos con Google
  - Envío de localización (desactivable)
  - Los usuarios no están concienciados de la dependencia de los servidores de Google y lo aceptan sin más.
  - Aunque el SO sea libre, gran parte de su valor está en los servicios gratuitos de Google, que no son libres ni conocemos el tratamiento que dan a nuestra información privada.

Dispositivos

Procesador

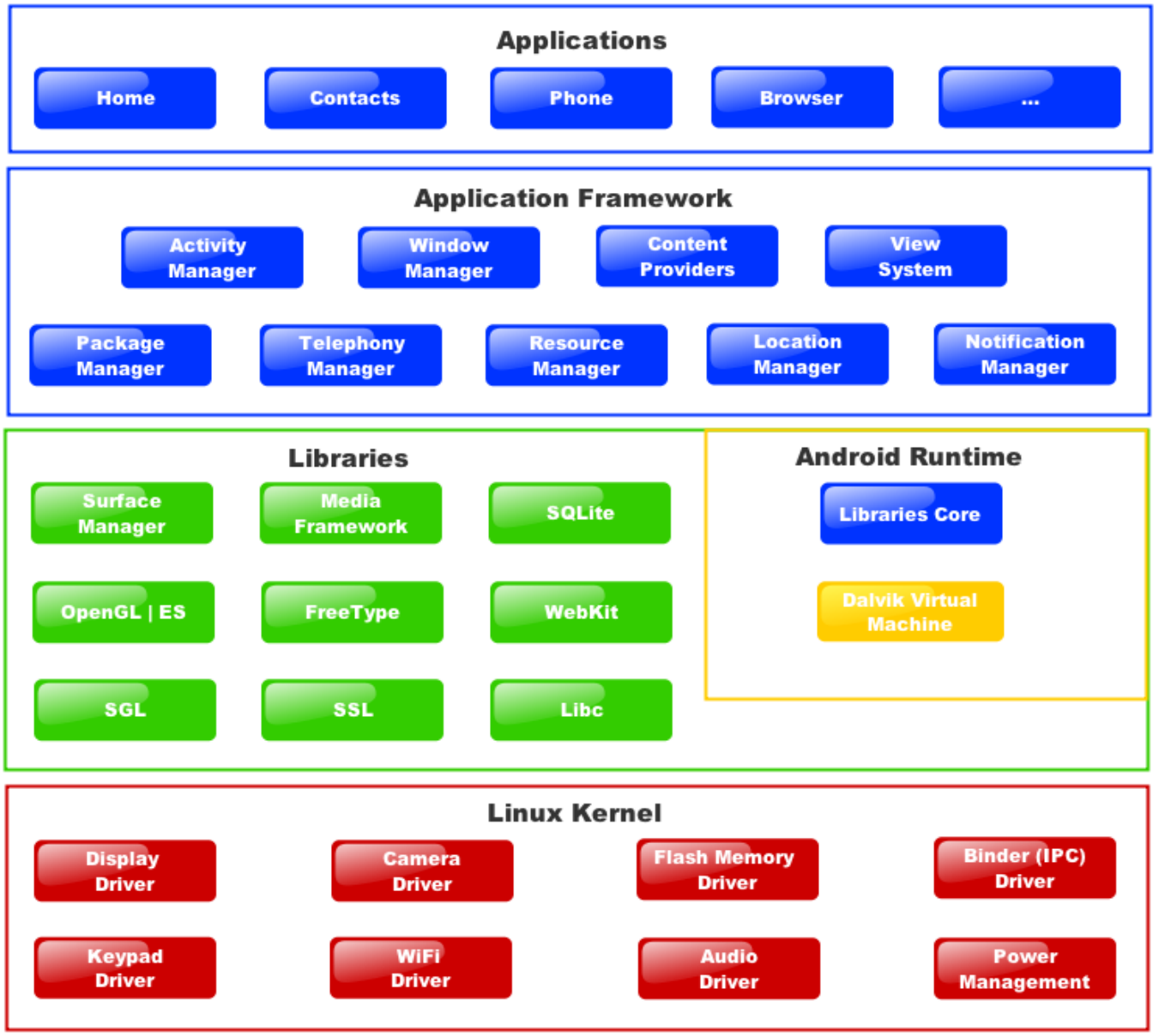
- Snapdragon de Qualcomm
  - Arquitectura ARM (el 98% de los móviles la usan)
    - Tipo RISC: reduced instruction set computer
  - Plataforma que incluye:
    - Hasta dos CPU's de 1.5 Ghz
    - HSPA+
    - GPS
    - Bluetooth
    - Video full definition
    - Wi-Fi
    - TV móvil



Android SDK

- Licencias, distribución y desarrollo gratuitos, tampoco hay procesos de aprobación del software. **No diferencia entre aplicaciones nativas y de terceros.**
- Acceso al hardware de WiFi, GPS, Bluetooth y telefonía, permitiendo realizar y recibir llamadas y SMS.
- Control completo de multimedia, incluyendo la cámara y el micrófono.
- APIs para los sensores: acelerómetros y brújula.
- **Mensajes entre procesos (IPC).**
- Almacenes de datos compartidos, **proveedores de contenidos**, SQLite, acceso a SD Card.
- **Aplicaciones y procesos en segundo plano.**
- **Widgets para la pantalla de inicio (escritorio).**
- Integración de los resultados de búsqueda de la aplicación con los del sistema.
- **Uso de mapas y sus controles desde las aplicaciones.**
- Aceleración gráfica por hardware, incluyendo OpenGL ES 2.0 para los 3D.

Capas



Tipos de aplicaciones

- Primer plano (activities)
- Segundo plano
  - Servicios puros
  - Servicios combinados con actividades
- Widgets de escritorio

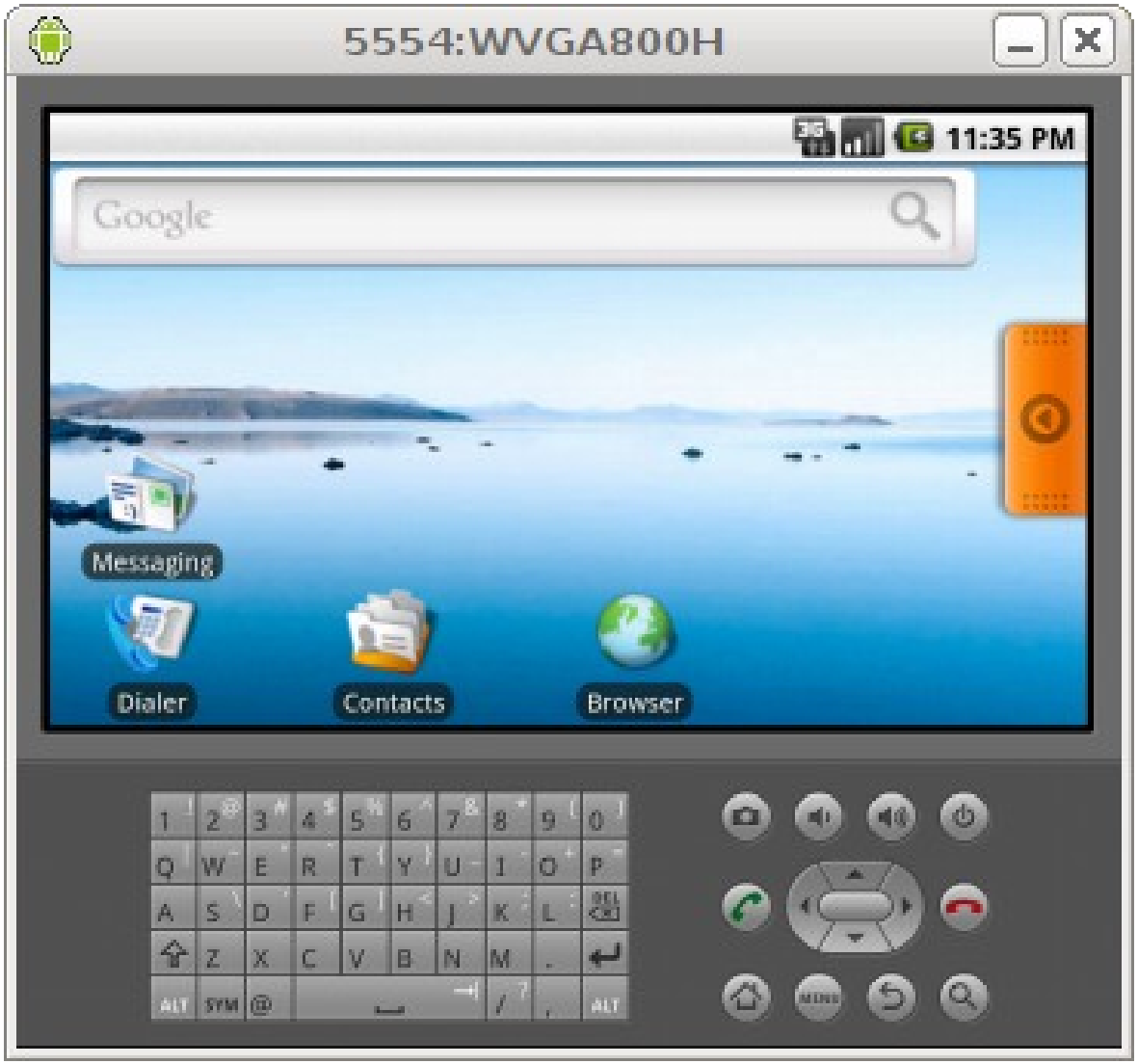
Consideraciones para el desarrollo

- Pequeña capacidad de procesamiento
- Memoria RAM limitada
- Memoria permanente de poca capacidad
- Pantallas pequeñas de poca resolución
- Transferencias de datos costosa (en términos de energía y económicos) y lenta
- Inestabilidad de las conexiones de datos
- Batería muy limitada
- Necesidad de terminar la aplicación en cualquier momento

Consideraciones para el desarrollo

- Ser eficiente
  - CPU
  - Memoria
  - Recursos y red
- Respetar al usuario
  - No robar el foco
  - Pocos avisos
  - Interfaz intuitiva y coherente con Android
- Ver Guía de desarrollo de Android antes de publicar

Emulador



Emulador

- Terminal al emulador:
  - `telnet localhos 5554`
- AVD (Android Virtual Device) tiene que ser especificado (características hardware a emular).
- Nivel de API
- Emulación de SD card

AndroidManifest.xml

- Presente en todo proyecto Android.
- Declaración de metadatos de la aplicación
  - Nombre del paquete y de la aplicación
  - Actividades, actividad principal
  - Servicios
  - Receptores broadcast
  - Proveedores de contenidos
  - Permisos
  - API mínima, librerías.

