

Android y Java para Dispositivos Móviles

Sesión 9: Introducción a Android



Puntos a tratar

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

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

- (imágenes de stealthcopter.com)



Nombres basados en repostería

- (imágenes de stealthcopter.com)



Nombres basados en repostería

- (imágenes de stealthcopter.com)



Nombres basados en repostería

- (imágenes de stealthcopter.com)





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

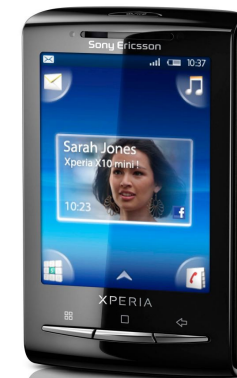
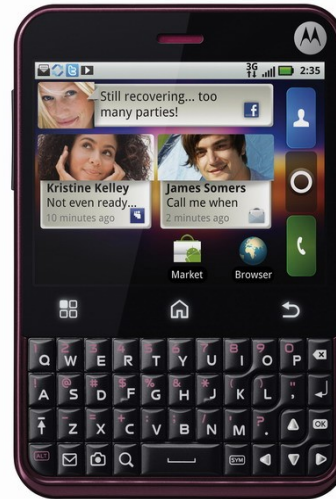


Cuestiones éticas

- 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.

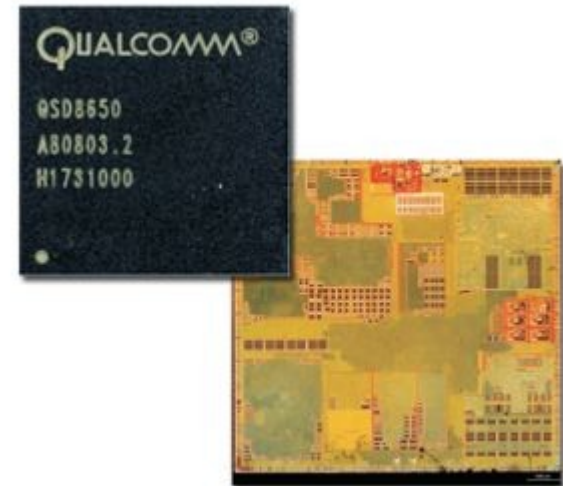


Dispositivos



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



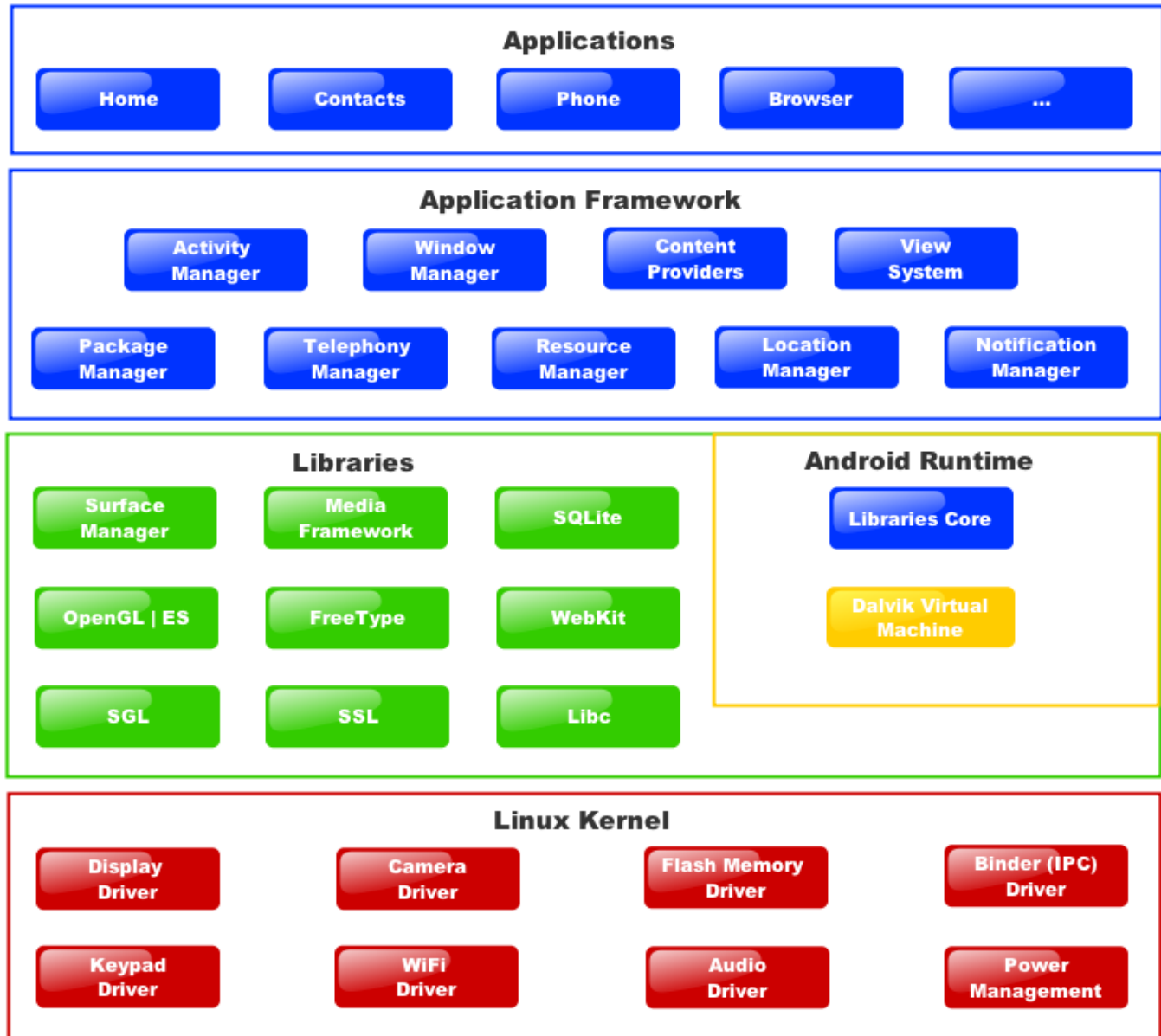


Android SDK

- 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





Capas

- Clases importantes para el desarrollo:
 - `ActivityManager`: Controla el ciclo de vida de las actividades.
 - `View`: Se usan para construir interfaces en las actividades.
 - `NotificationManager`: Mecanismo no intrusivo para mostrar avisos al usuario.
 - `ContentProvider`: Permiten intercambiar datos de una manera estandarizada.
 - *Resource Manager*: permite usar en la aplicación recursos que no forman parte del código, como XML, strings, recursos gráficos, audio, vídeo, etc.



