

# **El gran libro de Android Avanzado**



# **El gran libro de Android Avanzado**

Jesús Tomás

Vicente Carbonell

Carsten Vogt

Miguel García Pineda

Jordi Bataller Mascarell



*El gran libro de Android Avanzado*

3ª edición, 2017

1ª reimpresión, setiembre 2016

© 2017, Jesús Tomás Gironés, Vicente Carbonell, Carsten Vogt, Miguel García Pineda,  
Jordi Bataller Mascarell

© MARCOMBO, S.A., 2017

[www.marcombo.com](http://www.marcombo.com)

«Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra solo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Dirijase a CEDRO (Centro Español de Derechos Reprográficos, [www.cedro.org](http://www.cedro.org)) si necesita fotocopiar o escanear algún fragmento de esta obra».

ISBN: 978-84-267-2257-7

DL: B-16545-2015

Impreso en XXXXXXXX

*Printed in Spain*

A mi madre por creer en mí.

JESÚS TOMÁS

Para Maite, Inés y Carles con todo el cariño por su paciencia y comprensión.

VICENTE CARBONELL

A toda la gente que me ha apoyado en esta última etapa,  
en especial a Diana por su cariño y ayuda constante.

MIGUEL GARCÍA PINEDA

Para Mar, Ferran y Cristina.

JORDI BATALLER MASCARELL



# Índice general

Lista de acrónimos .....	xv
¿Cómo leer este libro? .....	<b>¡Error! Marcador no definido.</b>
<b>CAPÍTULO 1. Diseño avanzado de interfaces de usuario .....</b>	<b>17</b>
1.1. Acceder a objetos globales de la aplicación .....	17
1.1.1. La clase Application .....	17
1.1.2. Uso del patrón Singleton .....	19
1.2. Material Design .....	20
1.2.1. Definición de la paleta de colores de la aplicación .....	21
1.3. RecyclerView .....	21
1.4. Fragments .....	24
1.4.1. Insertar <i>fragments</i> desde XML .....	24
1.4.2. Insertar <i>fragments</i> desde código .....	29
1.4.3. Comunicación e intercambio de <i>fragments</i> .....	29
1.5. Otros elementos del interfaz de usuario .....	30
1.5.1. Menús contextuales .....	30
1.5.2. La barra de acciones ( <i>ActionBar</i> / <i>Toolbar</i> ) .....	<b>¡Error! Marcador no definido.</b>
1.5.2.1. Barra de acciones con <i>ActionBar</i> .....	31
1.5.2.2. Barra de acciones con <i>Toolbar</i> .....	32
1.5.2.3. Configurar la barra de acciones desde un <i>fragment</i> .....	32
1.5.3. Vistas animadas: <i>CoordinatorLayout</i> , <i>AppBarLayout</i> , <i>FloatingActionButton</i> y <i>SnackBar</i> .....	33
1.5.4. Pestañas con <i>TabLayout</i> .....	36
1.6. <i>Navigation Drawer</i> .....	39
1.6.1. Ocultar elementos de la interfaz de usuario .....	42
1.7. Servicios de búsquedas .....	43
1.7.1. Añadiendo preferencias de usuario mediante <i>PreferenceFragmentCompat</i> .....	45
1.8. Animaciones .....	48
1.8.1. Animaciones de vistas: transiciones entre actividades .....	48
1.8.1.1. Aplicando animaciones de vistas en Audiolibros .....	49

1.8.2. Animaciones de propiedades .....	50
1.8.2.1. El motor básico de animación: <i>ValueAnimator</i> .....	50
1.8.2.2. Automatizando las animaciones: <i>ObjectAnimator</i> .....	50
1.8.2.3. Combinando animaciones: <i>AnimatorSet</i> <b>Error! Marcador no definido.</b>	
1.8.2.4. Definiendo animaciones en XML .....	50
1.8.2.5. Nuevas propiedades de la clase <i>View</i> .....	52
1.8.2.6. Aplicando animaciones de propiedades en Audiolibros .....	52
1.8.3. Animaciones en <i>RecyclerView</i> .....	53
1.9. Otros aspectos introducidos en la versión 5.0 .....	54
1.9.1. Extraer paleta de colores desde imágenes .....	54
1.9.2. Uso de gráficos vectoriales .....	54
<b>CAPÍTULO 2. Diseño personalizado de vistas</b> .....	57
2.1. Algunos conceptos básicos.....	57
2.2. Una vista como la composición de varias vistas.....	57
2.2.1. Creación de escuchadores de eventos .....	59
2.3. Modificación de vistas existentes.....	60
2.3.1. Algo más de información sobre <i>TextView</i> .....	61
2.4. Creación de nuevos atributos XML .....	62
2.5. Una vista creada desde cero .....	64
2.5.1. Diseño y dibujo de la vista.....	64
2.5.2. Gestión de eventos.....	67
2.5.3. Cómo Android dibuja las vistas y obtiene sus tamaños .....	69
2.5.4. Interactuando con otros objetos .....	69
2.6. Creación de <i>widgets</i> de escritorio.....	70
2.6.1. Pasos a seguir para crear un <i>widget</i> .....	70
2.6.1.1. Definir las características del <i>widget</i> .....	70
2.6.1.2. Diseñar el <i>layout</i> del <i>widget</i> ..... <b>Error! Marcador no definido.</b>	
2.6.1.3. Crear una clase descendiente de <i>AppWidgetProvider</i> <b>Error! Marcador no de</b>	
2.6.1.4. Declarar el <i>widget</i> en <i>AndroidManifest</i> .....	70
2.6.1.5. Crear una actividad para configurarlo .....	70
2.6.2. Creación de un <i>widget</i> de escritorio sencillo .....	70
2.6.3. Actualizando el <i>widget</i> de escritorio .....	71



---

2.6.4. Actuando ante el evento onClick .....	72
2.6.5. Añadiendo una actividad de configuración .....	73
<b>CAPÍTULO 3. Hilos de ejecución en la interfaz del usuario .....</b>	<b>75</b>
3.1. Programación basada en eventos y el hilo de ejecución de usuario .....	75
3.1.1. Cola de eventos y bucle de eventos .....	75
3.1.2. El hilo de la interfaz de usuario de una aplicación Android .....	76
3.2. Concurrencia en programación orientada a eventos .....	77
3.2.1. Hilos para el manejo de eventos .....	77
3.2.2. El problema de los hilos en segundo plano: no tienen acceso a la interfaz gráfica de usuario .....	77
3.3. La clase AsyncTask .....	78
3.3.1. Extendiendo AsyncTask .....	78
3.3.2. Secuencia de operaciones .....	78
3.4. Animaciones con SurfaceView .....	82
3.4.1. Llamando al método onDraw() .....	<b>¡Error! Marcador no definido.</b>
3.4.2. Programación con SurfaceViews .....	<b>¡Error! Marcador no definido.</b>
3.4.3. Estructura de la aplicación .....	82
<b>CAPÍTULO 4. Comunicación con bluetooth .....</b>	<b>87</b>
4.1. Diferencias entre bluetooth e Internet .....	<b>¡Error! Marcador no definido.</b>
4.2. Pasos en la comunicación bluetooth .....	<b>¡Error! Marcador no definido.</b>
4.2.1. Emparejar dispositivos .....	<b>¡Error! Marcador no definido.</b>
4.2.2. Transmisión de datos .....	<b>¡Error! Marcador no definido.</b>
4.3. Las clases fundamentales en bluetooth .....	<b>¡Error! Marcador no definido.</b>
4.4. Permisos en bluetooth .....	87
4.5. Configuración bluetooth .....	87
4.6. Búsqueda de dispositivos bluetooth .....	91
4.6.1. Consultar dispositivos vinculados .....	91
4.6.2. Descubriendo dispositivos .....	91
4.6.3. Habilitar Descubrimiento .....	91
4.7. Conexión de dispositivos .....	93
4.7.1. Conexión como servidor .....	93
4.7.2. Conexión como cliente .....	94

4.7.3. Gestión de la conexión .....	95
4.8. Intercambio de información con Android bluetooth.....	97
4.8.1. ServidorBT .....	97
4.8.1.1. Actividad principal. ServidorBT .....	98
4.8.1.2. Hilo Servidor. ServServidorBT .....	104
4.8.1.3. Obtención de información desde ServServidorBT .....	109
4.8.2. ClienteBT .....	110
4.8.2.1. Actividad principal. ClienteBT .....	110
4.8.2.2. ArrayAdapterBT .....	116
4.8.2.3. Hilo Cliente. ServClienteBT .....	117
4.8.2.4. Finalización ClienteBT .....	122
4.9. La comunicación con los programas en Java SE .....	127
<b>CAPÍTULO 5. Servicios en la nube .....</b>	<b>131</b>
5.1. Introducción a los servicios en la nube .....	<b>¡Error! Marcador no definido.</b>
5.1.1. Ventajas y desventajas de utilizar los servicios en la nube	<b>¡Error! Marcador no definido.</b>
5.1.2. Servicios en la nube ofrecidos por Google	<b>¡Error! Marcador no definido.</b>
5.1.2.1. Registrarse en los servicios en la nube de Google	<b>¡Error! Marcador no definido.</b>
5.1.2.2. Servicios en la nube de Google .	<b>¡Error! Marcador no definido.</b>
5.2. Notificaciones push .....	<b>¡Error! Marcador no definido.</b>
5.2.1. Servicio Google Cloud Messaging .....	<b>¡Error! Marcador no definido.</b>
5.2.2. Activar Google Cloud Messaging en Google API Console	<b>¡Error! Marcador no definido.</b>
5.3. Aplicación cliente Google Cloud Messaging .....	131
5.3.1. Usar Google Play Services en un proyecto .....	131
5.3.2. Crear aplicación cliente GCM .....	131
5.3.3. Registrar dispositivo en GCM .....	134
5.3.4. Desregistrar dispositivo en GCM .....	136
5.3.5. Actualización del identificador desde GCM .....	138
5.3.6. Recepción de mensajes CGM en el dispositivo .....	138
5.3.7. Mostrar notificaciones en la barra de estado .....	138
5.4. Aplicación servidor Google Cloud Messaging .....	141
5.5. Almacenamiento en la nube .....	143
5.5.1. Almacenamiento en Google Drive .....	<b>¡Error! Marcador no definido.</b>

---

5.5.2. Google Drive API .....	<b>¡Error! Marcador no definido.</b>
5.5.3. Crear una aplicación Android para Google Drive .....	143
5.5.3.1. Extracción de la huella digital (SHA1) .....	143
5.5.3.2. Habilitar el servicio Google Drive API .....	<b>¡Error! Marcador no definido.</b>
5.5.3.3. Autorizar el acceso a Google Drive .....	143
5.5.3.4. Subir ficheros a Google Drive .....	147
5.6. Servicio de Backup de Google .....	151
5.6.1. Fundamentos .....	<b>¡Error! Marcador no definido.</b>
5.6.2. Declaración del agente de copia de seguridad en Manifest .....	151
5.6.3. Registro del servicio Android Backup .....	151
5.6.4. BackupAgent .....	<b>¡Error! Marcador no definido.</b>
5.6.5. BackupAgentHelper .....	152
5.6.5.1. Copia de seguridad de SharedPreferences .....	152
5.6.5.2. Copia de seguridad de archivos de almacenamiento interno .....	152
5.6.6. Comprobación de la versión al restaurar los datos .....	<b>¡Error! Marcador no definido.</b>
5.6.7. Solicitud de copia de seguridad y restauración .....	<b>¡Error! Marcador no definido.</b>
5.6.8. Un ejemplo paso a paso .....	152
5.6.9. Auto Backup for Apps .....	153
<b>CAPÍTULO 6. Aplicaciones web en Android .....</b>	<b>157</b>
6.1. Introducción a la tecnología web .....	157
6.1.1. Una aplicación web de ejemplo: 3 en raya .....	157
6.1.2. Aplicación web <i>online</i> y <i>offline</i> .....	162
6.2. Uso de WebView .....	163
6.2.1. Mostrar contenido web usando una intención .....	163
6.2.2. Uso de un WebView para mostrar contenido web .....	163
6.2.3. Aspectos básicos de un WebView .....	<b>¡Error! Marcador no definido.</b>
6.2.3.1. Evitar el reinicio de la actividad .....	164
6.2.3.2. Abrir los enlaces en el WebView .....	164
6.2.3.3. Opciones de inicio .....	164
6.2.3.4. Barra de progreso .....	164
6.2.3.5. Navegación .....	165
6.2.3.6. Controlar el botón «Volver» .....	166