AndroidManifest.xml

- Estructura

```
<?xml version="1.0" encoding="utf-8"?>
<manifest>
    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <instrumentation />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />
    <application>
        <activity>
            <intent-filter>
                <action />
                <category />
                <data />
            </intent-filter>
            <meta-data />
        </activity>
        <activity-alias>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </activity-alias>
        <service>
            <intent-filter> . . . </intent-filter>
            <meta-data/>
        </service>
        <receiver>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </receiver>
        <provider>
            <grant-uri-permission />
            <meta-data />
        </provider>
        <uses-library />
    </application>
</manifest>
```

AndroidManifest.xml

- Ejemplo

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="es.ua.jtech.ajdm.interfaces"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".Interfaces"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
          <activity android:name="SubActividad" android:label="SubActividad">
          </activity>
</application>
    <uses-sdk android:minSdkVersion="8" />
</manifest>
```

Externalizar recursos

- Hace la aplicación más mantenible y personalizable
- Adaptación a otros idiomas
- Carpeta `res` del proyecto
  - `res/values`
  - `res/drawable-ldpi`
  - `res/drawable-mdpi`
  - `res/drawable-hdpi`
  - `res/layout`
  - `res/xml`

Externalizar recursos

- Valores, en formato XML:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
	<string name="saludo">¡Hola!</string>
	<color name="verde_transparente">#7700FF00</color>
	<dimen name="altura_mifuente">12sp</dimen>
	<array name="ciudades">
		<item>Alicante</item>
		<item>Elche</item>
		<item>San Vicente</item>
	</array>
	<style name="EstiloTexto1">
		<item name="android:textSize">18sp</item>
		<item name="android:textColor">#00F</item>
	</style>
	...
</resources>
```

Externalizar recursos

- Acceso a los recursos desde el código

```
TextView tv = (TextView)findViewById(R.id.TextView01);
tv.setText(R.string.saludo);
```