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 localhost 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_mifuentes">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
```

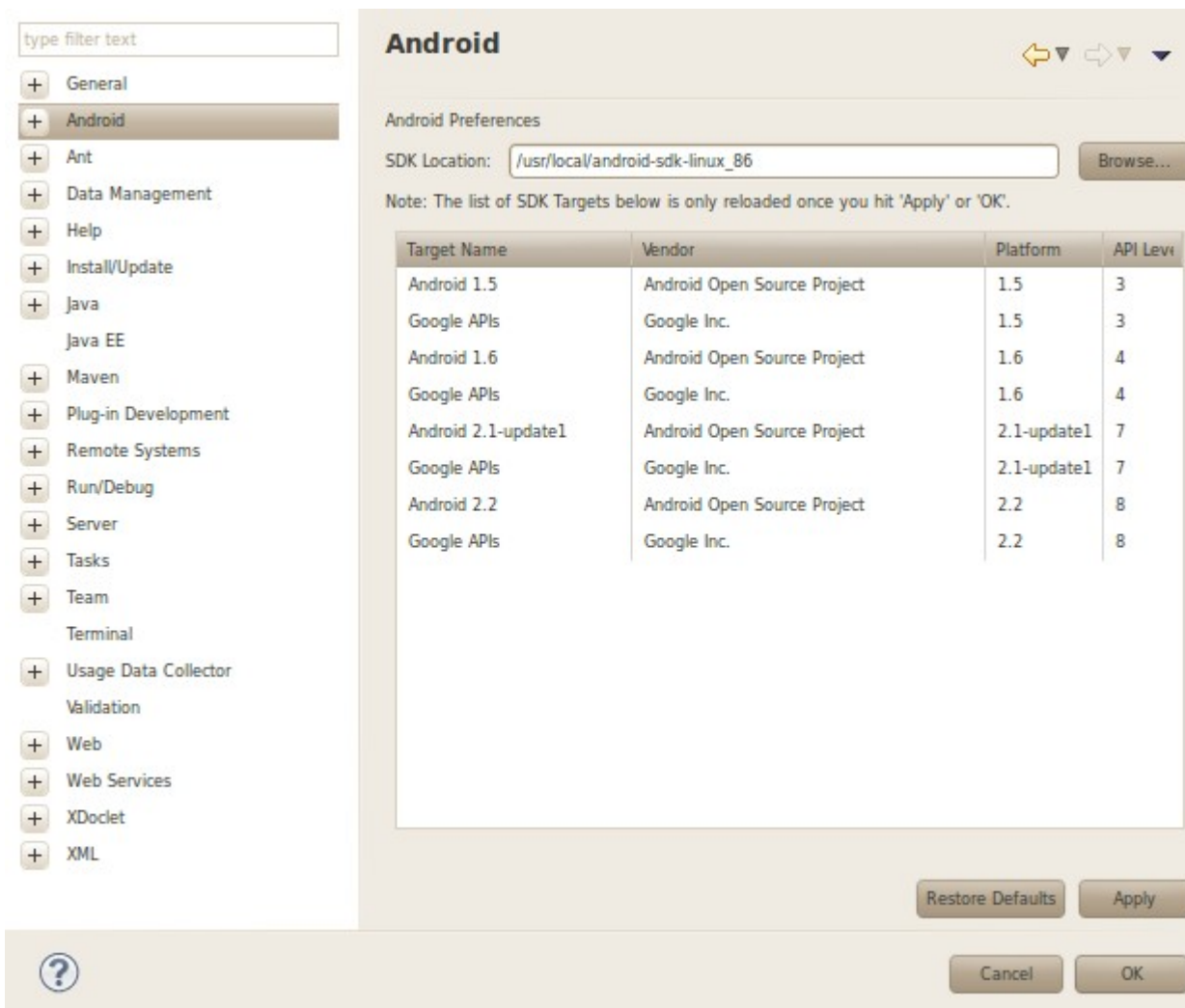


Plug-in para Eclipse

- 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.

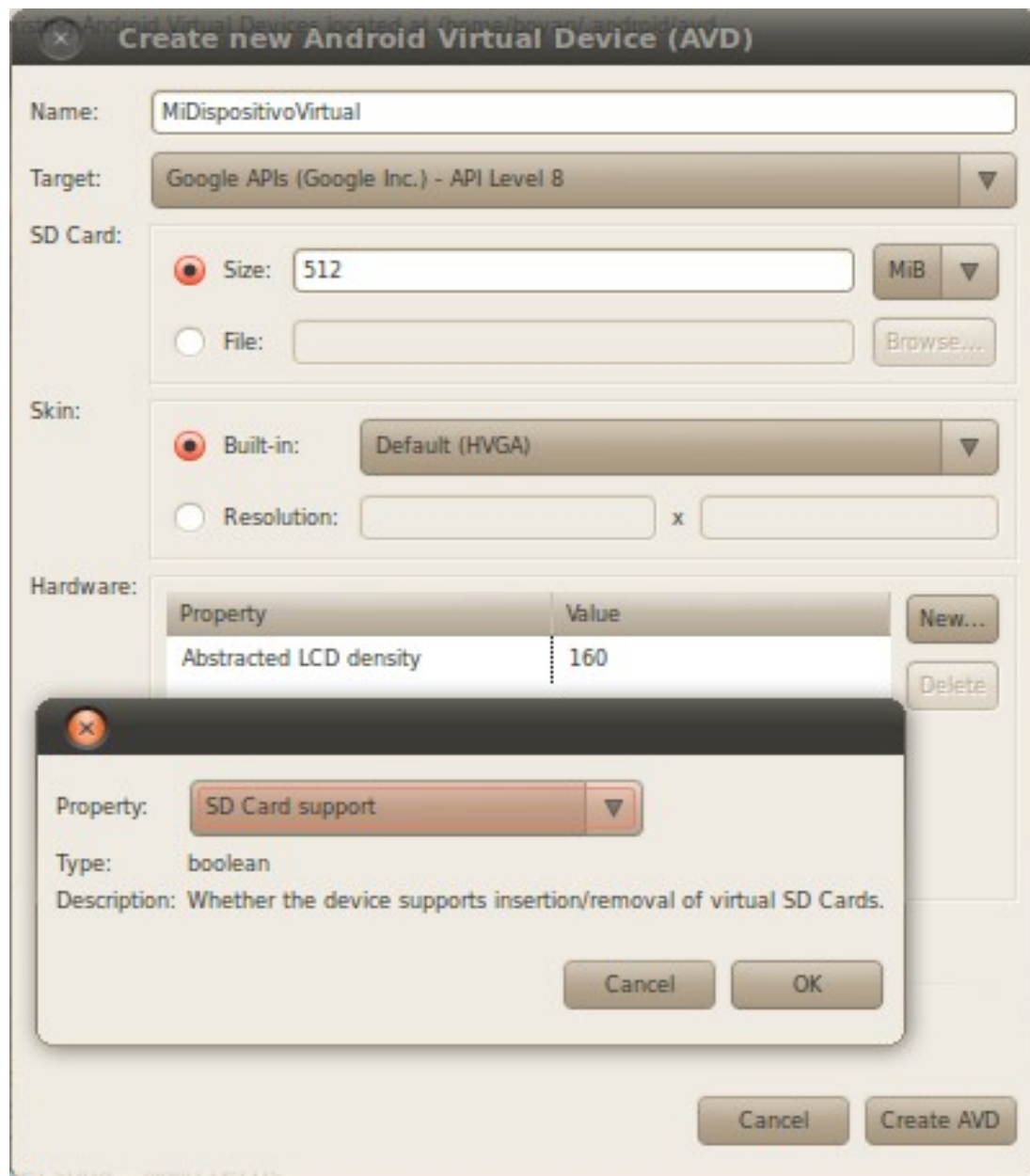


Plug-in para Eclipse



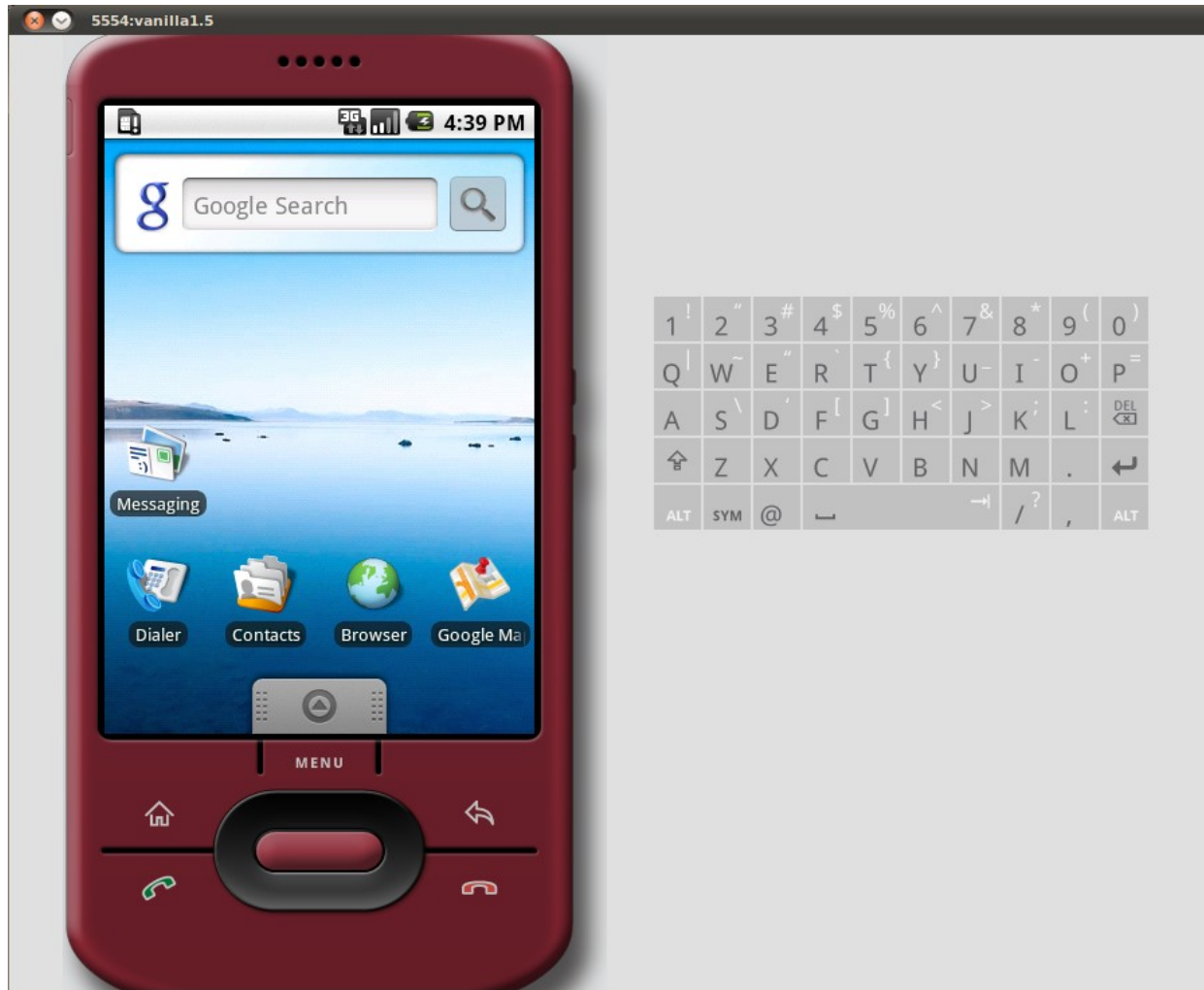
AVD Manager

- Crear nuevo dispositivo virtual (AVD):

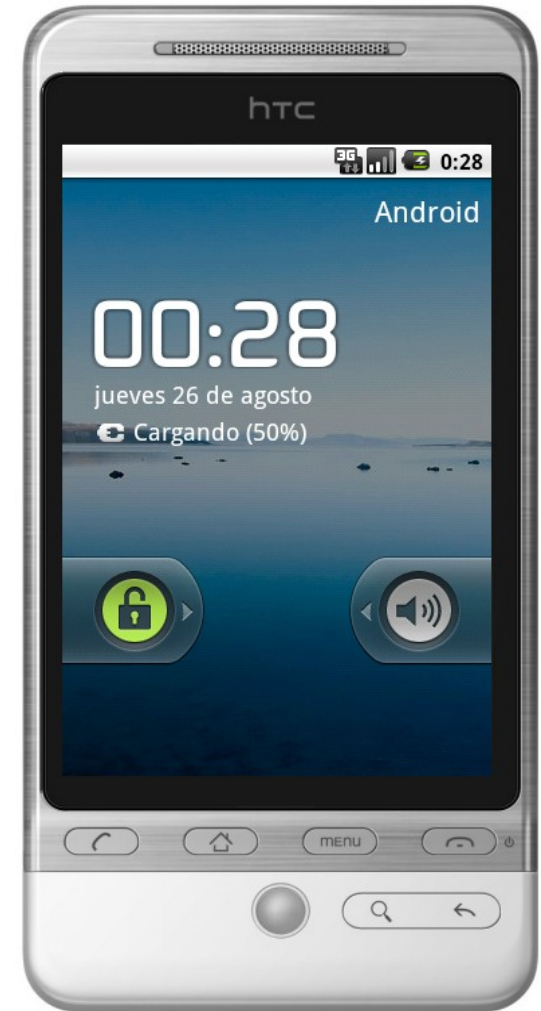




Emulador



Android y Java para Dispositivos Móviles



Introducción a Android-35



Asistente para la creación de proyectos

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