6.2.3.7. Capturar alertas JavaScript.....	167
6.2.3.8. Gestión de errores.....	167
6.2.3.9. Descargas .....	167
6.2.3.10. Conectividad.....	169
6.3. <i>Diseño web en Android</i> .....	170
6.3.1. Área de visualización y escalado .....	170
6.3.2. Escalado .....	<b>¡Error! Marcador no definido.</b>
6.3.3. Densidad de pantalla del dispositivo .....	171
6.3.4. Depuración remota en Android con Chrome.....	<b>¡Error! Marcador no definido.</b>
6.4. <i>Aplicaciones híbridas</i> .....	171
6.5. <i>Alternativas en la programación independiente de la plataforma para móviles</i> .....	173
6.5.1. Jquery Mobile .....	173
6.5.1.1. Crear una página básica .....	173
6.5.1.2. Añadir contenido .....	<b>¡Error! Marcador no definido.</b>
6.5.1.3. Crear una lista .....	175
6.5.1.4. Añadir un deslizador.....	177
6.5.1.5. Crear un botón .....	177
6.5.1.6. Temas .....	177
<b>CAPÍTULO 7. Programación en código nativo</b> .....	181
7.1. <i>Android NDK</i> .....	<b>¡Error! Marcador no definido.</b>
7.1.1. Cuándo utilizar código nativo .....	<b>¡Error! Marcador no definido.</b>
7.1.2. Contenido de Android NDK .....	<b>¡Error! Marcador no definido.</b>
7.1.2.1. Las herramientas de desarrollo..	<b>¡Error! Marcador no definido.</b>
7.1.2.2. Documentación .....	<b>¡Error! Marcador no definido.</b>
7.1.2.3. Aplicaciones de ejemplo .....	<b>¡Error! Marcador no definido.</b>
7.2. <i>Instalación de Android NDK</i> .....	181
7.2.1. Instalación Android NDK en Android Studio (Windows).....	<b>¡Error! Marcador no definido.</b>
7.2.2. Un primer ejemplo con Android NDK .....	181
7.3. <i>Funcionamiento y estructura de Android NDK</i> .....	181
7.3.1. Desarrollo práctico de Android NDK.....	<b>¡Error! Marcador no definido.</b>
7.3.2. Situación del código fuente nativo.....	<b>¡Error! Marcador no definido.</b>

---

7.3.2.1. Fichero <i>Android.mk</i> .....	182
7.3.2.2. Fichero <i>Application.mk</i> (opcional) .....	183
7.3.2.3. NDKLa herramienta <i>ndk-build</i> .....	183
7.4. Interfaz entre JAVA y C/C++ (JNI) .....	183
7.4.1. Librerías de enlace estático y dinámico ¡Error! Marcador no definido.	
7.4.2. Tipos fundamentales, referencias y <i>arrays</i> .....	183
7.4.3. Desarrollo paso a paso de un programa mediante JNI (I) .....	184
7.4.3.1. Declaración del método nativo y creación del archivo <i>Android.mk</i> .....	184
7.4.3.2. Creación del fichero de cabecera nativo .....	185
7.4.3.3. Implementación del método nativo .....	185
7.4.3.4. Compilación del fichero nativo ... ¡Error! Marcador no definido.	
7.4.3.5. Ejecución de la aplicación .....	¡Error! Marcador no definido.
7.4.4. Acceso a métodos Java desde código nativo (JNI <i>callback</i> ) .....	185
7.4.4.1. Métodos de instancia .....	185
7.4.4.2. Métodos de clase .....	186
7.4.4.3. Invocar constructores .....	187
7.4.5. Excepciones .....	¡Error! Marcador no definido.
7.5. Rendimiento de aplicaciones con código nativo .....	190
7.6. Procesado de imagen con código nativo .....	193
<b>CAPÍTULO 8. Redes sociales: Facebook y Twitter</b> .....	199
8.1. <i>Android y Facebook</i> .....	199
8.1.1. Preliminares .....	¡Error! Marcador no definido.
8.1.1.1. Darse de alta en Facebook como desarrollador ¡Error! Marcador no definido.	
8.1.1.2. SDK de Facebook para Android ¡Error! Marcador no definido.	
8.1.1.3. Configurando nuestra nueva aplicación (en Facebook) .....	199
8.1.1.4. Algunas consideraciones .....	¡Error! Marcador no definido.
8.1.2. Nuestro proyecto Android .....	200
8.1.2.1. Importando la biblioteca Facebook-Android-SDK en Android Studio .....	¡Error! Marcador no definido.
8.1.2.2. Configurando nuestra aplicación (en Facebook) ¡Error! Marcador no definido.	
8.1.2.3. Configurando el código de nuestra aplicación .....	200
8.1.3. Aplicación de ejemplo (Graph API) .....	201

8.1.4. Aplicación de ejemplo (Share Dialog) .....	213
8.2. <i>Android y Twitter</i> .....	220
8.2.1. Instalando Fabric en Android Studio ....	<b>¡Error! Marcador no definido.</b>
8.2.2. Configurando nuestra aplicación en Twitter Developers	<b>¡Error! Marcador no definido.</b>
8.2.3. Configurando Fabric en nuestra aplicación de Android Studio .....	220
8.2.4. Aplicación de Ejemplo .....	220
<b>CAPÍTULO 9. Ingeniería inversa en Android</b> .....	224
9.1. <i>El formato APK</i> .....	224
9.2. <i>Decompilando aplicaciones Android</i> .....	225
9.2.1. La máquina virtual Dalvik/ART .....	225
9.2.2. Decompilando aplicaciones Android .....	225
9.3. <i>Modificando aplicaciones Android</i> .....	226
9.3.1. Modificando recursos binarios de una aplicación	<b>¡Error! Marcador no definido.</b>
9.3.2. Modificando recursos XML de una aplicación .....	226
9.3.3. Modificando el código de una aplicación .....	227
9.4. <i>Ofuscación del código</i> .....	228
9.5. <i>Obtención de licencias con Google Play</i> .....	230
9.5.1. Cómo funciona el servicio de licencias.	<b>¡Error! Marcador no definido.</b>
9.5.2. Como añadir una licencia a nuestra aplicación .....	230
9.5.3. ¿Qué es una política de licencia? .....	<b>¡Error! Marcador no definido.</b>
9.6. <i>Cómo evitar que se elimine la verificación de licencia en nuestras aplicaciones</i> .....	232
9.6.1. Ingeniería inversa en una aplicación con licencia .....	232
9.6.2. Primera contramedida: ofuscar el código .....	234
9.6.3. Segunda contramedida: no usar la librería LVL estándar .....	234
9.6.4. Tercera contramedida: verificar que no ha modificado nuestra APK.....	236

---

# Lista de acrónimos

API	<i>Application Programming Interface</i>
APK	<i>Application Package File</i>
ARM	<i>Advanced RISC Machine</i>
ASP	<i>Active Server Pages</i>
AVD	<i>Android Virtual Device</i>
BGMR	<i>Backup Manager</i>
C2DM	<i>Cloud to Device Messaging</i>
CSS	<i>Cascading Style Sheets</i>
DOM	<i>Modelo de Objetos del Documento</i>
DPI	<i>Dots Per Inch</i>
DVM	<i>Dalvik Virtual Machine</i>
GCM	<i>Google Cloud Message</i>
HTML	<i>Hyper Text Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IGU	<i>Interfaz Gráfica de Usuario</i>
IU	<i>Interfaz de Usuario</i>
Java SE	<i>Java Standard Edition</i>
JDK	<i>Java Development Kit</i>
JNI	<i>Java Native Interface</i>
JSON	<i>JavaScript Object Notation</i>
JSR	<i>Java Specification Request</i>
JVM	<i>Java Virtual Machine</i>
Mbps	<i>Megabits per second</i>
MIME	<i>Multipurpose Internet Mail Extensions</i>
MIPS	<i>Microprocessor without Interlocked Pipeline Stages</i>
NDK	<i>Native Development Kit</i>
PHP	<i>Hypertext Preprocessor</i>
REST	<i>Representational State Transfer</i>
RFCOMM	<i>Radio Frequency Communication</i>

SD	<i>Secure Digital</i>
SDK	<i>Software Developers Kit</i>
SDP	<i>Service Discovery Protocol</i>
SMS	<i>Short Message Service</i>
SOAP	<i>Simple Object Access Protocol</i>
SQL	<i>Structured Query Language</i>
SVG	<i>Scalable Vector Graphics</i>
TCP	<i>Transmission Control Protocol</i>
UI	<i>User Interface</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Universal Resource Locator</i>
USB	<i>Universal Serial Bus</i>
UUID	<i>Universally Unique Identifier</i>
XML	<i>Extensible Markup Language</i>
XMPP	<i>Protocolo extensible de mensajería y comunicación de presencia</i>



# CAPÍTULO 1.

## Diseño avanzado de interfaces de usuario

Por JESÚS TOMÁS

### 1.1. Acceder a objetos globales de la aplicación

#### 1.1.1. La clase Application

```
public class Aplicacion extends Application {
    private int saldo;

    @Override
    public void onCreate() {
        SharedPreferences pref = getSharedPreferences("pref", MODE_PRIVATE);
        int saldo = pref.getInt("saldo_inicial", -1);
    }

    public int getSaldo(){
        return saldo;
    }

    public void setSaldo(int saldo){
        this.saldo=saldo;
    }
}

...
<application
    android:name="Aplicacion"
    android:allowBackup="true"
```

```
android:icon="@drawable/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme">
...
```

```
Aplicacion aplicacion = (Aplicacion) contexto.getApplication();
```

```
Aplicacion aplicacion = (Aplicacion) getApplication();
```

```
int miSaldo = ((Aplicacion) getApplication()).getSaldo();
```

```
public class Aplicacion extends Application {

    private Vector<Libro> vectorLibros;
    private AdaptadorLibros adaptador;

    @Override
    public void onCreate() {
        vectorLibros = Libro.ejemploLibros();
        adaptador = new AdaptadorLibros (this, vectorLibros);
    }

    public AdaptadorLibros getAdaptador() {
        return adaptador;
    }

    public Vector<Libro> getVectorLibros() {
        return vectorLibros;
    }
}
```

```
<application
    android:name="Aplicacion"
    android:allowBackup="true"
    ...
```

```
public class Libro {
    public String titulo;
    public String autor;
    public int recursoImagen;
    public String urlAudio;
    public String genero; // Género literario
    public Boolean novedad; // Es una novedad
    public Boolean leído; // Leído por el usuario
    public final static String G_TODOS = "Todos los géneros";
    public final static String G_EPICO = "Poema épico";
    public final static String G_S_XIX = "Literatura siglo XIX";
}
```

```

public final static String G_SUSPENSE = "Suspense";
public final static String[] G_ARRAY = new String[] {G_TODOS, G_EPICO,
    G_S_XIX, G_SUSPENSE };

public Libro(String titulo, String autor, int recursoImagen,
    String urlAudio, String genero, Boolean novedad, Boolean leído) {
    this.titulo = titulo; this.autor = autor;
    this.recursoImagen = recursoImagen; this.urlAudio = urlAudio;
    this.genero = genero; this.novedad = novedad; this.leído = leído;
}

public static Vector<Libro> ejemploLibros() {
    final String SERVIDOR =
        "http://www.dcomg.upv.es/~jtomas/android/audiolibros/";
    Vector<Libro> libros = new Vector<Libro>();
    libros.add(new Libro("Kappa", "Akutagawa",
        R.drawable.kappa, SERVIDOR+"kappa.mp3",
        Libro.G_S_XIX, false, false));
    libros.add(new Libro("Avecilla", "Alas Clarín, Leopoldo",
        R.drawable.avecilla, SERVIDOR+"avecilla.mp3",
        Libro.G_S_XIX, true, false));
    libros.add(new Libro("Divina Comedia", "Dante",
        R.drawable.divinacomedia, SERVIDOR+"divina_comedia.mp3",
        Libro.G_EPICO, true, false));
    libros.add(new Libro("Viejo Pancho, El", "Alonso y Trelles, José",
        R.drawable.viejo_pancho, SERVIDOR+"viejo_pancho.mp3",
        Libro.G_S_XIX, true, true));
    libros.add(new Libro("Canción de Rolando", "Anónimo",
        R.drawable.cancion_rolando, SERVIDOR+"cancion_rolando.mp3",
        Libro.G_EPICO, false, true));
    libros.add(new Libro("Matrimonio de sabuesos", "Agata Christie",
        R.drawable.matrimonio_sabuesos, SERVIDOR+"matrim_sabuesos.mp3",
        Libro.G_SUSPENSE, false, true));
    libros.add(new Libro("La iliada", "Homero",
        R.drawable.iliada, SERVIDOR+"la_iliada.mp3",
        Libro.G_EPICO, true, false));
    return libros;
}
}