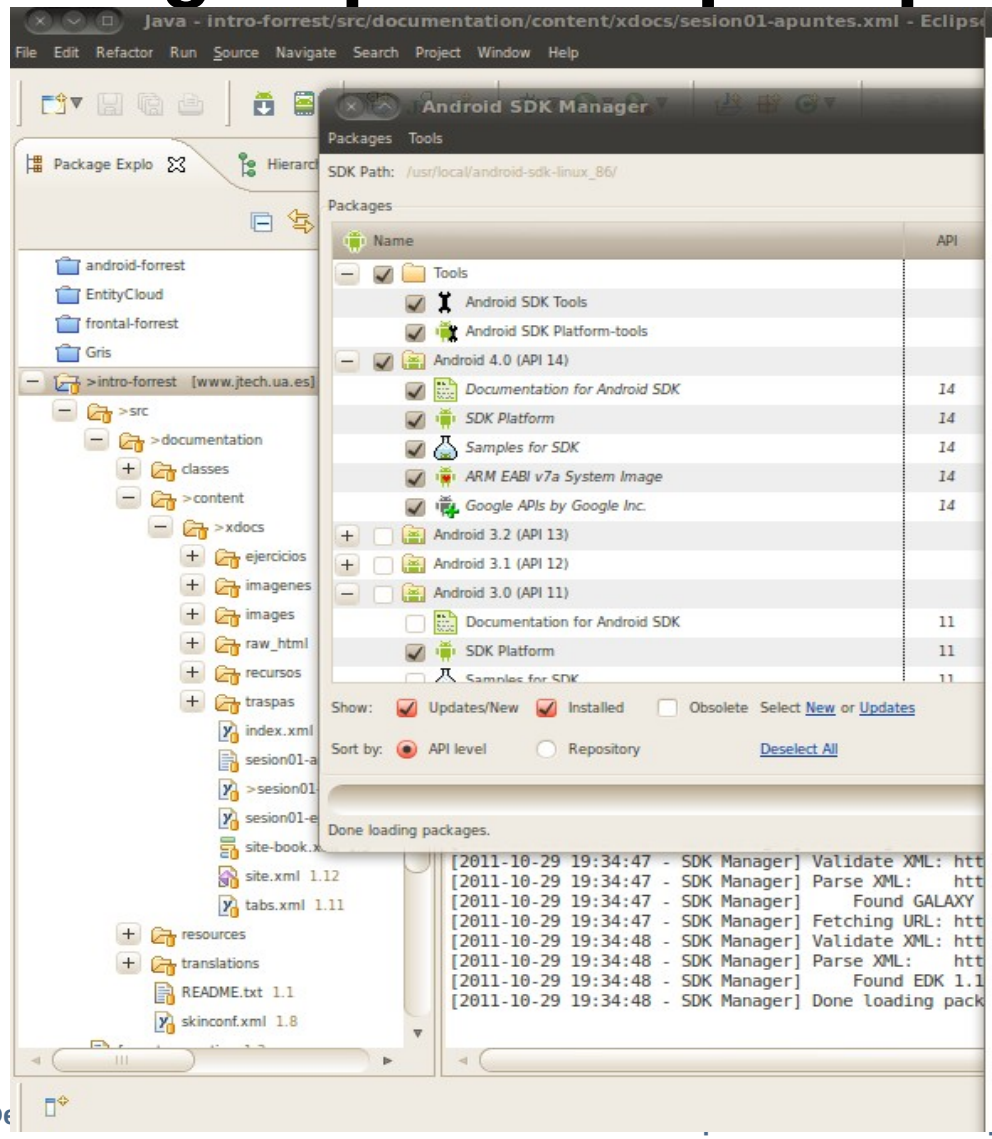
- Acceso a los recursos desde el XML

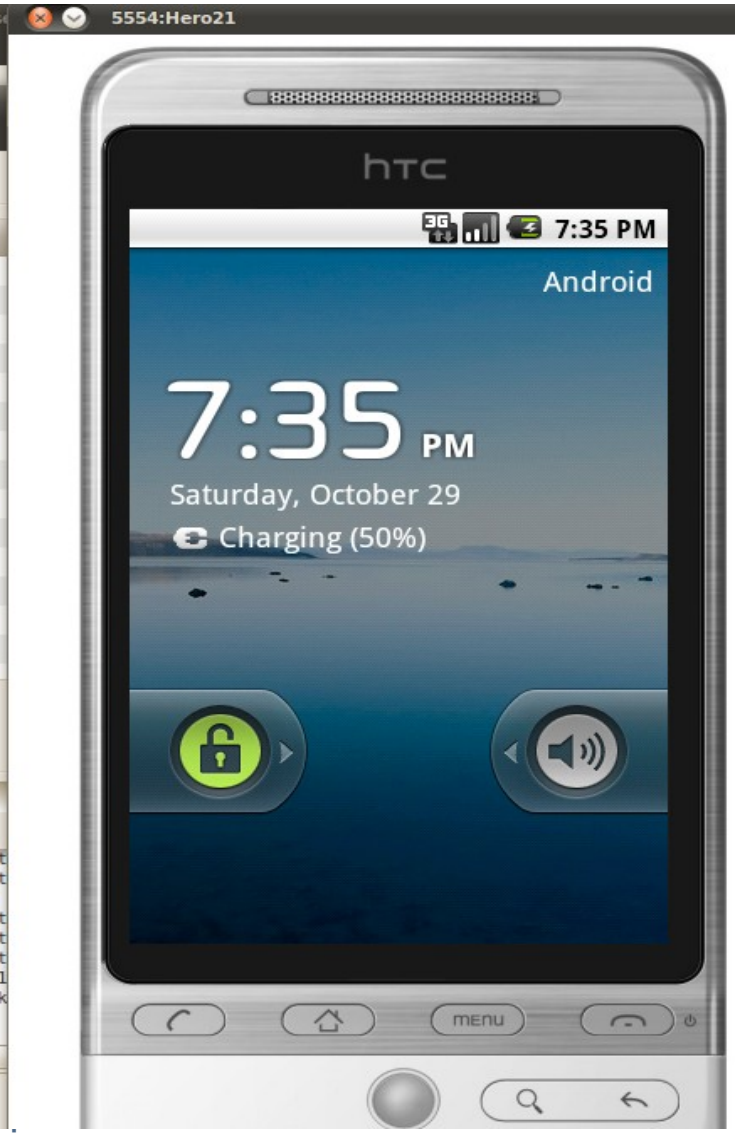
```
@string/nombrestring
```

Plug-in para Eclipse

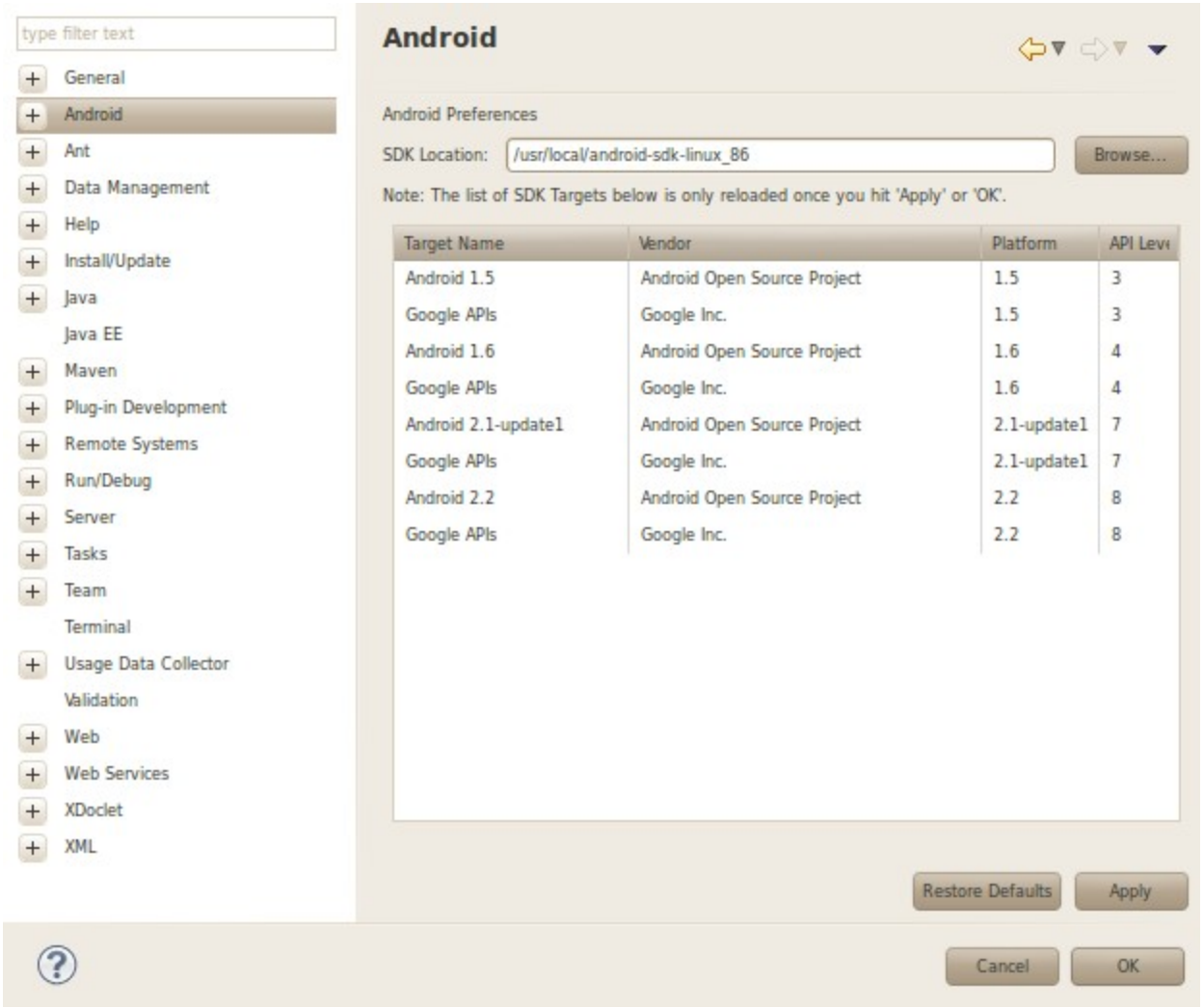
- Instalación desde Eclipse:
  - Help > Install new software > Available software > Add:
    - https://dl-ssl.google.com/android/eclipse/
  - Ok, seleccionar el software, Next, Finish.
  - Reiniciar Eclipse.
- Configuración:
  - Windows > Preferences > Android > SDK Location:
    - Indicamos la ruta del Android SDK que deberemos haber bajado aparte y descomprimido.