New Android Project
Creates a new Android Project resource.

Project name:

Contents

- ☒ Create new project in workspace
- ☐ Create project from existing source
- ☒ Use default location

Location:

- ☐ Create project from existing sample

Samples:

Build Target

Target Name	Vendor	Platform	API Level
<input checked="" type="checkbox"/> Android 1.5	Android Open Source Project	1.5	3
<input type="checkbox"/> Google APIs	Google Inc.	1.5	3
<input type="checkbox"/> Android 1.6	Android Open Source Project	1.6	4
<input type="checkbox"/> Google APIs	Google Inc.	1.6	4
<input type="checkbox"/> Android 2.1-update1	Android Open Source Project	2.1-update1	7
<input type="checkbox"/> Google APIs	Google Inc.	2.1-update1	7
<input type="checkbox"/> Android 2.2	Android Open Source Project	2.2	8
<input type="checkbox"/> Google APIs	Google Inc.	2.2	8

Standard Android platform 1.5

Properties

Application name:

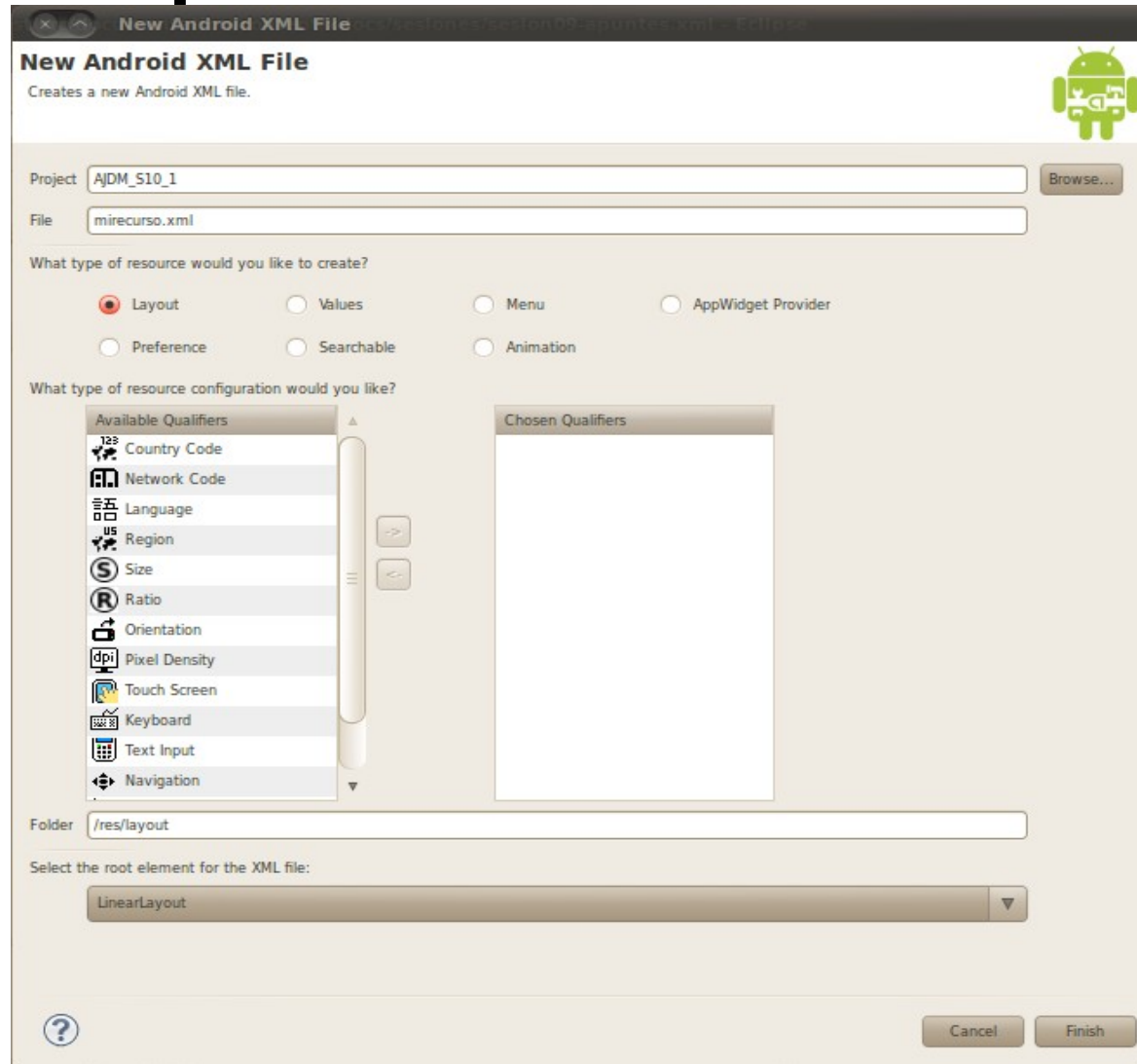
Package name:

☒ Create Activity:

Min SDK Version:



Asistente para crear recursos XML





Vista Dalvik Debug Monitor Service

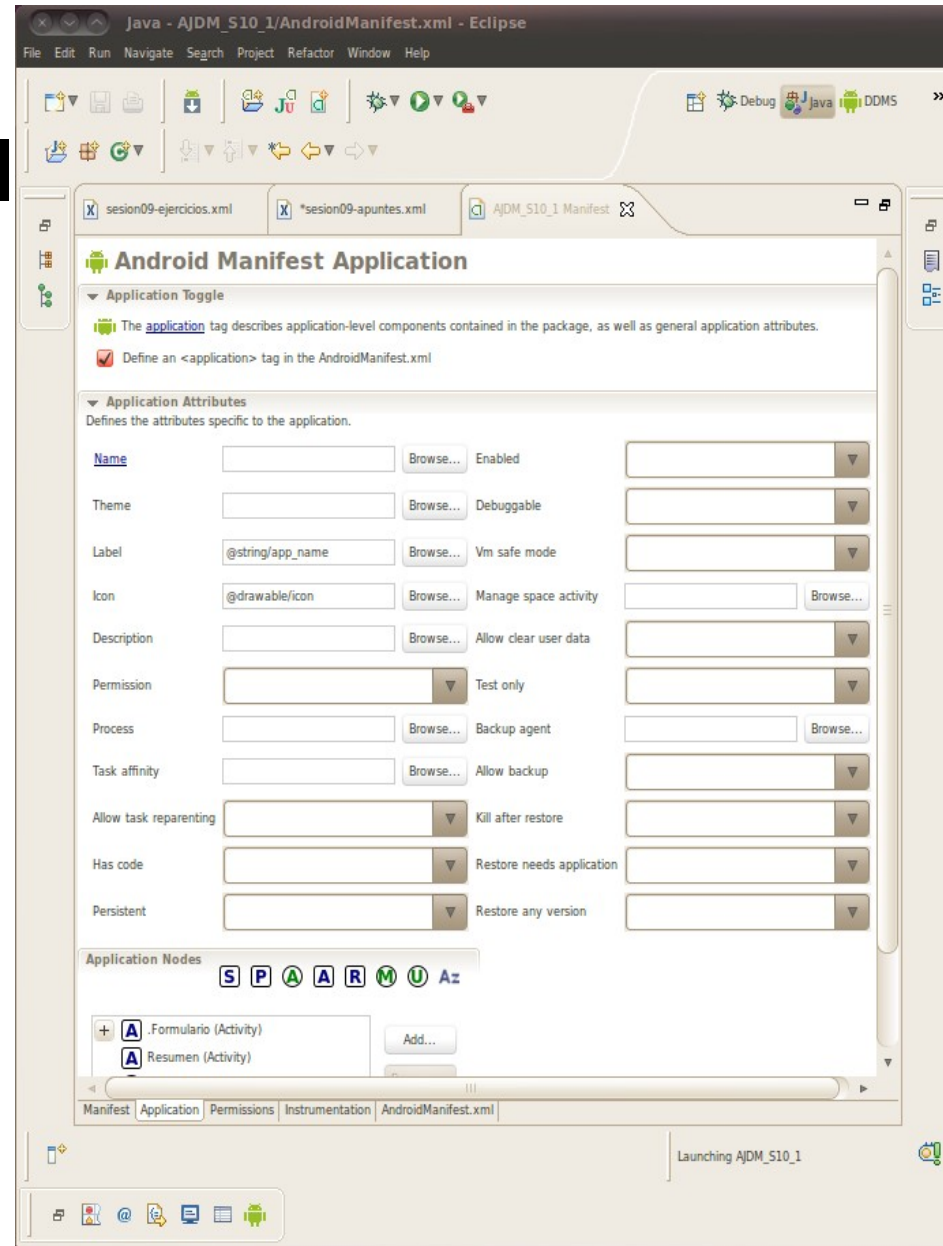
The screenshot displays the DDMS interface within the Eclipse IDE. The top toolbar includes icons for file operations, running, and debugging. The main workspace is divided into several panes:

- Devices:** Shows a list of virtual devices. The selected device is 'emulator-5554', which is online and running on a Hero22 [2.2, d].