```

### 1.1.2. Uso del patrón Singleton

```

public class Singleton {

    // Esta será la instancia única de esta clase
    private static Singleton INSTANCIA = new Singleton();

    // El constructor es private para evitar su acceso desde fuera.
    private Singleton() {}

    // Método para obtener la única instancia de la clase
    public static Singleton getInstancia() {

```

```
        return INSTANCIA;
    }
}
```

```
Singleton referencia = Singleton.getInstance();
```

```
public class Saldo {
    private static Saldo INSTANCIA = new Saldo();
    // En Android casi siempre necesitas conocer el contexto
    private Context contexto;
    // Otras variables de la clase
    private int saldo = -1;

    private Saldo() {}

    public static Saldo getInstance() {
        return INSTANCIA;
    }

    // Método para inicializar el objeto
    public void inicializa(Context contexto){
        this.contexto = contexto;
        SharedPreferences pref = contexto.getSharedPreferences("pref",
            Context.MODE_PRIVATE);
        int saldo = pref.getInt("saldo_inicial", -1);
    }

    public int getSaldo() {
        return saldo;
    }

    public void putSaldo(int saldo) {
        this.saldo = saldo;
    }
}
```

```
Saldo saldo = Saldo.getInstance();
saldo.inicializa(contexto);
int n = saldo.getSaldo();
```

## 1.2. Material Design

```
dependencies {
    ...
    compile 'com.android.support:appcompat-v7:23.1.1'
```

```

compile 'com.android.support:design:23.1.1'
compile 'com.android.support:recyclerview-v7:23.1.1'
}

```

### 1.2.1. Definición de la paleta de colores de la aplicación

```

<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#8BC34A</color>
</resources>

```

```

<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
    ...

```

```

<manifest ...>
    <application
        ...
        android:theme="@style/AppTheme">
    ...

```

## 1.3. RecyclerView

```

dependencies {
    ...
    compile 'com.android.support:recyclerview-v7:23.1.1'
}

```

```

<android.support.v7.widget.RecyclerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/recycler_view"
    android:scrollbars="vertical"
    android:background="#ffffff" />

```

```

public class MainActivity extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.LayoutManager layoutManager;
}

```

```

@Override protected void onCreate(Bundle savedInstanceState) {
    ...
    Aplicacion app = (Aplicacion) getApplication();
    recyclerView = (RecyclerView) findViewById(R.id.recycler_view);
    recyclerView.setAdapter(app.getAdaptador());
    layoutManager = new LinearLayoutManager(this);
    recyclerView.setLayoutManager(layoutManager);
}
}

```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:id="@+id/portada"
        android:layout_width="120dp"
        android:layout_height="180dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_margin="12dp"
        android:src="@drawable/books" />
    <TextView
        android:id="@+id/titulo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Titulo"
        android:layout_alignParentLeft="true"
        android:layout_below="@id/portada"
        android:gravity="center"
        android:lines="2"
        android:textColor="#000000"
        android:textSize="18dp" />
</RelativeLayout>

```

```

public class AdaptadorLibros extends
    RecyclerView.Adapter<AdaptadorLibros.ViewHolder> {
    private LayoutInflater inflater; //Crea Layouts a partir del XML
    protected Vector<Libro> vectorLibros; //Vector con libros a visualizar
    private Context contexto;

    public AdaptadorLibros(Context contexto, Vector<Libro> vectorLibros) {
        inflater = (LayoutInflater) contexto
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        this.vectorLibros = vectorLibros;
        this.contexto = contexto;
    }

    //Creamos nuestro ViewHolder, con los tipos de elementos a modificar
    public static class ViewHolder extends RecyclerView.ViewHolder {
        public ImageView portada;
    }
}

```

```

    public TextView titulo;

    public ViewHolder(View itemView) {
        super(itemView);
        portada = (ImageView) itemView.findViewById(R.id.portada);
        portada.setScaleType(ImageView.ScaleType.CENTER_INSIDE);
        titulo = (TextView) itemView.findViewById(R.id.titulo);
    }
}

// Creamos el ViewHolder con las vista de un elemento sin personalizar
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    // Inflamos la vista desde el xml
    View v = inflater.inflate(R.layout.elemento_selector, null);
    return new ViewHolder(v);
}

// Usando como base el ViewHolder y lo personalizamos
@Override
public void onBindViewHolder(ViewHolder holder, int posicion) {
    Libro libro = vectorLibros.elementAt(posicion);
    holder.portada.setImageResource(libro.recursoImagen);
    holder.titulo.setText(libro.titulo);
}

// Indicamos el número de elementos de la lista
@Override public int getItemCount() {
    return vectorLibros.size();
}
}

```

```
layoutManager = new LinearLayoutManager(this, 2);
```

```
layoutManager = new GridLayoutManager(this, 2,
    LinearLayoutManager.HORIZONTAL, false);
```

```
private View.OnClickListener onClickListener;
```

```
public void setOnItemClickListener(View.OnClickListener onClickListener) {
    this.onClickListener = onClickListener;
}

```

```
@Override
```

```
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
    // Inflamos la vista desde el xml  
    View v = inflater.inflate(R.layout.elemento_selector, null);  
    v.setOnClickListener(onClickListener);  
    return new ViewHolder(v);  
}
```

```
app.getAdaptador().setOnItemClickListener(new View.OnClickListener(){  
    @Override public void onClick(View v) {  
        Toast.makeText(MainActivity.this, "Seleccionado el elemento: "  
            + recyclerView.getChildAdapterPosition(v),  
            Toast.LENGTH_SHORT).show();  
    }  
});
```

## 1.4. Fragments

### 1.4.1. Insertar *fragments* desde XML

```
<android.support.v7.widget.RecyclerView  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/recycler_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="#ffffff"  
    android:scrollbars="vertical" />
```

```
public class SelectorFragment extends Fragment {  
    private Activity actividad;  
    private RecyclerView recyclerView;  
    private AdaptadorLibros adaptador;  
    private Vector<Libro> vectorLibros;  
  
    @Override public void onAttach(Context contexto) {  
        super.onAttach(contexto);  
        if (contexto instanceof Activity) {  
            this.actividad = (Activity) contexto;  
            Aplicacion app = (Aplicacion) actividad.getApplication();  
            adaptador = app.getAdaptador();  
            vectorLibros = app.getVectorLibros();  
        }  
    }  
  
    @Override public View onCreateView(LayoutInflater inflater, ViewGroup  
        contenedor, Bundle savedInstanceState) {  
        View vista = inflater.inflate(R.layout.fragment_selector,  
            contenedor, false);  
        recyclerView = (RecyclerView) vista.findViewById(  

```



```

R.id.recycler_view);
recyclerView.setLayoutManager(new GridLayoutManager(actividad,2));
recyclerView.setAdapter(adaptador);
adaptador.setOnItemClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(actividad, "Seleccionado el elemento: "
            + recyclerView.getChildAdapterPosition(v),
            Toast.LENGTH_SHORT).show();
    }
});
return vista;
}
}

```

```

<LinearLayout 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:orientation="horizontal"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.audiolibros.MainActivity"
    tools:showIn="@layout/activity_main">
    <fragment
        android:id="@+id/selector_fragment"
        android:name="com.example.audiolibros.fragments.SelectorFragment"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1" />
</LinearLayout>

```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/fragment_detalle"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffffff"
    tools:context=".MainActivity" >
    <TextView
        android:id="@+id/titulo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:gravity="center"
        android:text="Titulo"
        android:textAlignment="center"
        android:textColor="@android:color/black"
        android:textAppearance="?android:attr/textAppearanceLarge" />

```

```

<TextView
    android:id="@+id/autor"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/titulo"
    android:gravity="center"
    android:text="Autor"
    android:textAlignment="center"
    android:textColor="@android:color/darker_gray"
    android:textAppearance="?android:attr/textAppearanceMedium" />
<ImageView
    android:id="@+id/portada"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/autor"
    android:layout_centerHorizontal="true"
    android:src="@mipmap/ic_launcher" />
</RelativeLayout>

```

```

<fragment
    android:id="@+id/detalle_fragment"
    android:name="com.example.audiolibros.fragments.DetalleFragment"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1" />

```

```

public class DetalleFragment extends Fragment implements
    View.OnTouchListener, MediaPlayer.OnPreparedListener,
    MediaController.MediaPlayerControl {
    public static String ARG_ID_LIBRO = "id_libro";
    MediaPlayer mediaPlayer;
    MediaController mediaController;

    @Override public View onCreateView(LayoutInflater inflater, ViewGroup
        contenedor, Bundle savedInstanceState) {
        View vista = inflater.inflate(R.layout.fragment_detalle,
            contenedor, false);

        Bundle args = getArguments();
        if (args != null) {
            int position = args.getInt(ARG_ID_LIBRO);
            ponInfoLibro(position, vista);
        } else {
            ponInfoLibro(0, vista);
        }
        return vista;
    }

    private void ponInfoLibro(int id, View vista) {
        Libro libro = ((Aplicacion) getActivity().getApplication())

```

```

        .getVectorLibros().elementAt(id);
        ((TextView) vista.findViewById(R.id.titulo)).setText(libro.titulo);
        ((TextView) vista.findViewById(R.id.autor)).setText(libro.autor);
        ((ImageView) vista.findViewById(R.id.portada))
            .setImageResource(libro.recursoImagen);
        vista.setOnTouchListener(this);
        if (mediaPlayer != null){
            mediaPlayer.release();
        }
        mediaPlayer = new MediaPlayer();
        mediaPlayer.setOnPreparedListener(this);
        mediaController = new MediaController(getActivity());
        Uri audio = Uri.parse(libro.urlAudio);
        try {
            mediaPlayer.setDataSource(getActivity(), audio);
            mediaPlayer.prepareAsync();
        } catch (IOException e) {
            Log.e("Audiolibros", "ERROR: No se puede reproducir "+audio,e);
        }
    }

    public void ponInfoLibro(int id) {
        ponInfoLibro(id, getView());
    }

    @Override public void onPrepared(MediaPlayer mediaPlayer) {
        Log.d("Audiolibros", "Entramos en onPrepared de MediaPlayer");
        mediaPlayer.start();
        mediaController.setMediaPlayer(this);
        mediaController.setAnchorView(getView().findViewById(
            R.id.fragment_detalle));
        mediaController.setEnabled(true);
        mediaController.show();
    }

    @Override public boolean onTouch(View vista, MotionEvent evento) {
        mediaController.show();
        return false;
    }

    @Override public void onStop() {
        mediaController.hide();
        try {
            mediaPlayer.stop();
            mediaPlayer.release();
        } catch (Exception e) {
            Log.d("Audiolibros", "Error en mediaPlayer.stop()");
        }
        super.onStop();
    }

    @Override public boolean canPause() {
        return true;
    }

```

```
}

@Override public boolean canSeekBackward() {
    return true;
}

@Override public boolean canSeekForward() {
    return true;
}

@Override public int getBufferPercentage() {
    return 0;
}

@Override public int getCurrentPosition() {
    try {
        return mediaPlayer.getCurrentPosition();
    } catch (Exception e) {
        return 0;
    }
}

@Override public int getDuration() {
    return mediaPlayer.getDuration();
}

@Override public boolean isPlaying() {
    return mediaPlayer.isPlaying();
}

@Override public void pause() {
    mediaPlayer.pause();
}

@Override public void seekTo(int pos) {
    mediaPlayer.seekTo(pos);
}

@Override public void start() {
    mediaPlayer.start();
}

@Override public int getAudioSessionId() {
    return 0;
}
}
```

```
<uses-permission android:name="android.permission.INTERNET"/>
```