Plug-in para Eclipse: plataformas



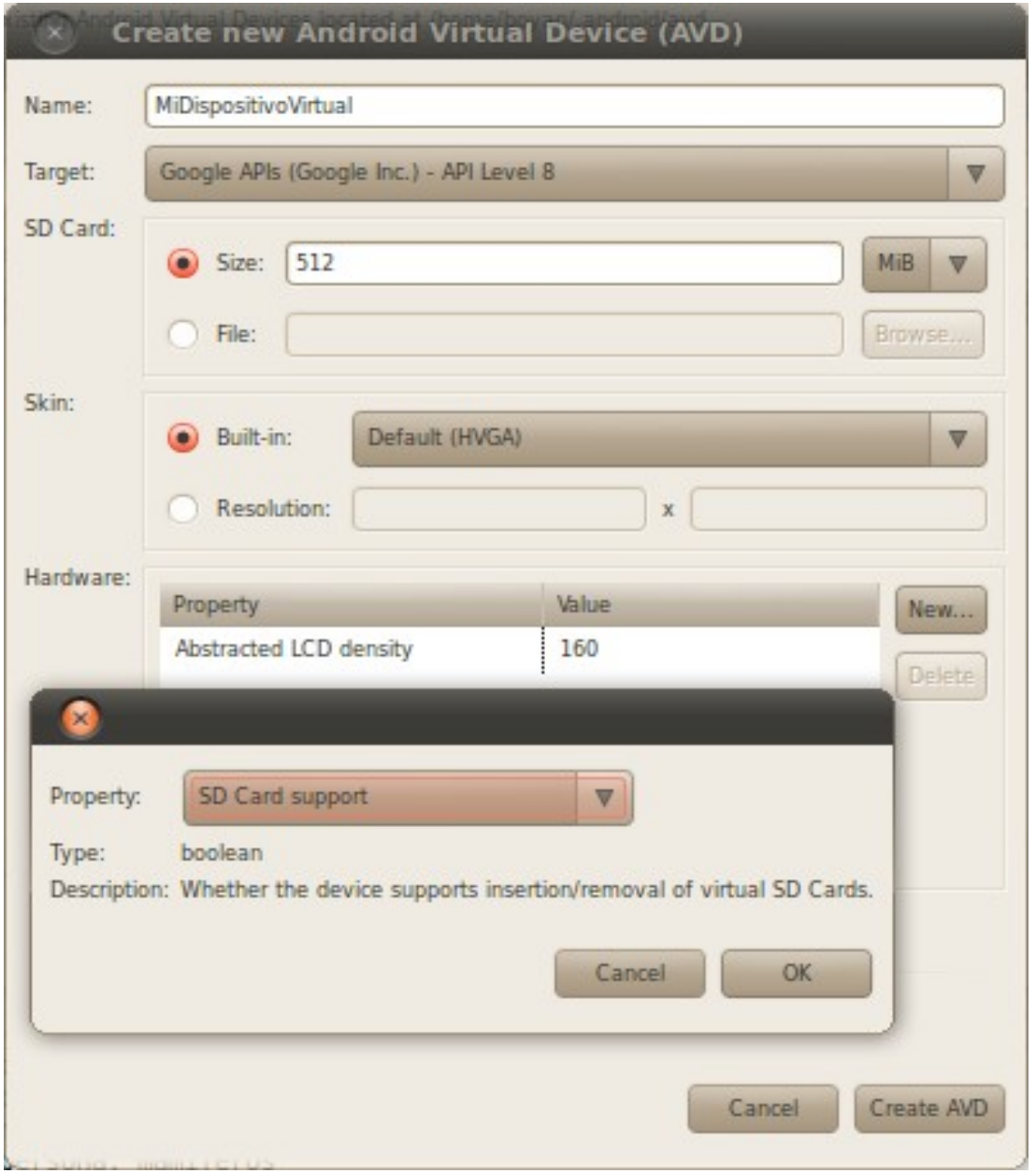


Plug-in para Eclipse: plataformas



AVD Manager

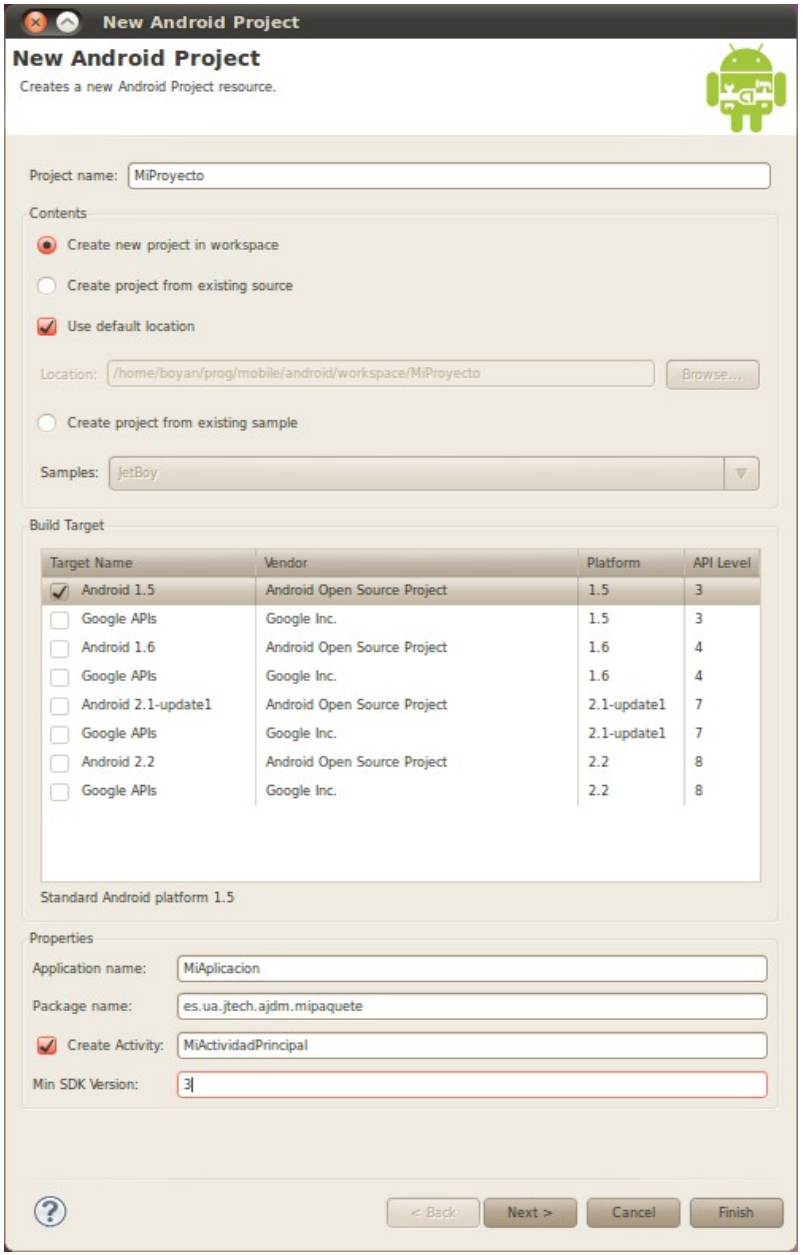
- Crear nuevo dispositivo virtual (AVD):



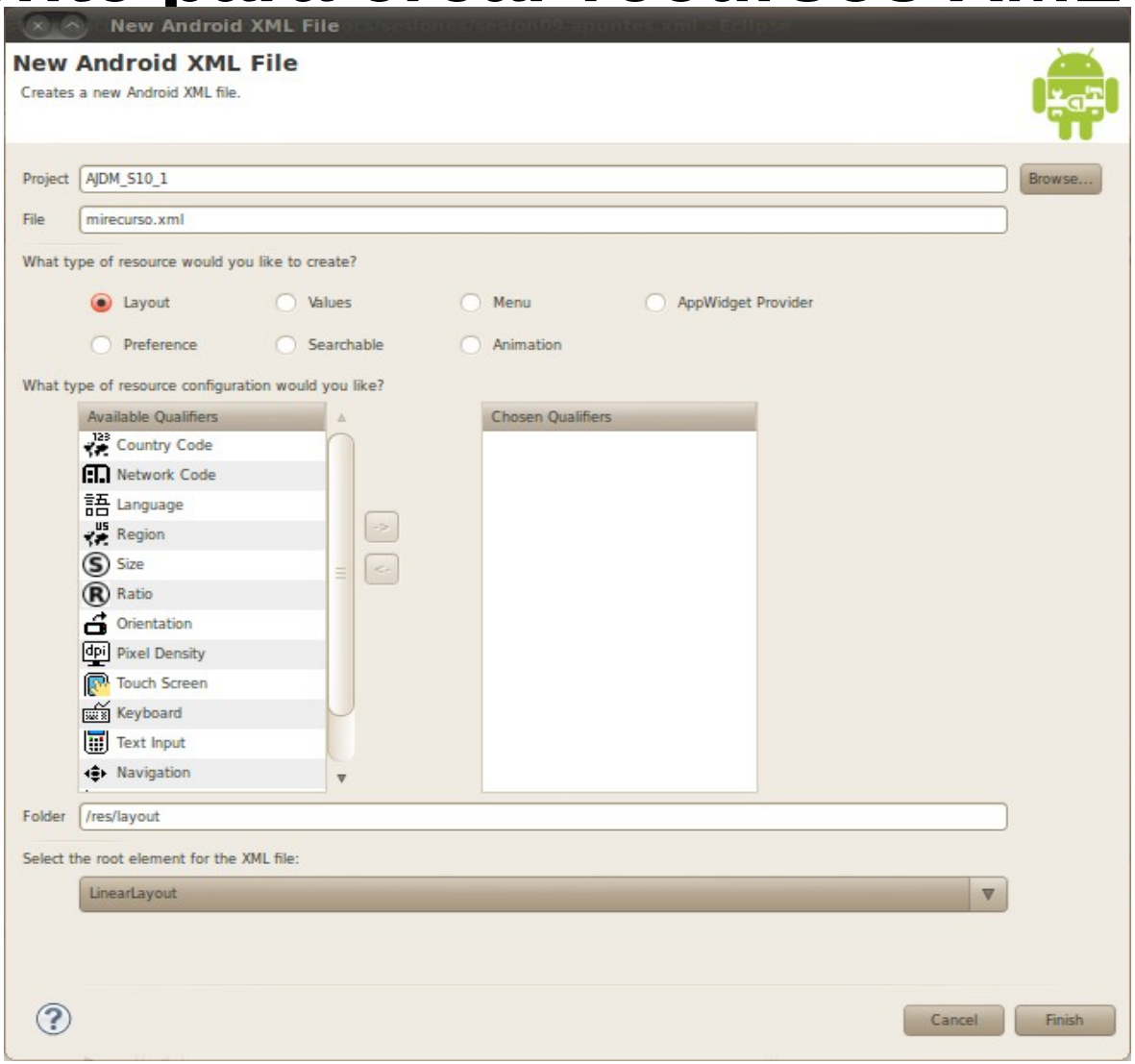
Emulador

Asistente para la creación de proyectos

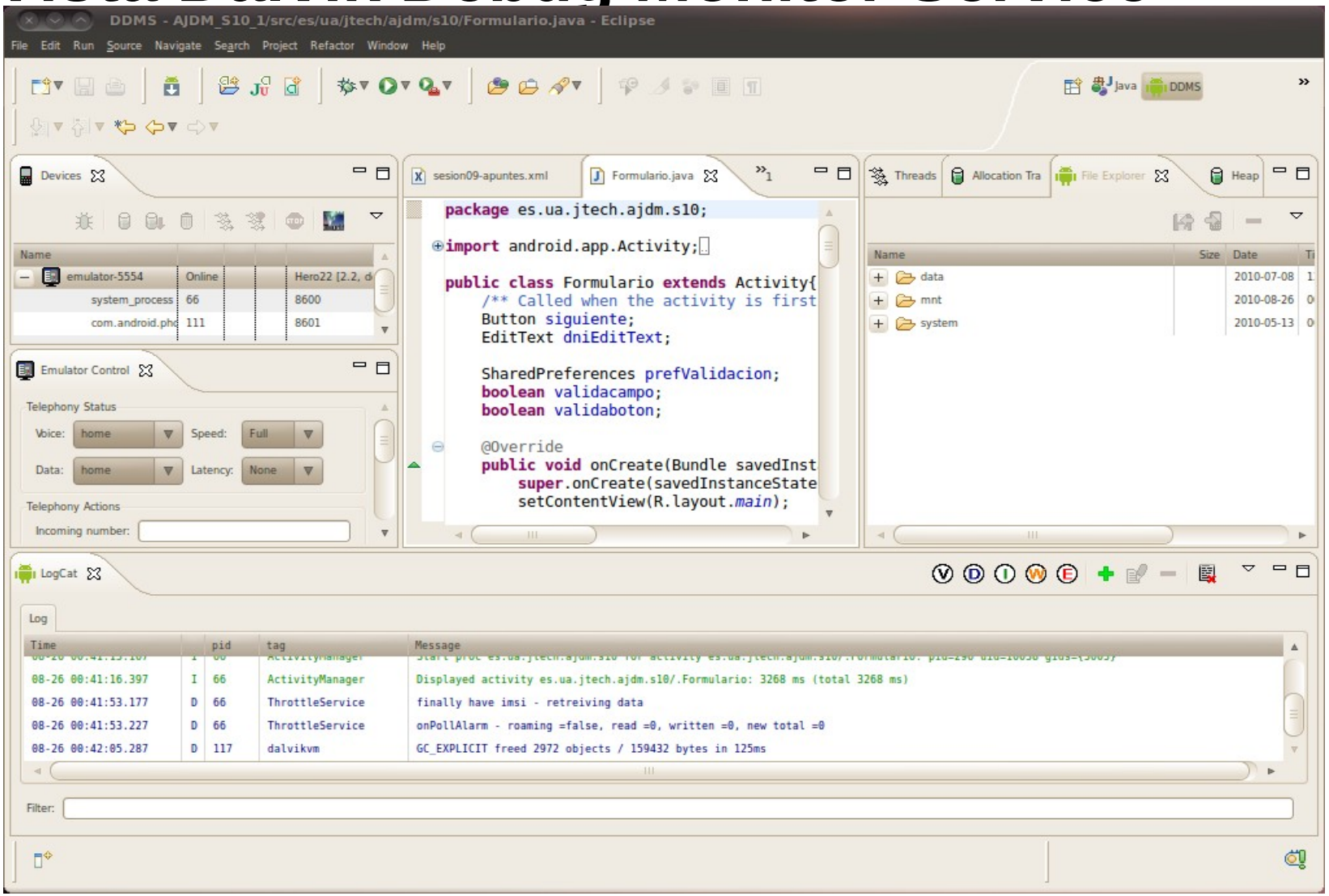
- Genera la estructura básica del proyecto
- AndroidManifest.xml
- Actividad principal
- Layout
- Resto de recursos