- Emulator Control:** Provides controls for the emulator's telephony status, including voice, data, speed, and latency settings.
- Source Editor:** Displays the Java source code for 'Formulario.java'. The code includes package declarations, imports, and the implementation of the 'onCreate' method.
- LogCat:** Shows the system log for the emulator. The log entries include:

Time	pid	tag	Message
08-26 00:41:13.107	I 66	ActivityManager	start proc es.ua.jtech.ajdm.s10 for activity es.ua.jtech.ajdm.s10/Formulario; pid=290 uid=10030 gids={30027}
08-26 00:41:16.397	I 66	ActivityManager	Displayed activity es.ua.jtech.ajdm.s10/Formulario: 3268 ms (total 3268 ms)
08-26 00:41:53.177	D 66	ThrottleService	finally have imsi - retrieving data
08-26 00:41:53.227	D 66	ThrottleService	onPollAlarm - roaming=false, read=0, written=0, new total=0
08-26 00:42:05.287	D 117	dalvikvm	GC_EXPLICIT freed 2972 objects / 159432 bytes in 125ms

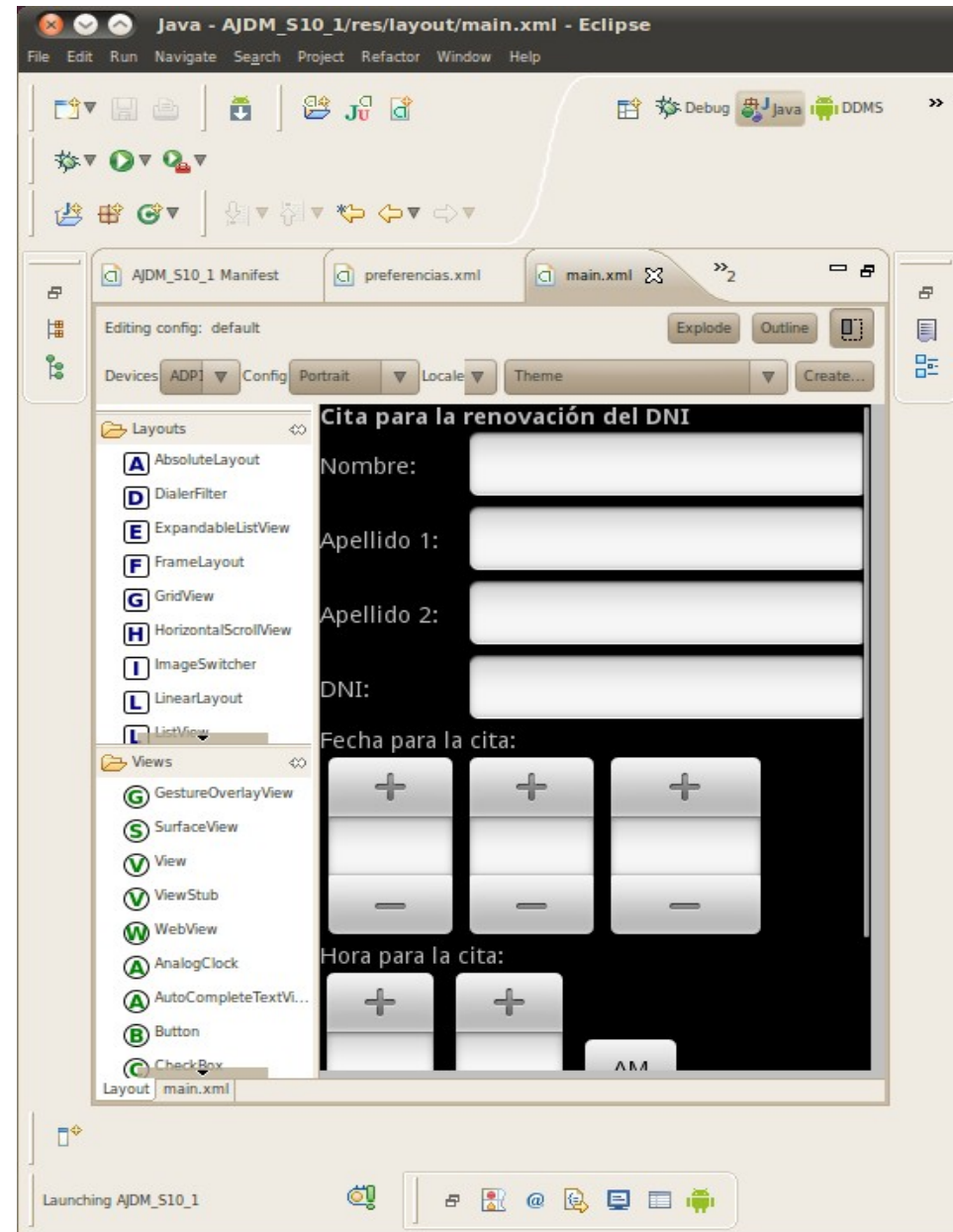


Editor del AndroidManifest.xml





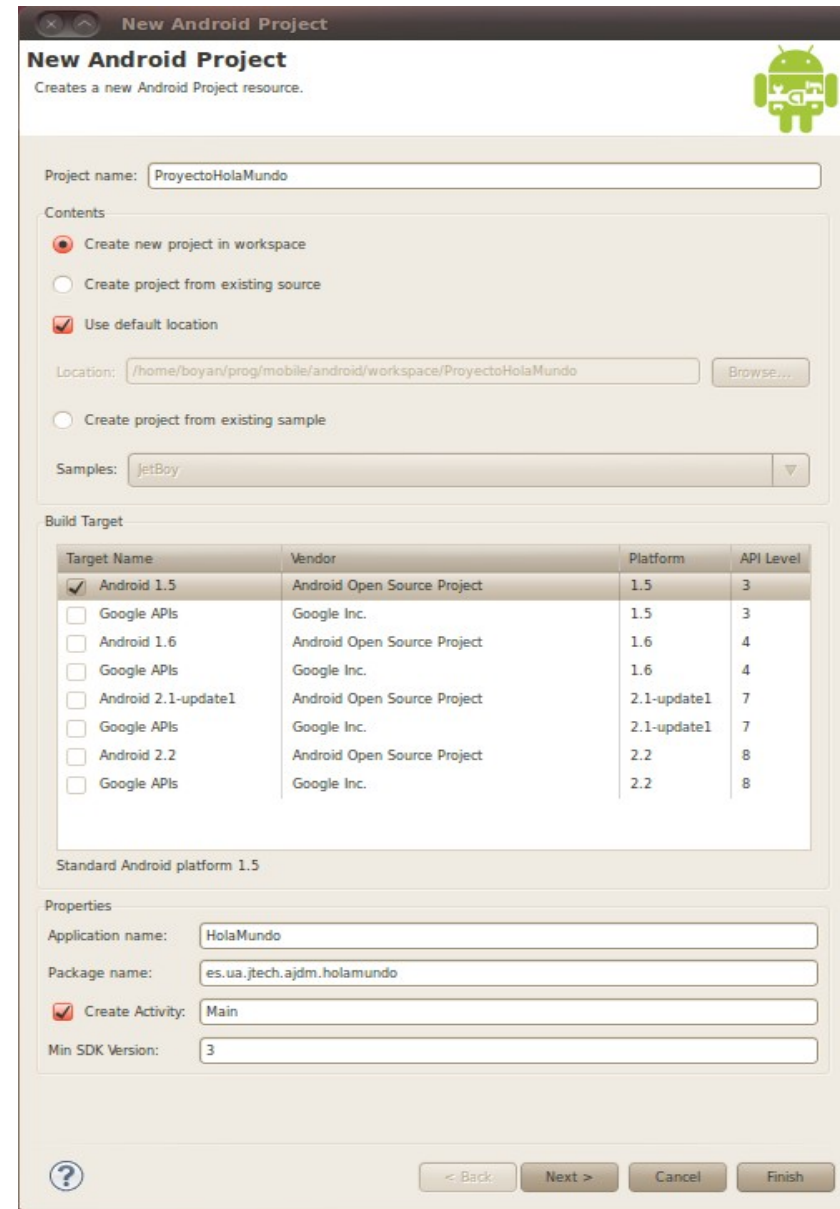
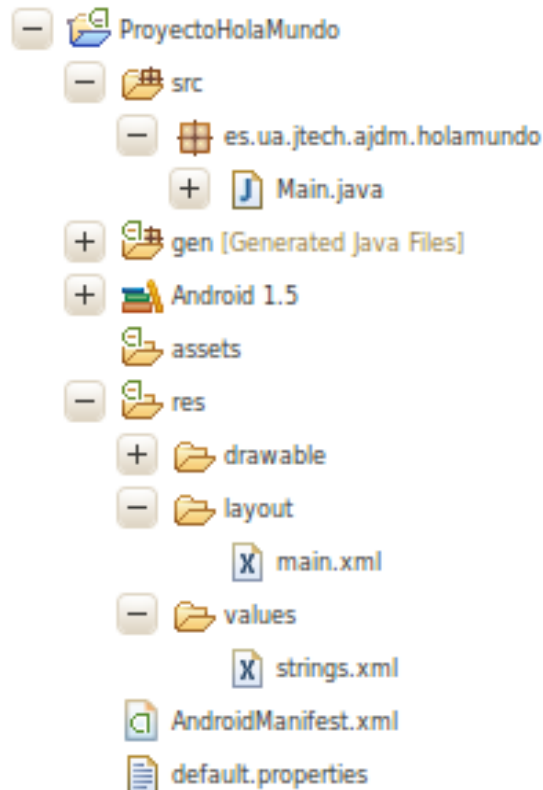
Editor visual de layouts





¡Hola, Mundo!

- Crear nuevo proyecto →
- Se genera la estructura 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>

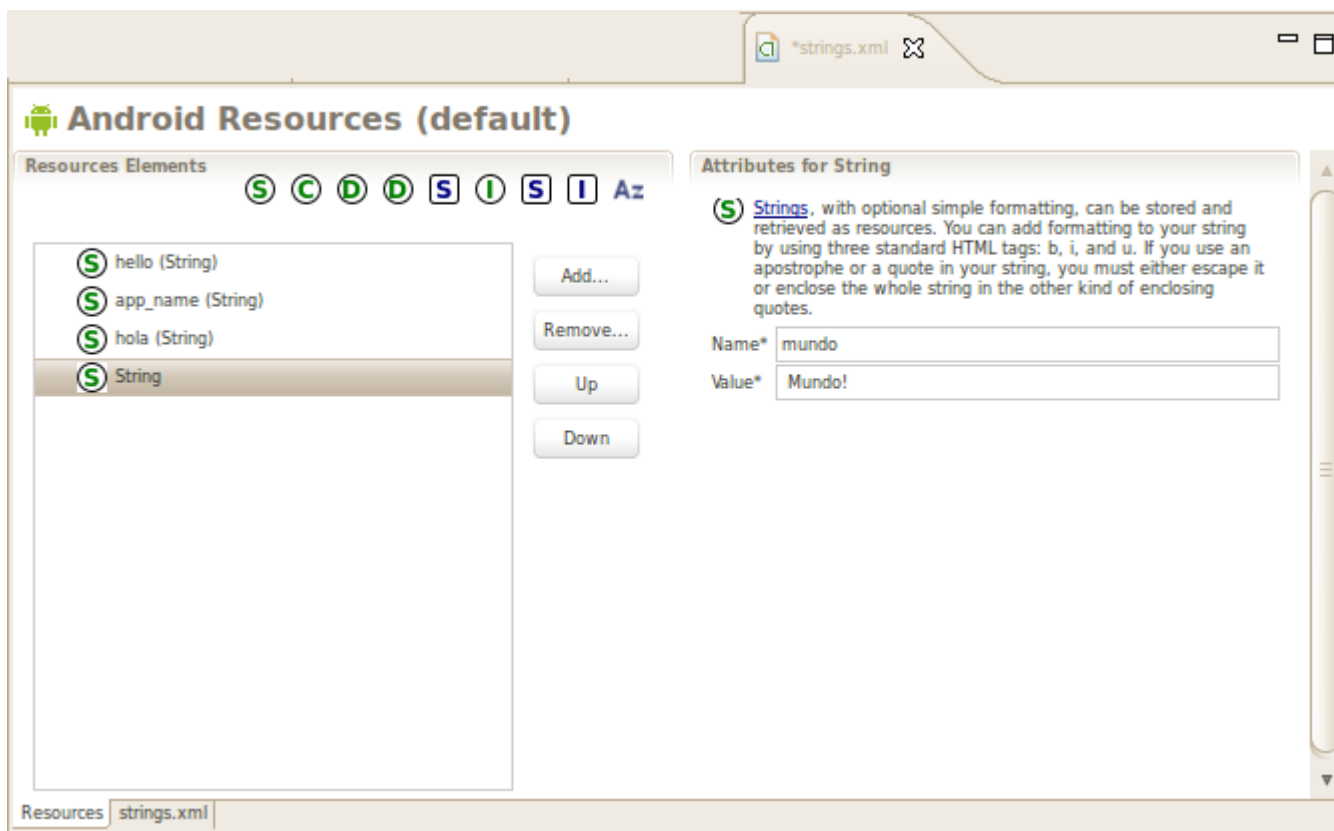
    </application>
    <uses-sdk android:minSdkVersion="3" />

</manifest>
```