### 1.4.2. Insertar *fragments* desde código

```
<LinearLayout 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:id="@+id/contenedor_pequeno"
    android:orientation="horizontal"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.audiolibros.MainActivity"
    tools:showIn="@layout/activity_main">
</LinearLayout>
```

```
if ((findViewById(R.id.contenedor_pequeno) != null) &&
    (getSupportFragmentManager().findFragmentById(
        R.id.contenedor_pequeno) == null)){
    SelectorFragment primerFragment = new SelectorFragment();
    getSupportFragmentManager().beginTransaction()
        .add(R.id.contenedor_pequeno, primerFragment).commit();
}
```

### 1.4.3. Comunicación e intercambio de *fragments*

```
adaptador.setOnItemClickListener(new View.OnClickListener(){
    @Override public void onClick(View v) {
        Toast.makeText(actividad, "Seleccionado el elemento: " + recycler
            ((MainActivity) actividad).mostrarDetalle(
                recyclerView.getChildAdapterPosition(v));
    }
});
```

```
public void mostrarDetalle(int id) {
    DetalleFragment detalleFragment = (DetalleFragment)
        getSupportFragmentManager().findFragmentById(R.id.detalle_fragment);
    if (detalleFragment != null) {
        detalleFragment.ponInfoLibro(id);
    } else {
        DetalleFragment nuevoFragment = new DetalleFragment();
        Bundle args = new Bundle();
        args.putInt(DetalleFragment.ARG_ID_LIBRO, id);
        nuevoFragment.setArguments(args);
        FragmentTransaction transaccion = getSupportFragmentManager()
            .beginTransaction();
        transaccion.replace(R.id.contenedor_pequeno, nuevoFragment);
        transaccion.addToBackStack(null);
    }
}
```

```

        transaccion.commit();
    }
}

```

## 1.5. Otros elementos del interfaz de usuario

### 1.5.1. Menús contextuales

```

adaptador.setOnItemClickListener(new View.OnClickListener() {
    public boolean onLongClick(final View v) {
        final int id = recicladorView.getChildAdapterPosition(v);
        AlertDialog.Builder menu = new AlertDialog.Builder(actividad);
        CharSequence[] opciones = { "Compartir", "Borrar ", "Insertar" };
        menu.setItems(opciones, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int opcion) {
                switch (opcion) {
                    case 0: //Compartir
                        Libro libro = vectorLibros.elementAt(id);
                        Intent i = new Intent(Intent.ACTION_SEND);
                        i.setType("text/plain");
                        i.putExtra(Intent.EXTRA_SUBJECT, libro.titulo);
                        i.putExtra(Intent.EXTRA_TEXT, libro.urlAudio);
                        startActivity(Intent.createChooser(i, "Compartir"));
                        break;
                    case 1: //Borrar
                        vectorLibros.remove(id);
                        adaptador.notifyDataSetChanged();
                        break;
                    case 2: //Insertar
                        vectorLibros.add(vectorLibros.elementAt(id));
                        adaptador.notifyDataSetChanged();
                        break;
                }
            }
        });
        menu.create().show();
        return true;
    }
});

```

```
private View.OnClickListener onLongClickListener;
```

```

public void setOnItemLongClickListener(View.OnClickListener
                                         onLongClickListener) {
    this.onLongClickListener = onLongClickListener;
}

```

```
@Override
```

```
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    ...
    v.setOnLongClickListener(onLongClickListener);
    return new ViewHolder(v);
}
```

### 1.5.1.1. Barra de acciones con *ActionBar*

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto" >
    <item
        android:id="@+id/menu_preferencias"
        android:icon="@android:drawable/ic_menu_preferences"
        android:orderInCategory="95"
        app:showAsAction="ifRoom"
        android:title="Preferencias"/>
    <item
        android:id="@+id/menu_ultimo"
        android:icon="@android:drawable/ic_menu_set_as"
        android:orderInCategory="100"
        app:showAsAction="ifRoom|withText"
        android:title="Último leído"/>
    <item
        android:id="@+id/menu_buscar"
        app:actionViewClass="android.support.v7.widget.SearchView"
        android:orderInCategory="90"
        android:icon="@android:drawable/ic_menu_search"
        app:showAsAction="ifRoom|collapseActionView"
        android:title="Buscar"/>
    <item
        android:id="@+id/menu_acerca"
        android:orderInCategory="110"
        app:showAsAction="never"
        android:title="Acerca de..."/>
</menu>
```

```
@Override public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.menu_preferencias) {
        Toast.makeText(this, "Preferencias", Toast.LENGTH_LONG).show();
        return true;
    } else if (id == R.id.menu_ultimo) {
        irUltimoVisitado();
        return true;
    } else if (id == R.id.menu_buscar) {
        return true;
    } else if (id == R.id.menu_acerca) {
```

```

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setMessage("Mensaje de Acerca De");
        builder.setPositiveButton(android.R.string.ok, null);
        builder.create().show();
        return true;
    }
    return super.onOptionsItemSelected(item);
}

```

```

public void irUltimoVisitado() {
    SharedPreferences pref = getSharedPreferences(
        "com.example.audiolibros_internal", MODE_PRIVATE);
    int id = pref.getInt("ultimo", -1);
    if (id >= 0) {
        mostrarDetalle(id);
    } else {
        Toast.makeText(this, "Sin última vista", Toast.LENGTH_LONG).show();
    }
}

```

```

SharedPreferences pref = getSharedPreferences(
    "com.example.audiolibros_internal", MODE_PRIVATE);
SharedPreferences.Editor editor = pref.edit();
editor.putInt("ultimo", id);
editor.commit();

```

### 1.5.1.2. Barra de acciones con Toolbar

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:minHeight="?attr/actionBarSize"
        android:popupTheme="@style/ThemeOverlay.AppCompat.Light"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar" />
    <!--Contenido del Layout -->
</LinearLayout>

```

```

// Toolbar
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
supportActionBar(toolbar);

```

### 1.5.1.3. Configurar la barra de acciones desde un *fragment*

```

setHasOptionsMenu(true);

```



```

@Override
public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
    inflater.inflate(R.menu.menu_selector, menu);
    super.onCreateOptionsMenu(menu, inflater);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.menu_ultimo) {
        ((MainActivity) actividad).irUltimoVisitado();
        return true;
    } else if (id == R.id.menu_buscar) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

```

### 1.5.2. Vistas animadas: CoordinationLayout, AppBarLayout, FloatingActionButton y SnackBar

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ...
    //Barra de acciones
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    //Botón flotante
    FloatingActionButton fab = (FloatingActionButton) findViewById(
                                                                    R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action",
                           Snackbar.LENGTH_LONG).setAction("Action", null).show();
        }
    });
    ...
}

```

```

case 1: //Borrar
    Snackbar.make(v, "¿Estás seguro?", Snackbar.LENGTH_LONG)
        .setAction("SI", new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                vectorLibros.remove(id);
                adaptador.notifyDataSetChanged();
            }
        })

```

```
        }  
    })  
    .show();  
break;
```

case 2: //Insertar

```
...  
Snackbar.make(v,"Libro insertado", Snackbar.LENGTH_INDEFINITE)  
    .show();  
break;
```

case 2: //Insertar

```
...  
Snackbar.make(v,"Libro insertado", Snackbar.LENGTH_INDEFINITE)  
    .setAction("OK", new View.OnClickListener() {  
        @Override public void onClick(View view) { }  
    })  
    .show();  
break;
```

```
<android.support.design.widget.CoordinatorLayout ...>  
    <android.support.design.widget.AppBarLayout ...>  
        <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"  
            app:layout_scrollFlags="scroll|enterAlways"/>  
    </android.support.design.widget.AppBarLayout>  
    <include layout="@layout/content_main" />  
    <android.support.design.widget.FloatingActionButton .../>  
</android.support.design.widget.CoordinatorLayout>
```

```
<android.support.design.widget.CoordinatorLayout ...>  
    <android.support.design.widget.AppBarLayout  
        android:layout_width="match_parent"  
        android:layout_height="170dp"  
        android:theme="@style/AppTheme.AppBarOverlay">  
        <android.support.design.widget.CollapsingToolbarLayout  
            android:id="@+id/collapsing"  
            android:layout_width="match_parent"  
            android:layout_height="match_parent"  
            app:layout_scrollFlags="scroll|enterAlways">  
            <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"
        app:layout_scrollFlags="scroll|enterAlways"
        app:layout_collapseMode="pin"/>
    </android.support.design.widget.CollapsingToolbarLayout>
</android.support.design.widget.AppBarLayout>
<include layout="@layout/content_main" />
<android.support.design.widget.FloatingActionButton .../>
</android.support.design.widget.CoordinatorLayout>

```

```

<RelativeLayout
<android.support.v4.widget.NestedScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/detalle_fragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffffff"
    tools:context=".MainActivity">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <TextView .../>
        <TextView .../>
        <ImageView .../>
    </RelativeLayout>
</android.support.v4.widget.NestedScrollView>

```

```

@Override public void onPrepared(MediaPlayer mediaPlayer) {
    Log.d("Audiolibros", "Entramos en onPrepared de MediaPlayer");
    mediaPlayer.start();
    mediaController.setMediaPlayer(this);
    mediaController.setAnchorView(getView());
    mediaController.setPadding(0, 0, 0, 110);
    mediaController.setEnabled(true);
    mediaController.show();
}

```

```

<android.support.design.widget.CoordinatorLayout ...>
    <android.support.design.widget.AppBarLayout
        android:id="@+id/appBarLayout"
        android:layout_width="match_parent"
        android:layout_height="170dp"
        android:theme="@style/AppTheme.AppBarOverlay">
        <android.support.design.widget.CollapsingToolbarLayout ...>
            <android.support.v7.widget.Toolbar .../>
        </android.support.design.widget.CollapsingToolbarLayout>
    </android.support.design.widget.AppBarLayout>
    <include layout="@layout/content_main" />
    <android.support.design.widget.FloatingActionButton

```

```
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        android:src="@android:drawable/ic_dialog_email"
        app:layout_anchor="@id/appBarLayout"
        app:layout_anchorGravity="bottom|right|end"/>
    </android.support.design.widget.CoordinatorLayout>
```

### 1.5.3. Pestañas con TabLayout

```
dependencies {
    ...
    compile 'com.android.support:design:23.1.1'
}
```

```
<android.support.design.widget.CoordinatorLayout ...>
    <android.support.design.widget.AppBarLayout ...>
        <android.support.v7.widget.Toolbar .../>
        <android.support.design.widget.TabLayout
            android:id="@+id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
    </android.support.design.widget.AppBarLayout>
    <include layout="@layout/content_main" />
    <android.support.design.widget.FloatingActionButton .../>
</android.support.design.widget.CoordinatorLayout>
```

```
<android.support.design.widget.CoordinatorLayout ...>
    <android.support.design.widget.AppBarLayout ...>
        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/collapsing"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:layout_scrollFlags="scroll|exitUntilCollapsed"
            app:expandedTitleMarginBottom="?attr/actionBarSize">
            <android.support.design.widget.TabLayout
                android:id="@+id/tabs"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                android:layout_gravity="bottom"
                app:tabMode="scrollable"
                app:layout_anchor="@id/appBarLayout"
                app:layout_anchorGravity="bottom"/>
            <android.support.v7.widget.Toolbar .../>
        </android.support.design.widget.CollapsingToolbarLayout>
    </android.support.design.widget.AppBarLayout>
```

```

<include layout="@layout/content_main" />
<android.support.design.widget.FloatingActionButton .../>
</android.support.design.widget.CoordinatorLayout>

```

```

//Pestañas
TabLayout tabs = (TabLayout) findViewById(R.id.tabs);
tabs.addTab(tabs.newTab().setText("Todos"));
tabs.addTab(tabs.newTab().setText("Nuevos"));
tabs.addTab(tabs.newTab().setText("Leídos"));
tabs.setTabMode(TabLayout.MODE_SCROLLABLE);
tabs.setOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
    @Override public void onTabSelected(TabLayout.Tab tab) {
        switch (tab.getPosition()) {
            case 0: //Todos
                adaptador.setNovedad(false);
                adaptador.setLeido(false);
                break;
            case 1: //Nuevos
                adaptador.setNovedad(true);
                adaptador.setLeido(false);
                break;
            case 2: //Leídos
                adaptador.setNovedad(false);
                adaptador.setLeido(true);
                break;
        }
        adaptador.notifyDataSetChanged();
    }
    @Override public void onTabUnselected(TabLayout.Tab tab) {}
    @Override public void onTabReselected(TabLayout.Tab tab) {}
});

```