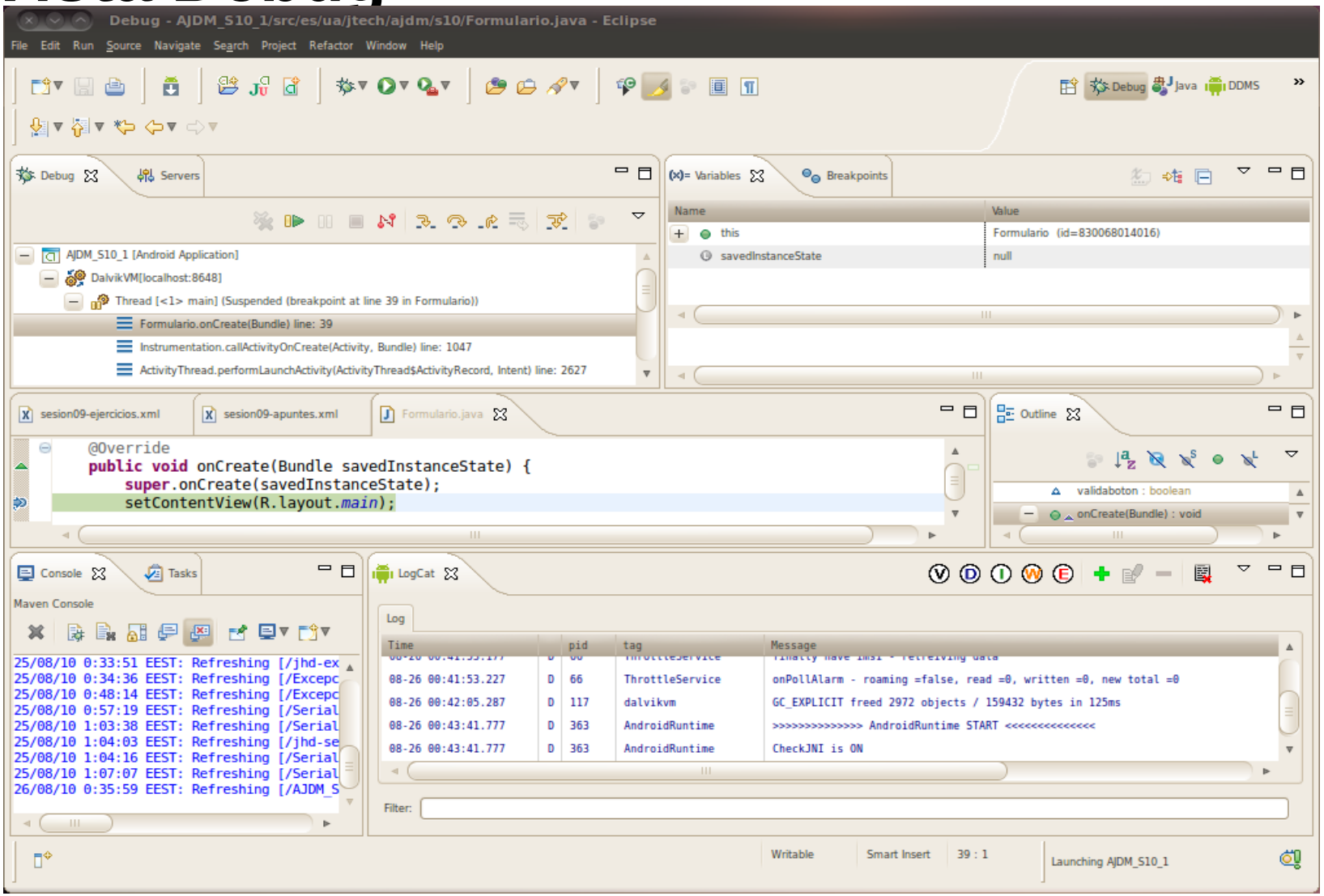
Asistente para crear recursos XML



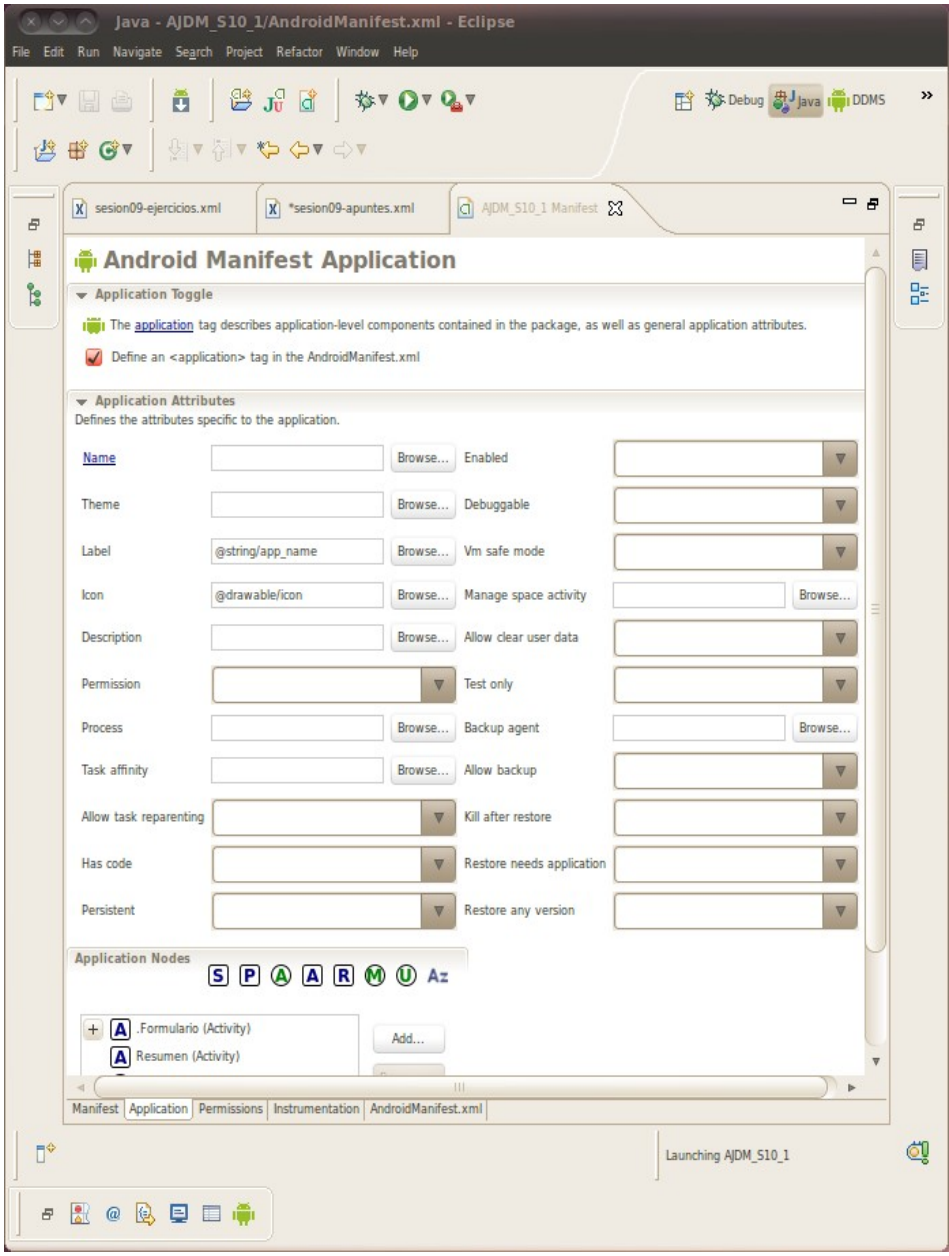
Vista Dalvik Debug Monitor Service



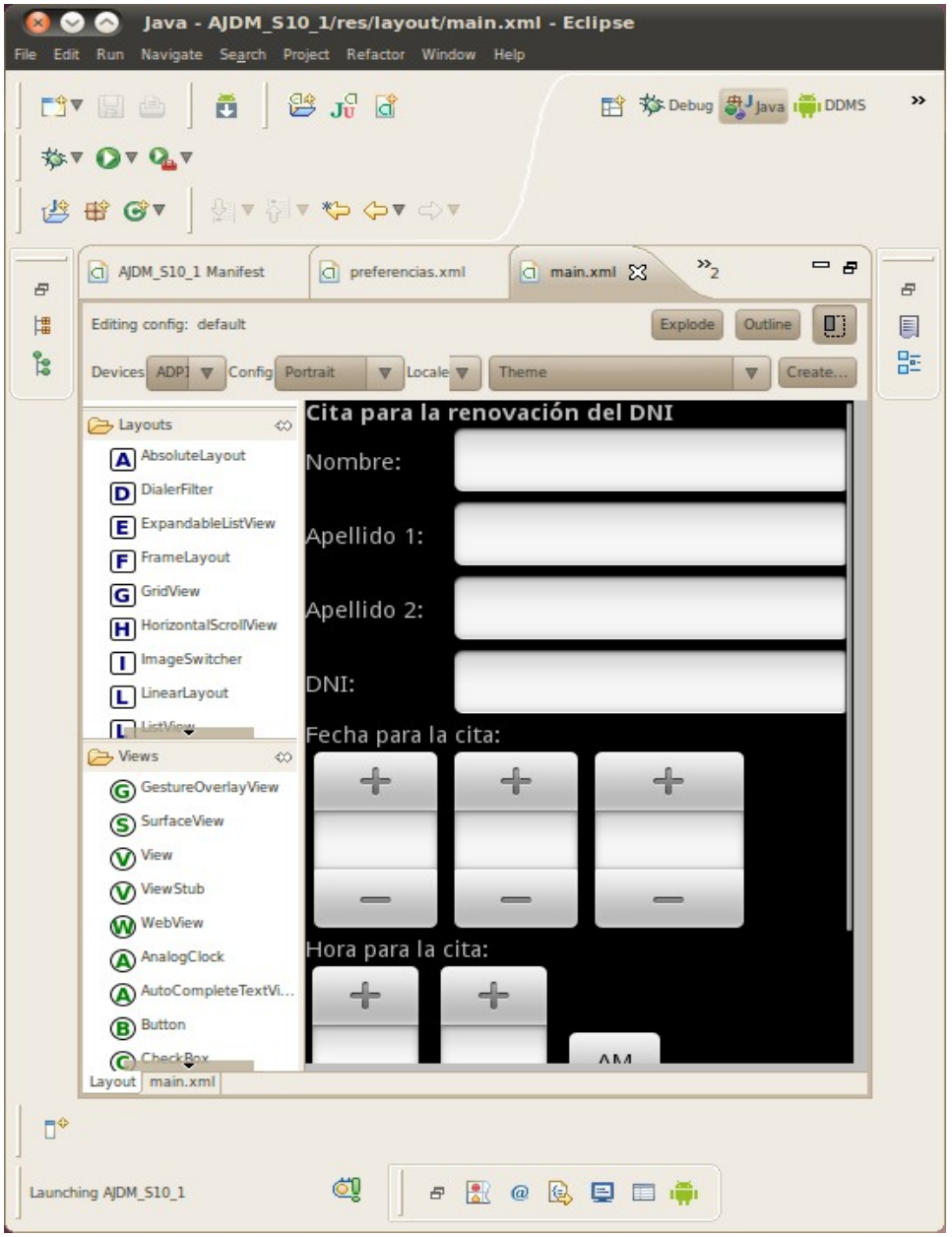