¡Hola, Mundo!

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





¡Hola, Mundo!

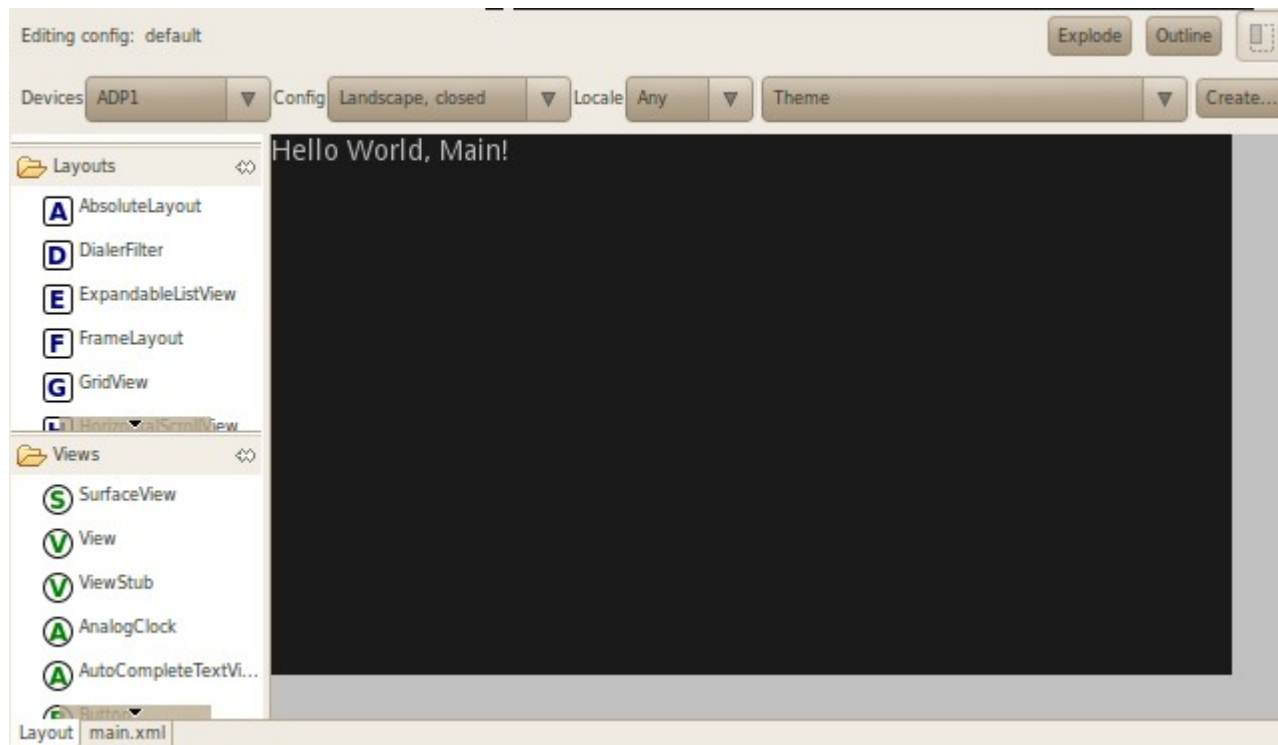
- Recursos strings en `res/values/strings.xml`
- Añadimos los string `hola`, `mundo` y `que`.

```
<?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!

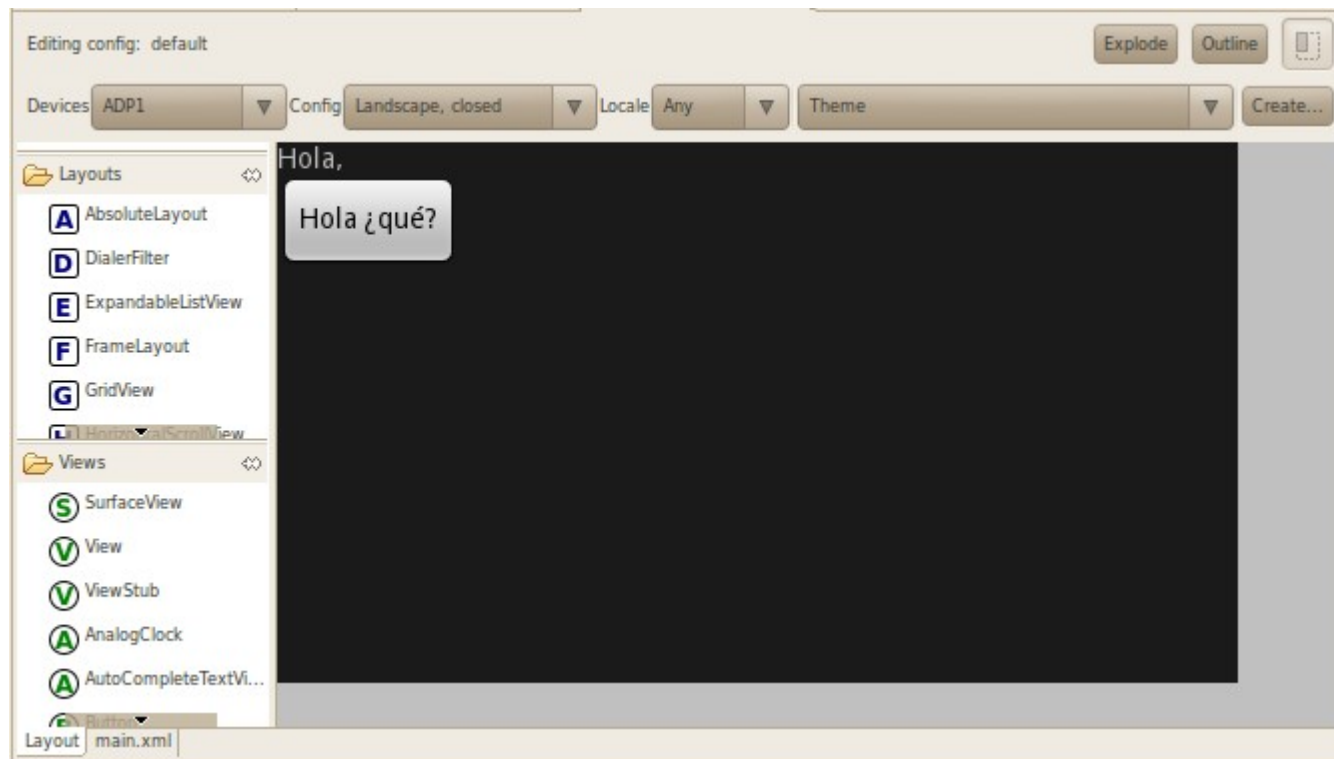
- Layout en `res/layouts/main.xml`
- Cambiamos los atributos `android:text` para que muestren los string de los recursos.