```

public class AdaptadorLibrosFiltro extends AdaptadorLibros {
    private Vector<Libro> vectorSinFiltro; // Vector con todos los libros
    private Vector<Integer> indiceFiltro; // Índice en vectorSinFiltro de
                                           // Cada elemento de vectorLibros

    private String busqueda = ""; // Búsqueda sobre autor o título
    private String genero = ""; // Género seleccionado
    private boolean novedad = false; // Si queremos ver solo novedades
    private boolean leido = false; // Si queremos ver solo leídos

    public AdaptadorLibrosFiltro(Context contexto,
                                   Vector<Libro> vectorLibros) {
        super(contexto, vectorLibros);
        vectorSinFiltro = vectorLibros;
        recalculaFiltro();
    }

    public void setBusqueda(String busqueda) {
        this.busqueda = busqueda.toLowerCase();
        recalculaFiltro();
    }
}

```

```
public void setGenero(String genero) {
    this.genero = genero;
    recalculaFiltro();
}

public void setNovedad(boolean novedad) {
    this.novedad = novedad;
    recalculaFiltro();
}

public void setLeido(boolean leido) {
    this.leido = leido;
    recalculaFiltro();
}

public void recalculaFiltro() {
    vectorLibros = new Vector<Libro>();
    indiceFiltro = new Vector<Integer>();
    for (int i = 0; i < vectorSinFiltro.size(); i++) {
        Libro libro = vectorSinFiltro.elementAt(i);
        if ((libro.titulo.toLowerCase().contains(búsqueda) ||
            libro.autor.toLowerCase().contains(búsqueda))
            && (libro.genero.startsWith(genero))
            && (!novedad || (novedad && libro.novedad))
            && (!leido || (leido && libro.leido))) {
            vectorLibros.add(libro);
            indiceFiltro.add(i);
        }
    }
}

public Libro getItem(int posicion) {
    return vectorSinFiltro.elementAt(indiceFiltro.elementAt(posicion));
}

public long getItemId(int posicion) {
    return indiceFiltro.elementAt(posicion);
}

public void borrar(int posicion){
    vectorSinFiltro.remove((int)getItemId(posicion));
    recalculaFiltro();
}

public void insertar(Libro libro){
    vectorSinFiltro.add(libro);
    recalculaFiltro();
}
}
```

```
private AdaptadorLibros adaptador;
```

```
private AdaptadorLibrosFiltro adaptador;
```

```
vectorLibros.remove(id);
```

```
adaptador.borrar(id);
```

```
vectorLibros.add(vectorLibros.elementAt(id));
```

```
int posicion = recyclerView.getChildLayoutPosition(v);
adaptador.insertar((Libro) adaptador.getItem(posicion));
```

```
((MainActivity) actividad).mostrarDetalle(
    recyclerView.getChildAdapterPosition(v));
```

```
((MainActivity) actividad).mostrarDetalle(
    (int) adaptador.getItemId(
        recyclerView.getChildAdapterPosition(v)));
```

```
private AdaptadorLibrosFiltro adaptador;
```

```
adaptador = ((Aplicacion) getApplicationContext()).getAdaptador();
```

## 1.6. Navigation Drawer

```
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">
    <!--El contenido de la actividad -->
    <include
        layout="@layout/app_bar_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
    <!--El contenido del Navigation Drawer -->
    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main"
        app:menu="@menu/activity_main_drawer" />
</android.support.v4.widget.DrawerLayout>
```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="160dp"
    android:background="@color/colorAccent"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="16dp"
        android:src="@android:drawable/sym_def_app_icon" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="16dp"
        android:text="Audiolibros"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="http://www.androidcurso.com" />
</LinearLayout>

```

```

<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <group android:checkableBehavior="single">
        <item android:id="@+id/nav_todos"
            android:icon="@android:drawable/ic_menu_gallery"
            android:title="Todos los géneros" />
        <item android:id="@+id/nav_epico"
            android:icon="@android:drawable/ic_menu_gallery"
            android:title="Poema épico" />
        <item android:id="@+id/nav_XIX"
            android:icon="@android:drawable/ic_menu_gallery"
            android:title="Literatura siglo XIX" />
        <item android:id="@+id/nav_suspense"
            android:icon="@android:drawable/ic_menu_gallery"
            android:title="Suspense" />
    </group>
    <item android:title="Acciones adicionales">
        <menu>
            <item android:id="@+id/nav_preferencias"
                android:icon="@android:drawable/ic_menu_manage"
                android:title="Preferencias" />

```



```

        <item android:id="@+id/nav_compartir"
            android:icon="@android:drawable/ic_menu_share"
            android:title="Compartir" />
    </menu>
</item>
</menu>

```

```

public class MainActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {

    protected void onCreate(Bundle savedInstanceState) {
        ...
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        ...
        // Navigation Drawer
        DrawerLayout drawer = (DrawerLayout) findViewById(
            R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this,
            drawer, toolbar, R.string.drawer_open, R.string.drawer_close);
        drawer.setDrawerListener(toggle);
        toggle.syncState();
        NavigationView navigationView = (NavigationView) findViewById(
            R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
    }

    @SuppressWarnings("StatementWithEmptyBody")
    @Override
    public boolean onNavigationItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.nav_todos) {
            // ...
        } else if (id == R.id.nav_epico) {
            // ...
        } else if (id == R.id.nav_XIX) {
            // ...
        } // ...
        DrawerLayout drawer = (DrawerLayout) findViewById(
            R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
        return true;
    }

    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout) findViewById(
            R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawer(GravityCompat.START);
        } else {
            super.onBackPressed();
        }
    }
}

```

```
    }
}
```

```
<string name="drawer_open">navigation drawer abierto</string>
<string name="drawer_close">navigation drawer cerrado</string>
```

```
public boolean onNavigationItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.nav_todos) {
        adaptador.setGenero("");
        adaptador.notifyDataSetChanged();
    } else if (id == R.id.nav_epico) {
        adaptador.setGenero(Libro.G_EPICO);
        adaptador.notifyDataSetChanged();
    } else if (id == R.id.nav_XIX) {
        adaptador.setGenero(Libro.G_S_XIX);
        adaptador.notifyDataSetChanged();
    } else if (id == R.id.nav_suspense) {
        adaptador.setGenero(Libro.G_SUSPENSE);
        adaptador.notifyDataSetChanged();
    }
}
```

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <gradient
        android:angle="135"
        android:centerColor="#4CAF50"
        android:endColor="#2E7D32"
        android:startColor="#81C784"
        android:type="linear" />
</shape>
```

```
android:background="@drawable/side_nav_bar"
```

### 1.6.1. Ocultar elementos de la interfaz de usuario

```
public void mostrarElementos(boolean mostrar) {
    appBarLayout.setExpanded(mostrar);
    toggle.setDrawerIndicatorEnabled(mostrar);
    if (mostrar) {
        drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_UNLOCKED);
        tabs.setVisibility(View.VISIBLE);
    } else {
        tabs.setVisibility(View.GONE);
        drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED);
    }
}
```

```
private AppBarLayout appBarLayout;
private TabLayout tabs;
private DrawerLayout drawer;
private ActionBarDrawerToggle toggle;
```

```
appBarLayout = (AppBarLayout) findViewById(R.id.appBarLayout);
```

```
@Override public void onResume(){
    DetalleFragment detalleFragment = (DetalleFragment)
        getFragmentManager().findFragmentById(R.id.detalle_fragment);
    if (detalleFragment == null ) {
        ((MainActivity) getActivity()).mostrarElementos(false);
    }
    super.onResume();
}
```

```
@Override public void onResume(){
    ((MainActivity) getActivity()).mostrarElementos(true);
    super.onResume();
}
```

```
<android.support.design.widget.CollapsingToolbarLayout
    ...
    app:layout_scrollFlags="scroll|exitUntilCollapsed"
```

```
ActionBar actionBar = getSupportActionBar();
if (actionBar != null) {
    actionBar.setDisplayHomeAsUpEnabled(true);
}
toggle.setToolbarNavigationClickListener(new View.OnClickListener() {
    @Override public void onClick(View v) {
        onBackPressed();
    }
});
```

## 1.7. Servicios de búsquedas

```
android {
    ...
    defaultConfig {
        applicationId "com.example.audiolibros"
        minSdkVersion 9 14
        ...
    }
}
```

```
}
```

```
<item android:id="@+id/menu_buscar"
    app:actionViewClass="android.support.v7.widget.SearchView"
    android:icon="@android:drawable/ic_menu_search"
    android:orderInCategory="90"
    app:showAsAction="ifRoom/collapseActionView"
    android:title="Buscar"/>
```

```
MenuItem searchItem = menu.findItem(R.id.menu_buscar);
SearchView searchView = (SearchView) searchItem.getActionView();
searchView.setOnQueryTextListener(
    new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextChanged(String query) {
            adaptador.setBusqueda(query);
            adaptador.notifyDataSetChanged();
            return false;
        }

        @Override
        public boolean onQueryTextSubmit(String query) {
            return false;
        }
    });
```

```
MenuItemCompat.setOnActionExpandListener(searchItem,
    new MenuItemCompat.OnActionExpandListener() {
        @Override
        public boolean onMenuItemActionCollapse(MenuItem item) {
            adaptador.setBusqueda("");
            adaptador.notifyDataSetChanged();
            return true; // Para permitir cierre
        }

        @Override
        public boolean onMenuItemActionExpand(MenuItem item) {
            return true; // Para permitir expansión
        }
    });
```

```
SearchManager searchManager = (SearchManager) getActivity()
    .getSystemService(Context.SEARCH_SERVICE);
MenuItem searchItem = menu.findItem(R.id.menu_buscar);
SearchView searchView = (SearchView) searchItem.getActionView();
searchView.setSearchableInfo(searchManager.getSearchableInfo(
    getActivity().getComponentName()));
```

```

<meta-data android:name="android.app.default_searchable"
           android:value="com.example.audiolibros.MainActivity" />
<activity
    android:name="com.example.audiolibros.MainActivity"
    android:label="@string/app_name"
    android:theme="@style/AppTheme.NoActionBar"

    android:launchMode="singleTop">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.SEARCH" />
    </intent-filter>
    <meta-data
        android:name="android.app.searchable"
        android:resource="@xml/searchable" />
</activity>

```

```

<searchable xmlns:android="http://schemas.android.com/apk/res/android"
    android:hint="Buscar Libro"
    android:label="@string/app_name">
</searchable>

```

```
android:launchMode="singleTop"
```

```

protected void onNewIntent(Intent intent) {
    if (intent.getAction() != null) {
        if (intent.getAction().equals(Intent.ACTION_SEARCH)) {
            adaptador.setBusqueda(intent.getStringExtra(SearchManager.QUERY));
            adaptador.notifyDataSetChanged();
        }
    }
}

```

### 1.7.1. Añadiendo preferencias de usuario mediante *PreferenceFragmentCompat*

```
import android.support.v4.app.Fragment;
```

```
import android.support.v4.app.FragmentTransaction;
```

```
getSupportFragmentManager()
```

```
dependencies {  
    ...  
    compile 'com.android.support:preference-v14:23.1.1'  
}
```

```
public class PreferenciasFragment extends PreferenceFragmentCompat {  
  
    @Override  
    public void onCreatePreferences(Bundle bundle, String s) {  
        addPreferencesFromResource(R.xml.preferences);  
    }  
}
```