Vista Debug



Editor del AndroidManifest.xml

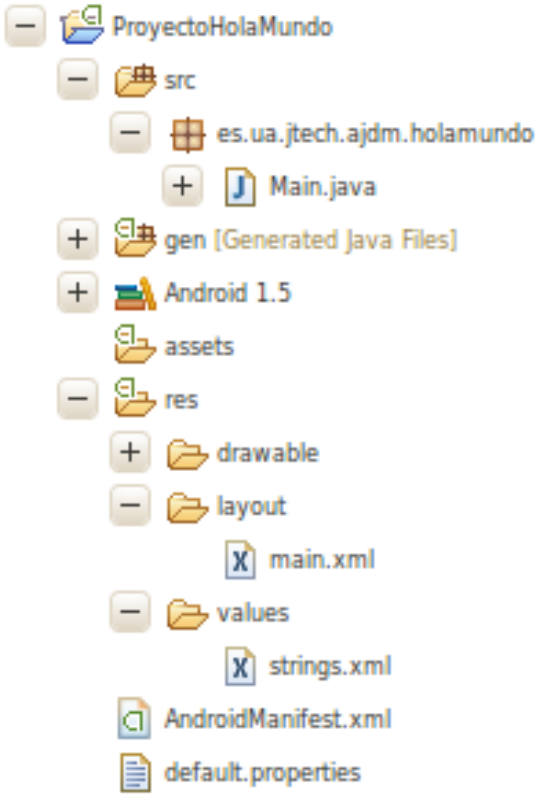


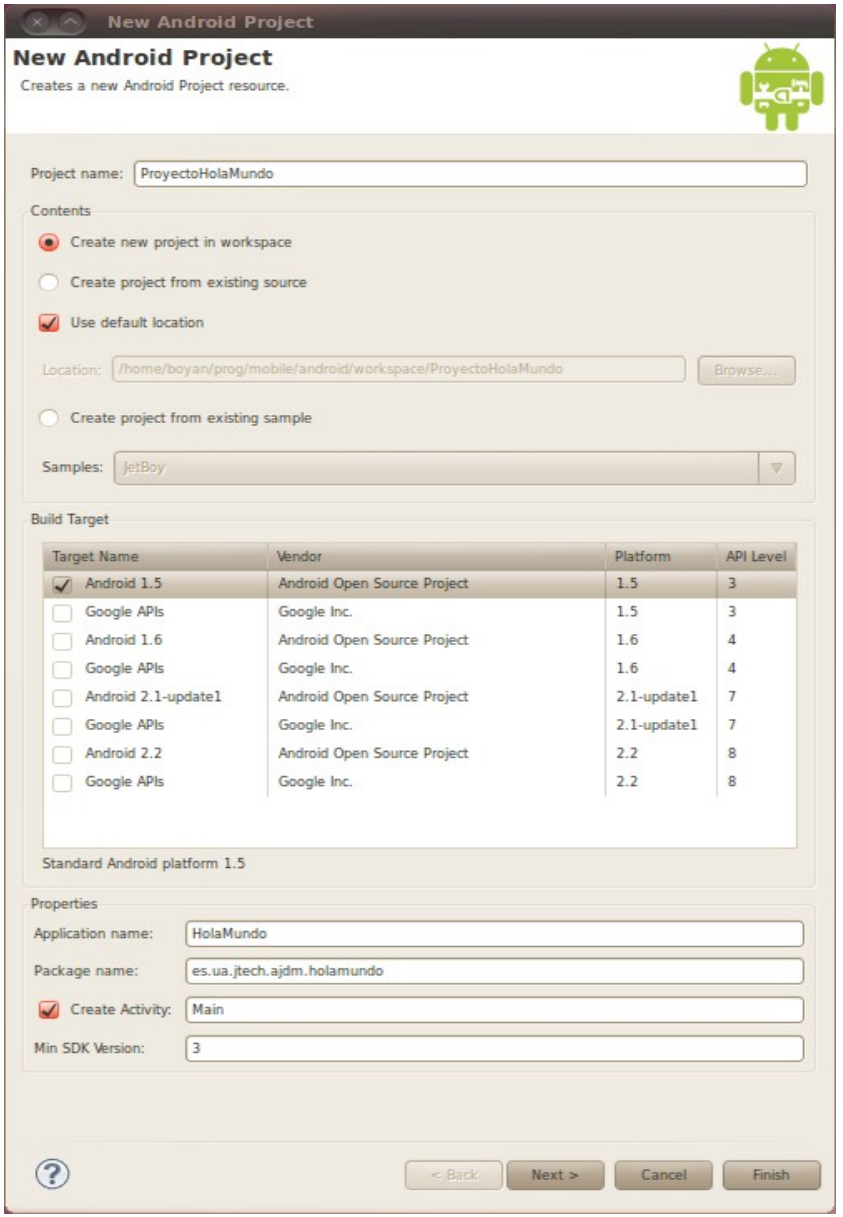
Editor visual de layouts



¡Hola, Mundo!