```
<?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>
```



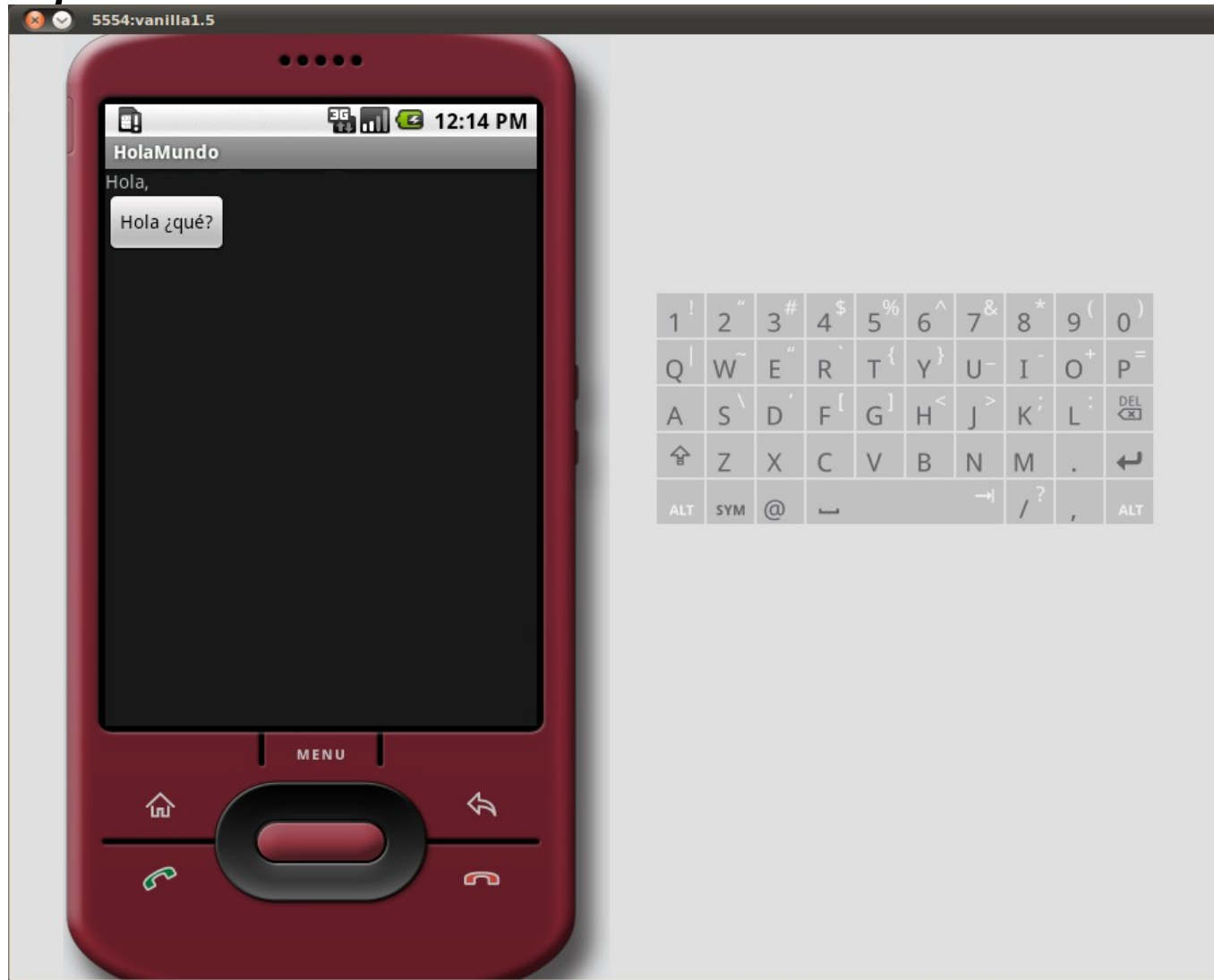
¡Hola, Mundo!

- 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;

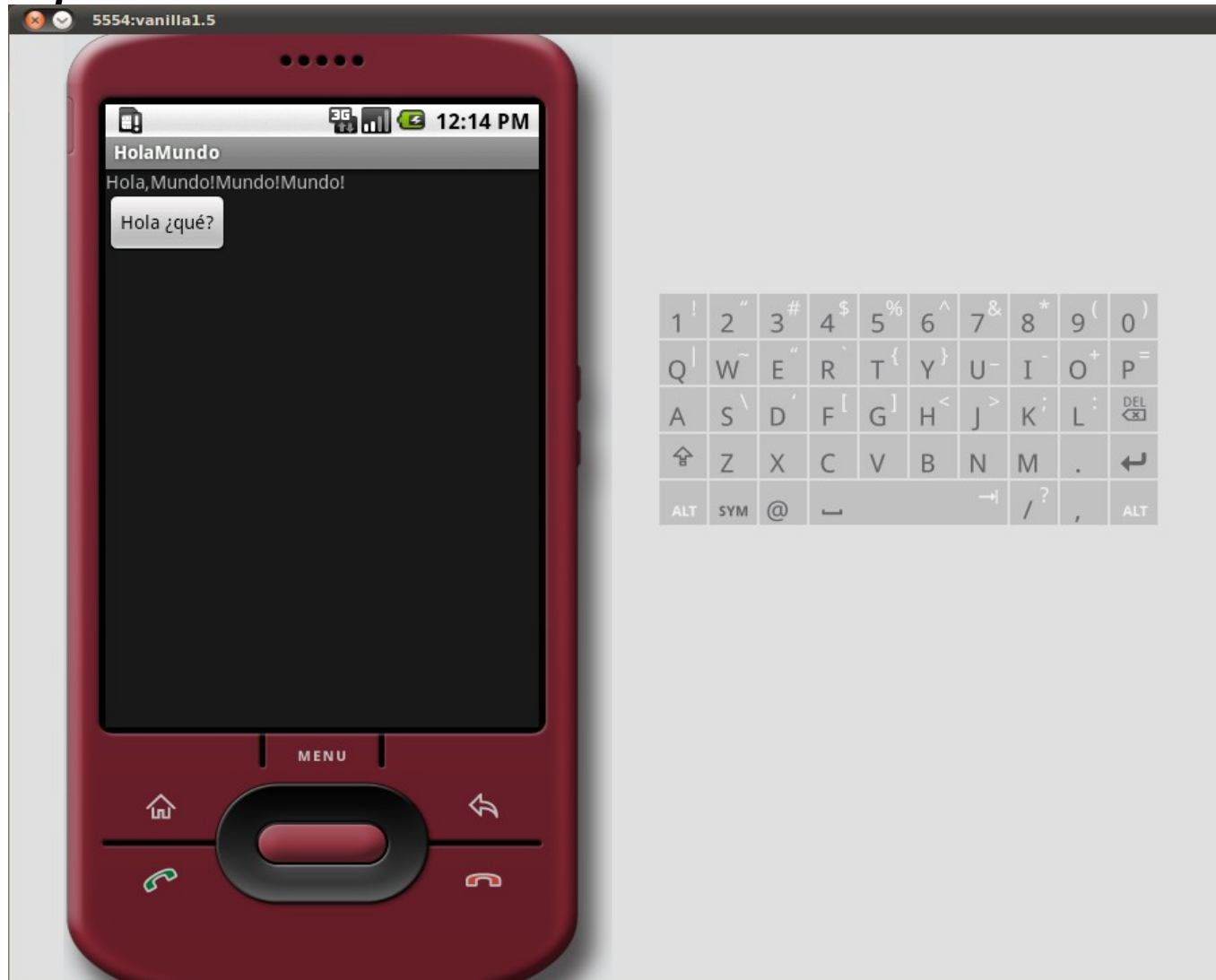
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

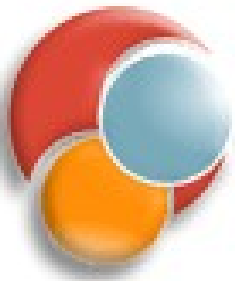
        textView = (TextView)findViewById(R.id.TextView01);
        button = (Button)findViewById(R.id.Button01);

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



¡Hola, Mundo! Emulación 2





¿Preguntas...?