```
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">  
    <SwitchPreferenceCompat  
        android:key="pref_autoreproducir"  
        android:title="Reproducir automáticamente"  
        android:summary="El audio se inicia automáticamente al abrirlo"  
        android:defaultValue="true"/>  
    <CheckBoxPreference  
        android:dependency="pref_autoreproducir"  
        android:layout="?android:attr/preferenceLayoutChild"  
        android:title="Otra preferencia"  
        android:summary="Ejemplo de CheckBox con dependencia" />  
    <PreferenceScreen  
        android:title="Información adicional"  
        android:summary="visualizamos página web">  
        <intent  
            android:action="android.intent.action.VIEW"  
            android:data="http://www.androidcurso.com" />  
        </PreferenceScreen>  
    </PreferenceScreen>
```

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
    ...  
    <item name="preferenceTheme">@style/PreferenceThemeOverlay</item>  
</style>
```

```
public class PreferenciasActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        getSupportFragmentManager().beginTransaction().replace(android.R.id.  
            content, new PreferenciasFragment()).commit();  
    }  
}
```

```
<activity android:name="PreferenciasActivity"/>
```

```
if (id == R.id.menu_preferencias) {
    Toast.makeText(this, "Preferencias", Toast.LENGTH_LONG).show();
    Intent i = new Intent(this, PreferenciasActivity.class);
    startActivity(i);
    return true;
}
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
    <fragment
        =>
        <LinearLayout
            android:id="@+id/contenedor_izquierdo"
            android:orientation="vertical"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="1" />
    <fragment
        android:id="@+id/detalle_fragment"
        android:name="com.example.audiolibros.fragments.DetalleFragment"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1" />
</LinearLayout>
```

```
int idContenedor = (findViewById(R.id.contenedor_pequeno) != null) ?
    R.id.contenedor_pequeno : R.id.contenedor_izquierdo;
SelectorFragment primerFragment = new SelectorFragment();
getSupportFragmentManager().beginTransaction()
    .add(idContenedor, primerFragment)
    .commit();
```

```
public void abrePreferencias() {
    int idContenedor = (findViewById(R.id.contenedor_pequeno) != null) ?
        R.id.contenedor_pequeno : R.id.contenedor_izquierdo;
    PreferenciasFragment prefFragment = new PreferenciasFragment();
    getSupportFragmentManager().beginTransaction()
        .replace(idContenedor, prefFragment)
        .addToBackStack(null)
        .commit();
}
```

```
if (id == R.id.menu_preferencias) {
    abrePreferencias();
}
```

```
    return true;
}
```

```
SharedPreferences preferencias = PreferenceManager
    .getDefaultSharedPreferences(getActivity());
if (preferencias.getBoolean("pref_autoreproducir", true)) {
    mediaPlayer.start();
}
```

## 1.8. Animaciones

### 1.8.1. Animaciones de vistas: transiciones entre actividades

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/hello_world" />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_below="@+id/textView1"
        android:onClick="sepulsa"
        android:text="Lanzar actividad" />
</RelativeLayout>
```

```
setContentView(R.layout.activity_segunda);
```

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="800"
    android:fromXDelta="-100%"
    android:toXDelta="0" />
```

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="800"
    android:fromXDelta="0"
```



```
android:toXDelta="100%" />
```

```
public void sepulsa(View view){
    Intent i = new Intent(this, SegundaActivity.class);
    startActivity(i);
    overridePendingTransition(R.anim.entrada_derecha,R.anim.salida_izquierda);
}
```

```
public void sepulsa(View view){
    Intent i = new Intent(this, SegundaActivity.class);
    ActivityOptions opts = ActivityOptions.makeCustomAnimation(
        this, R.anim.entrada_derecha, R.anim.salida_izquierda);
    startActivity(i, opts.toBundle());
}
```

### 1.8.1.1. Aplicando animaciones de vistas en Audiolibros

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="true">
    <scale android:interpolator="@android:anim/decelerate_interpolator"
        android:fromXScale="1.0" android:toXScale="0.0"
        android:fromYScale="1.0" android:toYScale="0.0"
        android:pivotX="50%" android:pivotY="50%"
        android:duration="500"/>
</set>
```

```
case 1: //Borrar
    Snackbar.make(v, "¿Estás seguro?", Snackbar.LENGTH_LONG)
        .setAction("SI", new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Animation anim = AnimationUtils.loadAnimation(actividad,
                                                            R.anim.menguar);
                anim.setAnimationListener(SelectorFragment.this);
                v.startAnimation(anim);
                adaptador.borrar(id);
                //adaptador.notifyDataSetChanged();
            }
        })
        .show();
    break;
```

```
//anim.setAnimationListener(SelectorFragment.this);
```

```
public class SelectorFragment extends Fragment
    implements Animation.AnimationListener {
```

## 1.8.2. Animaciones de propiedades

### 1.8.2.1. El motor básico de animación: *ValueAnimator*

```
<TextView
    android:id="@+id/text_view"
    ... />
```

```
public class MainActivity extends Activity implements
    ValueAnimator.AnimatorUpdateListener {
    private TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView = (TextView) findViewById(R.id.text_view);
        ValueAnimator animacion = ValueAnimator.ofFloat(10, 40);
        animacion.setDuration(1000);
        animacion.setInterpolator(new DecelerateInterpolator());
        animacion.setRepeatCount(4);
        animacion.setRepeatMode(ValueAnimator.REVERSE);
        animacion.addUpdateListener(this);
        animacion.start();
    }

    @Override
    public void onAnimationUpdate(ValueAnimator animacion) {
        float value = ((Float) (animacion.getAnimatedValue())).floatValue();
        textView.setTextSize(value);
    }
}
```

### 1.8.2.2. Automatizando las animaciones: *ObjectAnimator*

```
ValueAnimator animacion = ValueAnimator.ofFloat(10,40);
```

```
ObjectAnimator animacion =
    ObjectAnimator.ofFloat(textView, "textSize", 10, 40);
```

### 1.8.2.3. Definiendo animaciones en XML

```
<animator
    android:duration="int"
    android:valueFrom="float | int | color"
    android:valueTo="float | int | color"
    android:startOffset="int"
    android:interpolator="tipo de interpolación"
    android:repeatCount="int"
```

```

        android:repeatMode=["repeat" | "reverse"]
        android:valueType=["intType" | "floatType"]/>
<objectAnimator
    android:propertyName="string"
    ... />

```

```

<set android:ordering=["together" | "sequentially"]
    android:interpolator="tipo de interpolación">

    <animator ... />

    <objectAnimator ... />

    <set>
        ...
    </set>

</set>

```

```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:ordering="sequentially" >
    <set>
        <objectAnimator
            android:duration="1000"
            android:propertyName="textSize"
            android:valueTo="50"
            android:valueType="floatType" />
        <objectAnimator
            android:duration="2000"
            android:propertyName="textScaleX"
            android:valueFrom="0.2"
            android:valueTo="1.5" />
    </set>
    <objectAnimator
        android:duration="2000"
        android:propertyName="textColor"
        android:repeatCount="infinite"
        android:repeatMode="reverse"
        android:valueFrom="#000000"
        android:valueTo="#0000FF" />
    </set>

```

```

<TextView
    android:id="@+id/text_view"
    ... />

```

```

TextView textView = (TextView) findViewById(R.id.text_view);

```

```
AnimatorSet set = (AnimatorSet) AnimatorInflater.loadAnimator(this,
    R.animator.animacion);
set.setTarget(textView);
set.start();
```

#### 1.8.2.4. Nuevas propiedades de la clase View

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:ordering="sequentially">
    <set>
        <objectAnimator
            android:propertyName="x"
            android:duration="800"
            android:valueTo="500"
            android:valueType="floatType"/>
        <objectAnimator
            android:propertyName="y"
            android:duration="800"
            android:valueTo="400"
            android:valueType="floatType"/>
    </set>
    <objectAnimator
        android:propertyName="alpha"
        android:duration="800"
        android:repeatCount="infinite"
        android:repeatMode="reverse"
        android:valueFrom="1"
        android:valueTo="0.2"/>
</set>
```

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <objectAnimator
        android:propertyName="pivotY"
        android:valueTo="0"/>
    <objectAnimator
        android:propertyName="rotationX"
        android:duration="2000"
        .../>
</set>
```

#### 1.8.2.5. Aplicando animaciones de propiedades en Audiolibros

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android" >
    <objectAnimator
        android:propertyName="scaleX"
        android:duration="500"
        android:valueTo="0"
        android:valueType="floatType"/>
```

```

<objectAnimator
    android:propertyName="scaleY"
    android:duration="500"
    android:valueTo="0"
    android:valueType="floatType"/>
</set>

```

```

case 1: //Borrar
    Snackbar.make(v, "¿Estás seguro?", Snackbar.LENGTH_LONG)
        .setAction("SI", new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Animator anim = AnimatorInflater.loadAnimator(actividad,
                                                                R.animator.menguar);
                anim.addListener(SelectorFragment.this);
                anim.setTarget(v);
                anim.start();
                adaptador.borrar(id);
                //adaptador.notifyDataSetChanged();
            }
        })
        .show();
    break;

```

```

public class SelectorFragment extends Fragment
    implements Animator.AnimatorListener {

```

```

@Override public void onAnimationEnd(Animator animation) {
    adaptador.notifyDataSetChanged();
}

```

```

@Override
public void onBindViewHolder(ViewHolder holder, int posicion) {
    Libro libro = vectorLibros.elementAt(posicion);
    holder.portada.setImageResource(libro.recursoImagen);
    holder.titulo.setText(libro.titulo);
    holder.itemView.setScaleX(1);
    holder.itemView.setScaleY(1);
}

```

### 1.8.3. Animaciones en RecyclerView

```

case 2: //Insertar
    int posicion = recyclerView.getChildLayoutPosition(v);
    adaptador.insertar((Libro) adaptador.getItem(posicion));
    //adaptador.notifyDataSetChanged();
    adaptador.notifyItemInserted(0);

```

```
public void insertar(Libro libro){
    vectorSinFiltro.add(0, libro);
    recalculaFiltro();
}
```

```
DefaultItemAnimator animator = new DefaultItemAnimator();
animator.setAddDuration(2000);
animator.setMoveDuration(2000);
recyclerView.setItemAnimator(animator);
```

## 1.9. Otros aspectos introducidos en la versión 5.0

### 1.9.1. Extraer paleta de colores desde imágenes

```
dependencies {
    ...
    compile 'com.android.support:palette-v7:23.1.1'
}
```

```
//Extraemos el color principal de un bitmap
Bitmap bitmap = BitmapFactory.decodeResource(contexto.getResources(),
    libro.recursoImagen);
Palette palette = Palette.from(bitmap).generate();
holder.itemView.setBackgroundColor(palette.getLightMutedColor(0));
holder.titulo.setBackgroundColor(palette.getLightVibrantColor(0));
```

```
public int colorVibrante = -1, colorApagado = -1;
```

### 1.9.2. Uso de gráficos vectoriales

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="60dp"
    android:height="60dp"
    android:viewportHeight="60"
    android:viewportWidth="60">
    <path
        android:fillColor="@color/colorPrimary"
        android:strokeColor="#000000"
        android:strokeWidth="2"
        android:pathData="M0 20 L20 20 L30 0 L40 20 L60 20 L45 35 L50 60
L30 45 L10 60 L15 35 Z" />
</vector>
```

```
<svg width="640" height="480" xmlns="http://www.w3.org/2000/svg" ...
<defs>
```

```

<svg id="svg_2" height="137" width="137">
  <title>Cool smiley</title>
  <path id="svg_1" fill="url(#svg_4)"
        stroke-width="3" stroke="#000000"
        d="m32.18682,97.71674q36.3159,24.94076 72.54585,0m-64.6..."/>
  <path id="svg_5" fill="#000000" stroke-width="3"
        d="m23.84005,41.03445l17.57052,015.42937,-19.6791415.42..."/>
  <path id="svg_6" fill="#000000" stroke-width="3"
        d="m65.84005,41.03445l17.57052,015.42937,-19.6791415.42..."/>
  <defs>
    <linearGradient id="svg_4" x1="0.44531" y1="0.94922"
    ...