- Crear nuevo proyecto →
- Se genera la estrctura básica: →



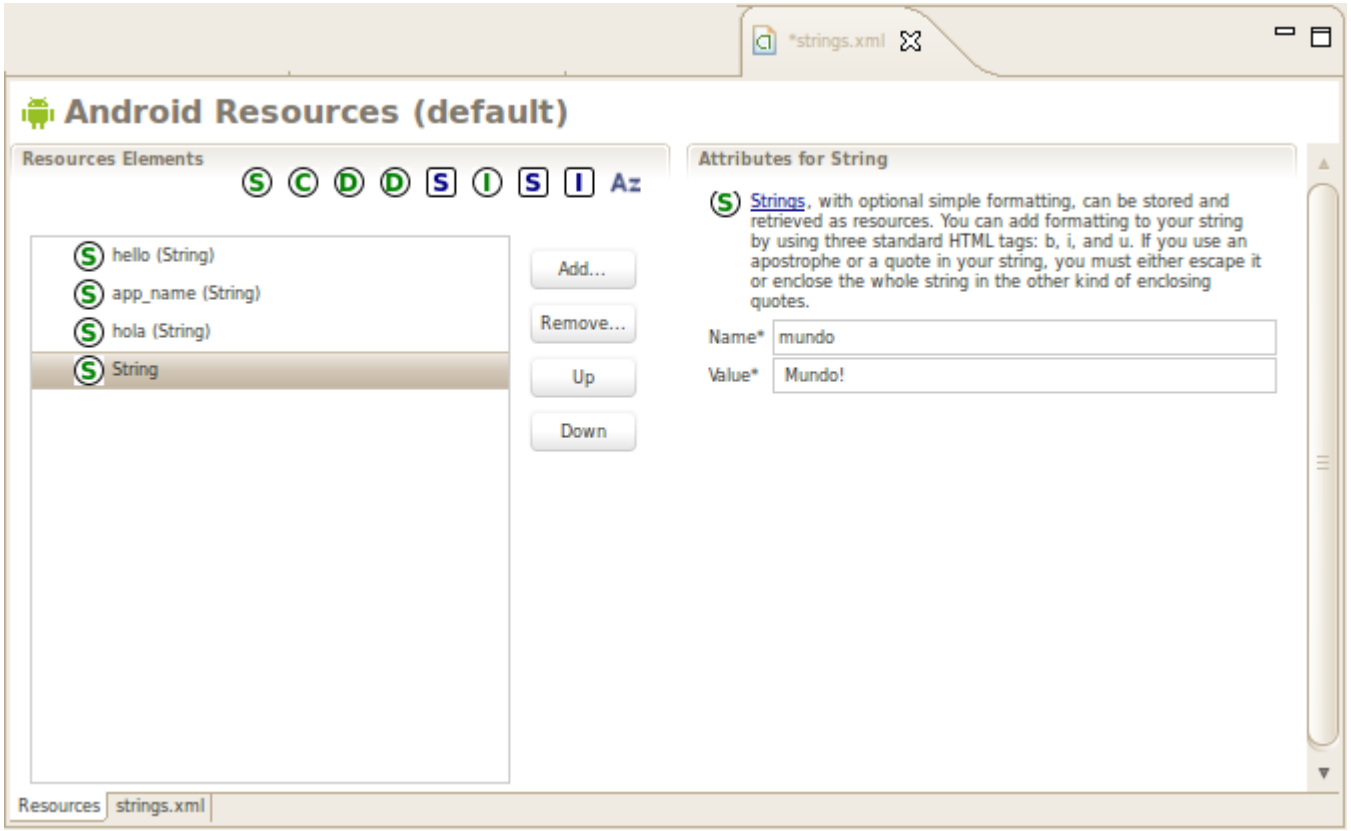


¡Hola, Mundo! AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="es.ua.jtech.ajdm.holamundo"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".Main"
                  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>
    <uses-sdk android:minSdkVersion="3" />

</manifest>
```

¡Hola, Mundo!

- Recursos strings en `res/values/strings.xml`



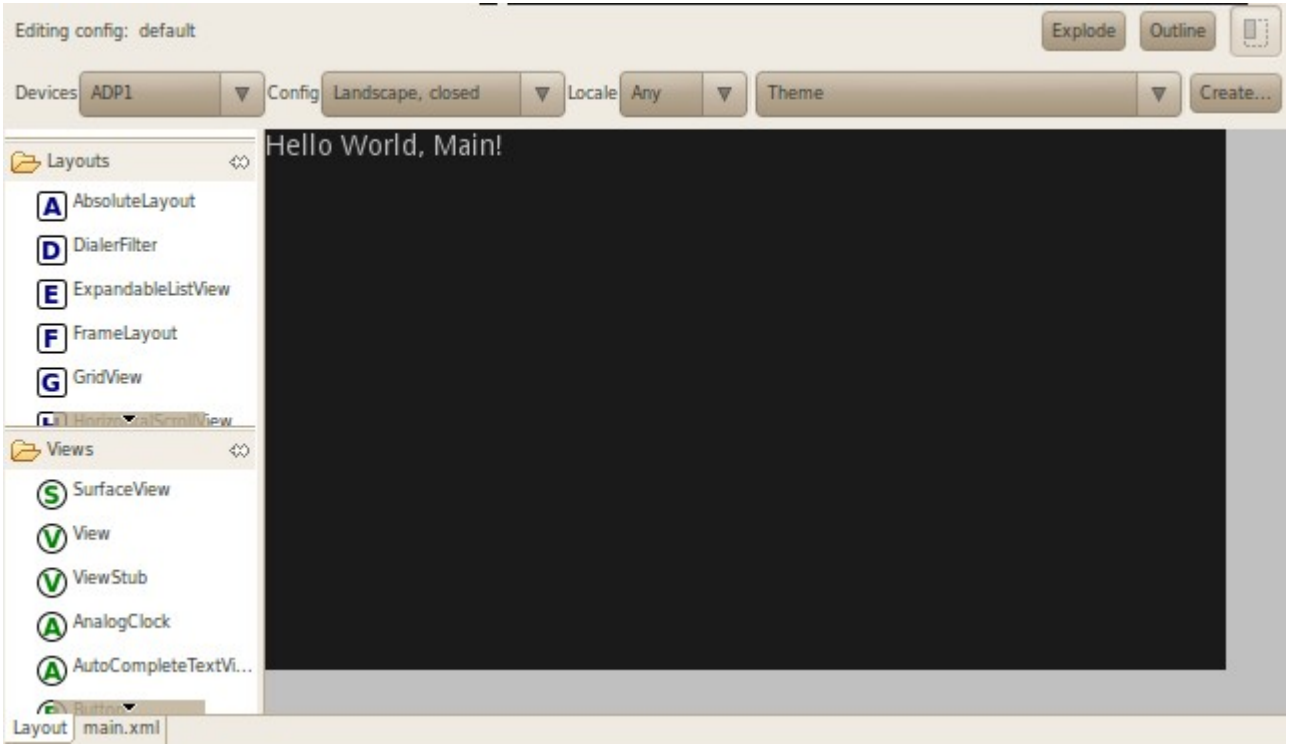
¡Hola, Mundo!

- Recursos strings en `res/values/strings.xml`
- Añadimos los string `hola`, `mundo` y `que`.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, Main!</string>
    <string name="app_name">HolaMundo</string>
        <string name="hola">Hola, </string>
        <string name="mundo"> Mundo!</string>
        <string name="que">Hola ¿qué?</string>
</resources>
```

¡Hola, Mundo!

- Layout en `res/layouts/main.xml`



¡Hola, Mundo!

- Layout en `res/layouts/main.xml`
- Eliminamos la etiqueta y ponemos una nueva etiqueta `TextView` y un botón `Button`:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView android:text="@+id/TextView01" android:id="@+id/TextView01"
          android:layout_width="wrap_content" android:layout_height="wrap_content" />
<Button android:text="@+id/Button01" android:id="@+id/Button01"
          android:layout_width="wrap_content" android:layout_height="wrap_content" />
</LinearLayout>
```

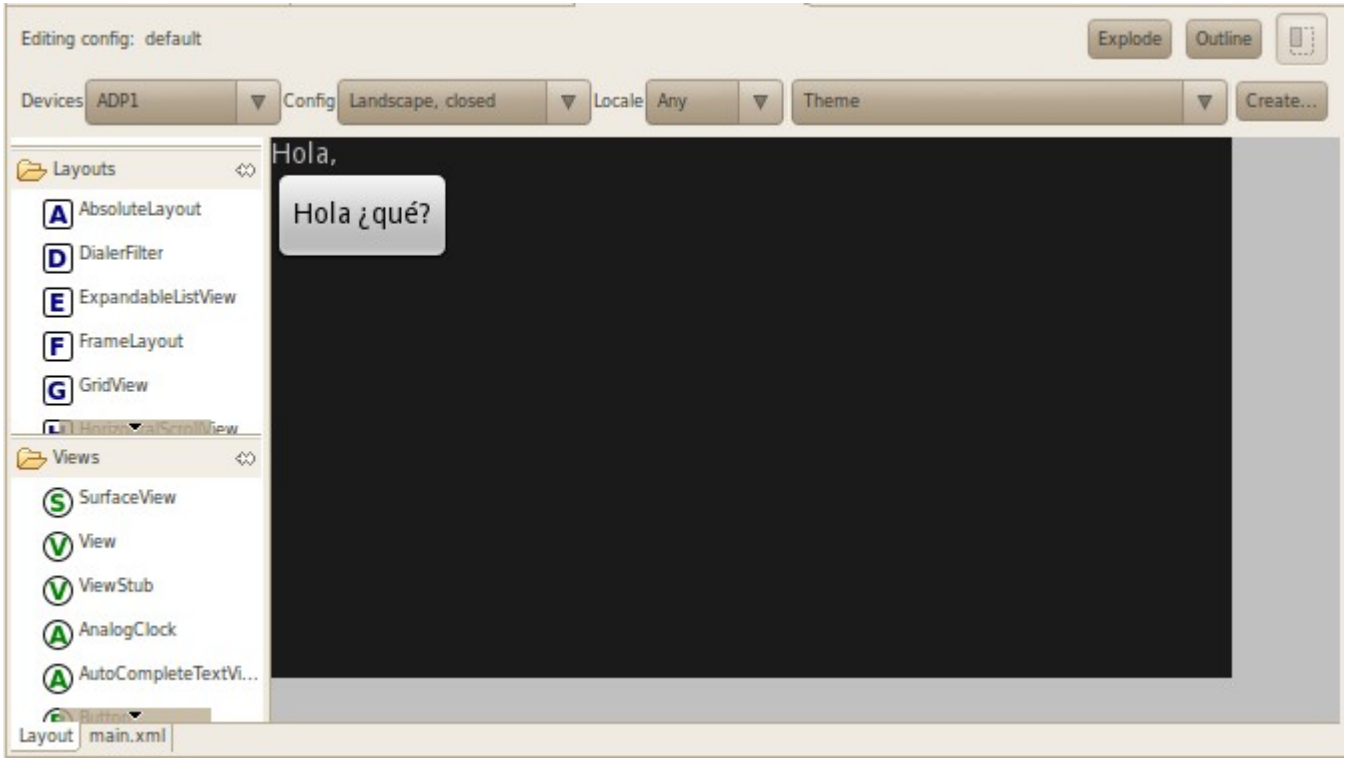
¡Hola, Mundo!

- Layout en `res/layouts/main.xml`
- Cambiamos los atributos `android:text` para que muestren los string de los recursos.

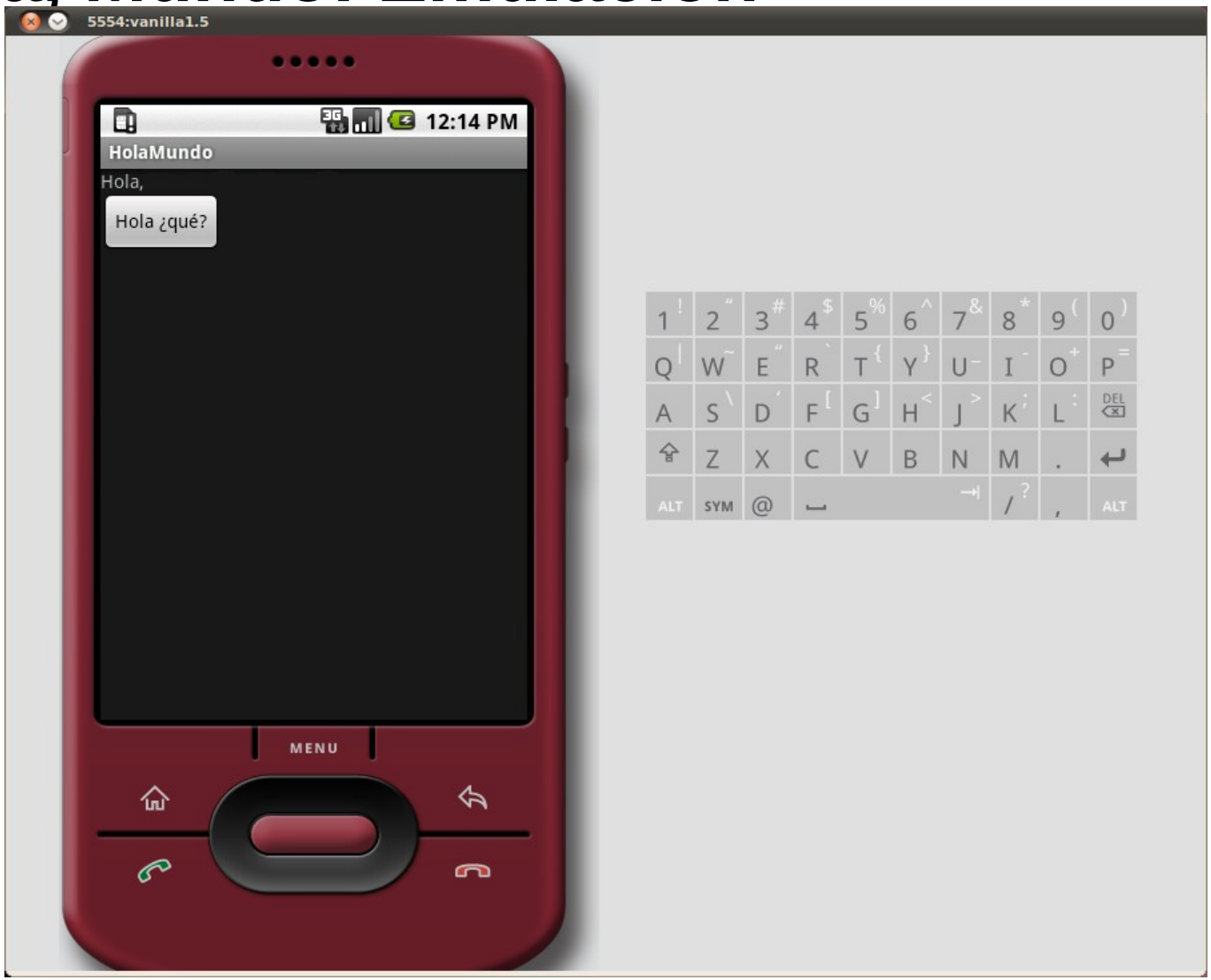
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView android:text="@string/hola" android:id="@+id/TextView01"
          android:layout_width="wrap_content" android:layout_height="wrap_content" />
<Button android:text="@string/que" android:id="@+id/Button01"
          android:layout_width="wrap_content" android:layout_height="wrap_content" />
</LinearLayout>
```

¡Hola, Mundo!

- Layout en `res/layouts/main.xml`



¡Hola, Mundo! Emulación

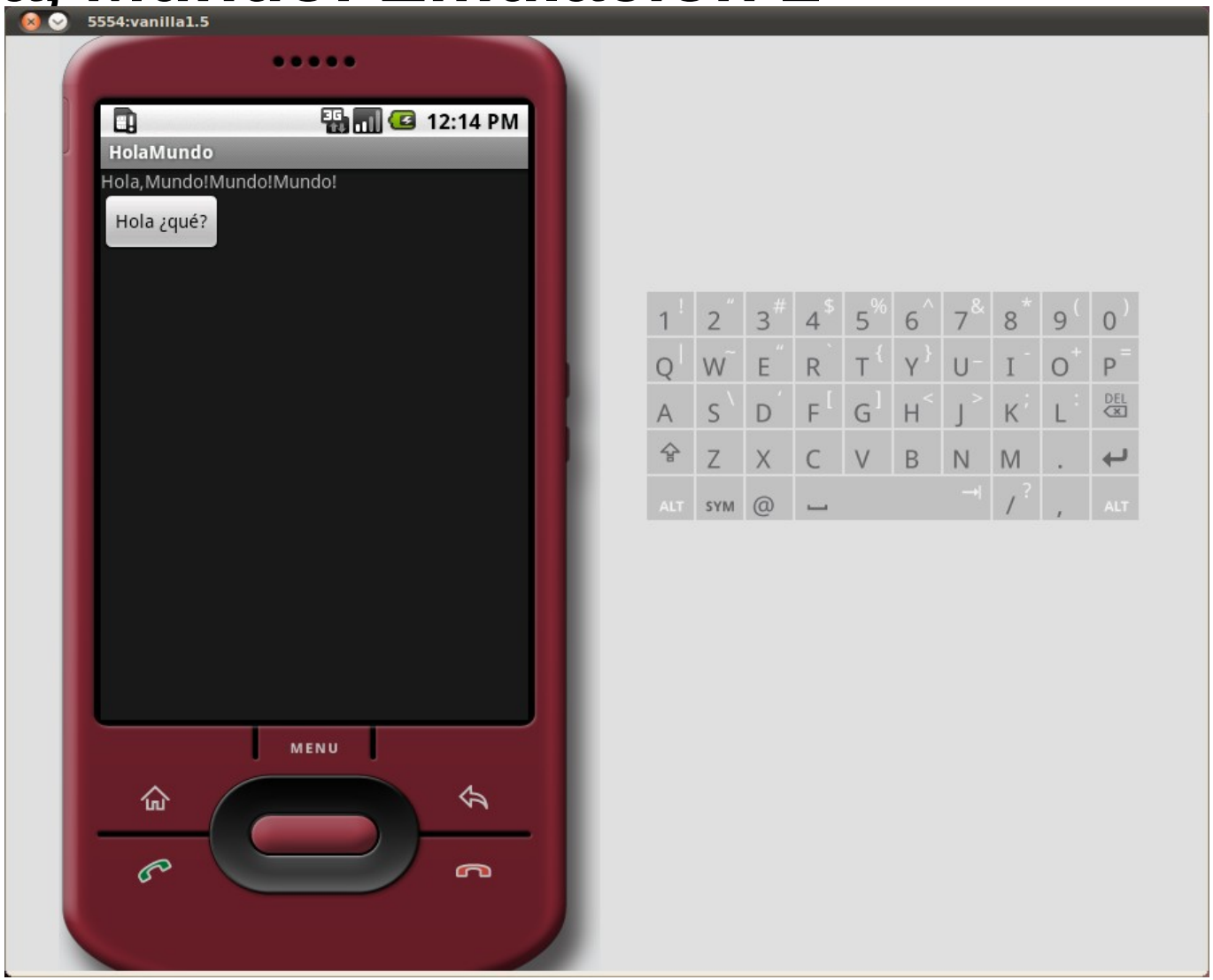


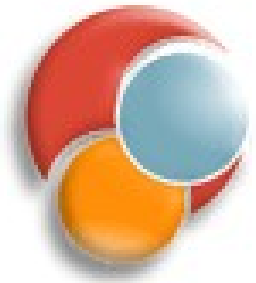
¡Hola, Mundo! Actividad y eventos

- Main.java

```
package es.ua.jtech.ajdm.holamundo;

import android.app.Activity;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

public class Main extends Activity {
    /** Called when the activity is first created. */
        TextView textView;
        Button   button;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        textView = (TextView)findViewById(R.id.TextView01);
        button = (Button)findViewById(R.id.Button01);

        button.setOnClickListener(new OnClickListener() {
                        @Override
                        public void onClick(View v) {
                                textView.append(getString(R.string.mundo));
                        }

                });

    }
}
```

¡Hola, Mundo! Emulación 2





¿Preguntas...?