```

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
  android:width="60dp"
  android:height="60dp"
  android:viewportHeight="137"
  android:viewportWidth="137">
  <path android:fillColor="@color/colorPrimary"
        android:strokeColor="#000000"
        android:strokeWidth="3"
        android:pathData="m32.18682,97.71674q36.3159,24.94076 72.545...
  <path android:fillColor="#000000"
        android:strokeWidth="3"
        android:pathData="m23.84005,41.03445l17.57052,015.42937,-19.6...
  <path android:fillColor="#000000"
        android:strokeWidth="3"
        android:pathData="m65.84005,41.03445l17.57052,015.429,-19.2...
</vector>

```





# CAPÍTULO 2.

## Diseño personalizado de vistas

Por JESÚS TOMÁS

### 2.1. Algunos conceptos básicos

```
public class VistaJuego extends View {  
    @Override void onSizeChanged(); //Nos informan de nuestro tamaño  
    @Override void onDraw(); //Dibujamos la vista  
    @Override boolean onTouchEvent(); //Se ha pulsado la pantalla  
}
```

### 2.2. Una vista como la composición de varias vistas

```
<?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="horizontal" >  
    <LinearLayout  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_weight="70"  
        android:orientation="vertical" >  
        <TextView  
            android:id="@+id/textView1"  
            android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:text="ip" />
    <EditText
        android:id="@+id/ip"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:digits="0123456789." />
    <TextView
        android:id="@+id/estado"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"
        android:text="Desconectado"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="30"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="puerto" />
    <EditText
        android:id="@+id/puerto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="number" />
    <Button
        android:id="@+id/conectar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Conectar" />
</LinearLayout>
</LinearLayout>

```

```

public class VistaConectar extends LinearLayout {
    private EditText ip;
    private EditText puerto;
    private TextView estado;
    private Button conectar;

    public VistaConectar(Context context, AttributeSet attrs) {
        super(context, attrs);
        LayoutInflater.from(context).inflate(R.layout.conectar, this, true);
        ip = (EditText) findViewById(R.id.ip);
        puerto = (EditText) findViewById(R.id.puerto);
        estado = (TextView) findViewById(R.id.estado);
    }
}

```

```

        conectar = (Button) findViewById(R.id.conectar);
    }
}

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity" >
    <com.example.vistaconectar.VistaConectar
        android:id="@+id/vistaConectar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>

```

### 2.2.1. Creación de escuchadores de eventos

```

package com.example.vistaconectar;

public interface OnConectarListener {
    void onConectar(String ip, int puerto);
    void onConectado(String ip, int puerto);
    void onDesconectado();
    void onError(String mensaje);
}

```

```

private OnConectarListener escuchador;

```

```

public void setOnConectarListener(OnConectarListener escuchador) {
    this.escuchador = escuchador;
}

```

```

conectar.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        int nPuerto;
        try {
            nPuerto = Integer.parseInt(puerto.getText().toString());
        } catch (Exception e) {
            if (escuchador != null) {
                escuchador.onError("El puerto ha de ser un valor numérico");
            }
            estado.setText("ERROR");
            return;
        }
        if (nPuerto < 0 || nPuerto > 65535) {

```

```

        if (escuchador != null) {
            escuchador.onError("El puerto ha de un entero menor de 65536");
        }
        estado.setText("ERROR");
    } else {
        if (escuchador != null) {
            escuchador.onConectar(ip.getText().toString(), nPuerto);
        }
        estado.setText("Conectando ...");
    }
    // Conectar el socket ...
}
});

```

```

public class MainActivity extends Activity implements OnConectarListener {

```

```

    VistaConectar conectar = (VistaConectar) findViewById(R.id.vistaConectar);
    conectar.setOnConectarListener(this);

```

```

@Override public void onConectar(String ip, int puerto) {
    Toast.makeText(getApplicationContext(), "Conectando " + ip + ":" +
        puerto, Toast.LENGTH_SHORT).show();
}

@Override public void onConectado(String ip, int puerto) {
    // TODO Auto-generated method stub
}

@Override public void onDesconectado() {
    // TODO Auto-generated method stub
}

@Override public void onError(String message) {
    Toast.makeText(getApplicationContext(), message,
        Toast.LENGTH_SHORT).show();
}

```

## 2.3. Modificación de vistas existentes

```

public class EditTextTuneado extends EditText {
    private Paint pincel;
    public EditTextTuneado(Context context, AttributeSet attrs) {
        super(context, attrs);
        pincel = new Paint();
        pincel.setColor(Color.BLACK);
        pincel.setTextAlign(Paint.Align.RIGHT);
        pincel.setTextSize(28);
    }
}

```

```

    }

    @Override
    protected void onDraw(Canvas canvas) {
        Rect rect = new Rect();
        for (int linea = 0; linea < getLineCount(); linea++) {
            int lineaBase = getLineBounds(linea, rect);
            canvas.drawLine(rect.left, lineaBase + 2, rect.right,
                           lineaBase + 2, pincel);
            canvas.drawText("'" + (linea + 1), getWidth() - 2,
                           lineaBase, pincel);
        }
        super.onDraw(canvas);
    }
}

```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
    <com.example.edittexttuneado.EditTextTuneado
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="curso de Android\nSegunda línea\nTercera línea"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:typeface="monospace" />
</RelativeLayout>

```

### 2.3.1. Algo más de información sobre TextView

```

private Paint pincel2= new Paint();
private Path path = new Path();
private Vector<String> resaltar = new Vector<String>();

```

```

pincel2.setColor(Color.YELLOW);
pincel2.setStyle(Style.FILL);
resaltar.add("Android");
resaltar.add("curso");

```

```

final Layout layout = getLayout();
final String texto = getText().toString();
for (String palabra : resaltar) {
    int pos = 0;
    do {
        pos = texto.indexOf(palabra, pos);
        if (pos != -1) {
            pos++;
            layout.getSelectionPath(pos, pos + palabra.length(), path);
        }
    } while (pos != -1);
}

```

```

        canvas.drawPath(path, pincel2);
    }
} while (pos != -1);
}

```

```

@Override
public boolean onTouchEvent(MotionEvent evento) {
    final Layout layout = getLayout();
    final String texto = getText().toString();
    int linea = layout.getLineForVertical((int) evento.getY());
    int offset = layout.getOffsetForHorizontal(linea, evento.getX())-1;
    String s = sacaPalabra(texto, offset);
    if (s.length() != 0 && resaltar.indexOf(s) == -1) {
        resaltar.add(s);
        invalidate();
        return true;
    } else {
        return super.onTouchEvent(evento);
    }
}

```

```

String sacaPalabra(String texto, int pos) {
    int ini = pos;
    while (ini>0 && texto.charAt(ini)!=' ' && texto.charAt(ini)!='\n') {
        ini--;
    }
    int fin = pos;
    while (fin < texto.length() && texto.charAt(fin) != ' '
        && texto.charAt(fin) != '\n') {
        fin++;
    }
    return texto.substring(ini, fin).trim();
}

```

## 2.4. Creación de nuevos atributos XML

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="EditTextTuneado">
        <attr name="dibujarRayas" format="boolean" />
        <attr name="posicionNumeros" format="enum">
            <enum name="derecha" value="0"/>
            <enum name="izquierda" value="1"/>
            <enum name="sin_numeros" value="2"/>
        </attr>
        <attr name="colorNumeros" format="color" />
        <attr name="tamanyoNumeros" format="dimension" />
    </declare-styleable>
</resources>

```

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:nuevos="http://schemas.android.com/apk/res/com.example.
    edittexttuneado"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <com.example.edittexttuneado.EditTextTuneado
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="curso de Android\nSegunda línea\nTercera línea"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:typeface="monospace"
        nuevos:dibujarRayas="true"
        nuevos:posicionNumeros="izquierda"
        nuevos:tamanyoNumeros="16dp"
        nuevos:colorNumeros="#FF0000" />
</RelativeLayout>

```

```

TypedArray a = context.getTheme().obtainStyledAttributes(attrs,
    R.styleable.EditTextTuneado, 0, 0);
try {
    dibujarRayas = a.getBoolean(
        R.styleable.EditTextTuneado_dibujarRayas, true);
    posicionNumeros = a.getInteger(
        R.styleable.EditTextTuneado_posicionNumeros, 0);
    switch (posicionNumeros) {
        case 0:
            pincel.setTextAlign(Paint.Align.RIGHT);
            break;
        case 1:
            pincel.setTextAlign(Paint.Align.LEFT);
            break;
    }
    int colorNumeros = a.getColor(
        R.styleable.EditTextTuneado_colorNumeros, Color.BLACK);
    pincel.setColor(colorNumeros);
    float tamanyoNumeros = a.getDimension(
        R.styleable.EditTextTuneado_tamanyoNumeros, 12*densidad);
    pincel.setTextSize(tamanyoNumeros);
} finally {
    a.recycle();
}

```

```

private boolean dibujarRayas;
private int posicionNumeros;

```

```

@Override
protected void onDraw(Canvas canvas) {
    ...
    for (int linea = 0; linea < getLineCount(); linea++) {
        int lineaBase = getLineBounds(linea, rect);
        if (dibujarRayas) {
            canvas.drawLine(rect.left, lineaBase+2, rect.right,
                           lineaBase+2, pincel);
        }
        switch (posicionNumeros) {
            case 0:
                canvas.drawText("" + (linea+1), getWidth()-2, lineaBase, pincel);
                break;
            case 1:
                canvas.drawText("" + (linea+1), 2, lineaBase, pincel);
                break;
        }
    }
    super.onDraw(canvas);
}

```

## 2.5. Una vista creada desde cero

### 2.5.1. Diseño y dibujo de la vista

```

Application Name: ZoomSeekBar
Package name: com.example.zoomseekbar
Minimum Required SDK: API 4 (1.6)
Target SDK: API 17 (4.2)
Compile With: API 17 (4.2)

```

```

public class ZoomSeekBar extends View {
    // Valor a controlar
    private int val = 160; // valor seleccionado
    private int valMin = 100; // valor mínimo
    private int valMax = 200; // valor máximo
    private int escalaMin = 150; // valor mínimo visualizado
    private int escalaMax = 180; // valor máximo visualizado
    private int escalaIni = 100; // origen de la escala
    private int escalaRaya = 2; // cada cuantas unidades una rayas
    private int escalaRayaLarga = 5; // cada cuantas rayas una larga
    // Dimensiones en pixels
    private int altoNumeros;
    private int altoRegla;
    private int altoBar;
    private int altoPalanca;
    private int anchoPalanca;
    private int altoGuia;
    // Valores que indican donde dibujar
    private int xIni;
}

```



```

private int yIni;
private int ancho;
// Objetos Rect con diferentes regiones
private Rect escalaRect = new Rect();
private Rect barRect = new Rect();
private Rect guiaRect = new Rect();
private Rect palancaRect = new Rect();
// Objetos Paint globales para no tener que crearlos cada vez
private Paint textoPaint = new Paint();
private Paint reglaPaint = new Paint();
private Paint guiaPaint = new Paint();
private Paint palancaPaint = new Paint();

```

```

public ZoomSeekBar(Context context, AttributeSet attrs) {
    super(context, attrs);
    float dp = getResources().getDisplayMetrics().density;
    TypedArray a = context.getTheme().obtainStyledAttributes(attrs,
        R.styleable.ZoomSeekBar, 0, 0);

    try {
        altoNumeros = a.getDimensionPixelSize(
            R.styleable.ZoomSeekBar_altoNumeros, (int) (30 * dp));
        altoRegla = a.getDimensionPixelSize(
            R.styleable.ZoomSeekBar_altoRegla, (int) (20 * dp));
        altoBar = a.getDimensionPixelSize(
            R.styleable.ZoomSeekBar_altoBar, (int) (70 * dp));
        altoPalanca = a.getDimensionPixelSize(
            R.styleable.ZoomSeekBar_altoPalanca, (int) (40 * dp));
        altoGuia = a.getDimensionPixelSize(
            R.styleable.ZoomSeekBar_altoGuia, (int) (10 * dp));
        anchoPalanca = a.getDimensionPixelSize(
            R.styleable.ZoomSeekBar_anchoPalanca, (int) (20 * dp));
        textoPaint.setTextSize(a.getDimension(
            R.styleable.ZoomSeekBar_altoTexto, 16 * dp));
        textoPaint.setColor(a.getColor(
            R.styleable.ZoomSeekBar_colorTexto, Color.BLACK));
        reglaPaint.setColor(a.getColor(
            R.styleable.ZoomSeekBar_colorRegla, Color.BLACK));
        guiaPaint.setColor(a.getColor(
            R.styleable.ZoomSeekBar_colorGuia, Color.BLUE));
        palancaPaint.setColor(a.getColor(
            R.styleable.ZoomSeekBar_colorPalanca, 0xFF00007F));
    } finally {
        a.recycle();
    }
    textoPaint.setAntiAlias(true);
    textoPaint.setTextAlign(Paint.Align.CENTER);
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="ZoomSeekBar">

```

```

<attr name="altoNumeros" format="dimension" />
<attr name="altoRegla" format="dimension" />
<attr name="altoBar" format="dimension" />
<attr name="altoPalanca" format="dimension" />
<attr name="altoGuia" format="dimension" />
<attr name="anchoPalanca" format="dimension" />
<attr name="altoTexto" format="dimension" />
<attr name="colorTexto" format="color" />
<attr name="colorRegla" format="color" />
<attr name="colorPalanca" format="color" />
<attr name="colorGuia" format="color" />
</declare-styleable>
</resources>

```

```

@Override
protected void onSizeChanged(int w, int h, int oldw, int oldh) {
    super.onSizeChanged(w, h, oldw, oldh);
    xIni = getPaddingLeft();
    yIni = getPaddingTop();
    ancho = getWidth() - getPaddingRight() - getPaddingLeft();
    barRect.set(xIni, yIni, xIni + ancho, yIni + altoBar);
    escalaRect.set(xIni, yIni + altoBar, xIni + ancho, yIni + altoBar
        + altoNumeros + altoRegla);

    int y = yIni + (altoBar - altoGuia) / 2;
    guiaRect.set(xIni, y, xIni + ancho, y + altoGuia);
}

```

```

@Override protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    // Dibujamos Barra con palanca
    canvas.drawRect(guiaRect, guiaPaint);
    int y = yIni + (altoBar - altoPalanca) / 2;
    int x = xIni + ancho * (val - escalaMin) / (escalaMax - escalaMin)
        - anchoPalanca / 2;
    palancaRect.set(x, y, x + anchoPalanca, y + altoPalanca);
    canvas.drawRect(palancaRect, palancaPaint);
    palancaRect.set(x - anchoPalanca / 2, y, x + 3 * anchoPalanca / 2, y
        + altoPalanca);

    // Dibujamos Escala
    int v = escalaIni;
    while (v <= escalaMax) {
        if (v >= escalaMin) {
            x = xIni + ancho * (v - escalaMin) / (escalaMax - escalaMin);
            if (((v - escalaIni) / escalaRaya) % escalaRayaLarga == 0) {
                y = yIni + altoBar + altoRegla;
                canvas.drawText(Integer.toString(v), x, y + altoNumeros,
                    textoPaint);
            } else {
                y = yIni + altoBar + altoRegla * 1 / 3;
            }
            canvas.drawLine(x, yIni + altoBar, x, y, reglaPaint);
        }
        v++;
    }
}

```

```

    }
    v += escalaRaya;
}
}

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <com.example.zoomseekbar.ZoomSeekBar
        android:layout_width="match_parent"
        android:layout_height="140dp"
        android:padding="10dp"/>
    <com.example.zoomseekbar.ZoomSeekBar
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="20dp"
        app:altoBar="40dp"
        app:altoGuia="4dp"
        app:altoNumeros="15dp"
        app:altoPalanca="40dp"
        app:altoRegla="10dp"
        app:altoTexto="12dp"
        app:anchoPalanca="10dp"
        app:colorGuia="#FF9050"
        app:colorPalanca="#FF0000"
        app:colorRegla="#B0B070"
        app:colorTexto="#00A0A0" />
</LinearLayout>

```

## 2.5.2. Gestión de eventos

```

// Variables globales usadas en onTouchEvent()
enum Estado {
    SIN_PULSACION, PALANCA_PULSADA, ESCALA_PULSADA, ESCALA_PULSADA_DOBLE };
Estado estado = Estado.SIN_PULSACION;
int antVal_0, antVal_1;

@Override public boolean onTouchEvent(MotionEvent event) {
    int x_0, y_0, x_1, y_1;
    x_0 = (int) event.getX(0);
    y_0 = (int) event.getY(0);
    int val_0 = escalaMin + (x_0 - xIni) * (escalaMax - escalaMin) / ancho;
    if (event.getPointerCount() > 1) {
        x_1 = (int) event.getX(1); y_1 = (int) event.getY(1);
    } else {
        x_1 = x_0; y_1 = y_0;
    }
    int val_1 = escalaMin + (x_1 - xIni) * (escalaMax - escalaMin) / ancho;
    switch (event.getAction() & MotionEvent.ACTION_MASK) {

```

```

case MotionEvent.ACTION_DOWN:
    if (palancaRect.contains(x_0, y_0)) {
        estado = Estado.PALANCA_PULSADA;
    } else if (barRect.contains(x_0, y_0)) {
        if (val_0 > val) val++; else val--;
        invalidate(barRect);
    } else if (escalaRect.contains(x_0, y_0)) {
        estado = Estado.ESCALA_PULSADA;
        antVal_0 = val_0;
    }
    break;
case MotionEvent.ACTION_POINTER_DOWN:
    if (estado == Estado.ESCALA_PULSADA) {
        if (escalaRect.contains(x_1, y_1)) {
            antVal_1 = val_1;
            estado = Estado.ESCALA_PULSADA_DOBLE;
        }
    }
    break;
case MotionEvent.ACTION_UP:
    estado = Estado.SIN_PULSACION;
    break;
case MotionEvent.ACTION_POINTER_UP:
    if (estado == Estado.ESCALA_PULSADA_DOBLE) {
        estado = Estado.ESCALA_PULSADA;
    }
    break;
case MotionEvent.ACTION_MOVE:
    if (estado == Estado.PALANCA_PULSADA) {
        val = ponDentroRango(val_0, escalaMin, escalaMax);
        invalidate(barRect);
    }
    if (estado == Estado.ESCALA_PULSADA_DOBLE) {
        escalaMin = antVal_0 + (xIni-x_0) * (antVal_0-antVal_1) / (x_0-x_1);
        escalaMin = ponDentroRango(escalaMin, valMin, val);
        escalaMax = antVal_0 + (ancho+xIni-x_0)
            * (antVal_0-antVal_1) / (x_0-x_1);
        escalaMax = ponDentroRango(escalaMax, val, valMax);
        invalidate();
    }
    break;
}
return true;
}

int ponDentroRango(int val, int valMin, int valMax) {
    if (val < valMin) {
        return valMin;
    } else if (val > valMax) {
        return valMax;
    } else {
        return val;
    }
}

```

```
}
```

### 2.5.3. Cómo Android dibuja las vistas y obtiene sus tamaños

```
<com.example.zoomseekbar.ZoomSeekBar
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp" />
```

```
@Override
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    int altoDeseado = altoNumeros + altoRegla + altoBar
                      + getPaddingBottom() + getPaddingTop();
    int alto = obtenDimension(heightMeasureSpec, altoDeseado);
    int anchoDeseado = 2 * altoDeseado;
    int ancho = obtenDimension(widthMeasureSpec, anchoDeseado);
    setMeasuredDimension(ancho, alto);
}

private int obtenDimension(int measureSpec, int deseado) {
    int dimension = MeasureSpec.getSize(measureSpec);
    int modo = MeasureSpec.getMode(measureSpec);
    if (modo == MeasureSpec.EXACTLY) {
        return dimension;
    } else if (modo == MeasureSpec.AT_MOST) {
        return Math.min(dimension, deseado);
    } else {
        return deseado;
    }
}
```

### 2.5.4. Interactuando con otros objetos

```
public int getVal() {
    return val;
}

public void setVal(int val) {
    if (valMin <= val && val <= valMax) {
        this.val = val;
        escalaMin = Math.min(escalaMin, val);
        escalaMax = Math.max(escalaMax, val);
        invalidate();
    }
}
```

## 2.6. Creación de *widgets* de escritorio

### 2.6.1. Pasos a seguir para crear un *widget*

#### 2.6.1.1. Definir las características del *widget*

```
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="250dp"
    android:minHeight="40dp"
    android:minResizeWidth="40dp"
    android:minResizeHeight="40dp"
    android:updatePeriodMillis="86400000"
    android:previewImage="@drawable/preview"
    android:initialLayout="@layout/mi_appwidget"
    android:configure="com.example.android.MiActividadDeConfiguracion"
    android:resizeMode="horizontal|vertical"
    android:widgetCategory="home_screen|keyguard"
    android:initialKeyguardLayout="@layout/mi_keyguard">
</appwidget-provider>
```

#### 2.6.1.2. Declarar el *widget* en *AndroidManifest*

```
<receiver android:name="MiAppWidgetProvider" >
    <intent-filter>
        <action
            android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>
    <meta-data android:name="android.appwidget.provider"
        android:resource="@xml/mi_appwidget_info" />
</receiver>
```

#### 2.6.1.3. Crear una actividad para configurarlo

```
<activity android:name=".MiActividadDeConfiguracion">
    <intent-filter>
        <action
            android:name="android.appwidget.action.APPWIDGET_CONFIGURE"/>
    </intent-filter>
</activity>
```

### 2.6.2. Creación de un *widget* de escritorio sencillo

```
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:minHeight="110dp"
    android:minWidth="110dp"
    android:minResizeWidth="40dp"
    android:minResizeHeight="40dp"
    android:updatePeriodMillis="1800000"
```

```

    android:initialLayout="@Layout/widget"
    android:previewImage="@drawable/preview"
    android:resizeMode="horizontal|vertical"
    android:widgetCategory="home_screen" >
</appwidget-provider>

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Widget Reloj"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    <AnalogClock
        android:id="@+id/analogClock1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>

```

```

public class MiAppWidgetProvider extends AppWidgetProvider{
}

```

```

<receiver android:name="MiAppWidgetProvider" >
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/widget_info" />
</receiver>

```

### 2.6.3. Actualizando el *widget* de escritorio

```

public void onUpdate(Context context, AppWidgetManager appWidgetManager,
    int[] widgetIds) {
    for (int widgetId: widgetIds) {
        actualizaWidget(context, widgetId);
    }
}

```

```

public static void actualizaWidget(Context context, int widgetId) {
    RemoteViews remoteViews = new RemoteViews(context.getPackageName(),
        R.layout.widget);
    remoteViews.setTextViewText(R.id.textView1, "Un texto");
}

```

```
AppWidgetManager.getInstance(context).updateAppWidget(widgetId,  
remoteViews);  
}
```

```
public static void actualizaWidget(Context context, int widgetId) {  
    int cont = incrementaContador(context, widgetId);  
    RemoteViews remoteViews = new RemoteViews(context.getPackageName(),  
R.layout.widget);  
    remoteViews.setTextViewText(R.id.textView1, "Contador: " + cont);  
    AppWidgetManager.getInstance(context).updateAppWidget(widgetId,  
remoteViews);  
}
```

```
private static int incrementaContador(Context context, int widgetId) {  
    SharedPreferences prefs = context.getSharedPreferences("contadores",  
Context.MODE_PRIVATE);  
  
    int cont = prefs.getInt("cont_" + widgetId, 0);  
    cont++;  
    SharedPreferences.Editor editor = prefs.edit();  
    editor.putInt("cont_" + widgetId, cont);  
    editor.commit();  
    return cont;  
}
```

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>  
<map>  
    <int name="cont_11" value="1" />  
    <int name="cont_12" value="1" />  
</map>
```

## 2.6.4. Actuando ante el evento onClick

```
Intent intent = new Intent(context, MainActivity.class);  
PendingIntent pendingIntent = PendingIntent.getActivity(context, 0,  
intent, 0);  
remoteViews.setOnClickPendingIntent(R.id.analogClock1, pendingIntent);
```

```
public static final String ACCION_INCR =  
"com.example.widgetescritorio.ACCION_INCR";  
public static final String EXTRA_PARAM =  
"com.example.widgetescritorio.EXTRA_ID";
```

```
intent = new Intent(context, MiAppWidgetProvider.class);  
intent.setAction(ACCION_INCR);  
intent.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, widgetId);  
intent.putExtra(EXTRA_PARAM, "otro parámetro");  
intent.setData(Uri.parse(intent.toUri(Intent.URI_INTENT_SCHEME)));
```



```
pendingIntent = PendingIntent.getBroadcast(context, 0, intent,
                                           PendingIntent.FLAG_UPDATE_CURRENT);
remoteViews.setOnClickPendingIntent(R.id.textView1, pendingIntent);
```

```
@Override
public void onReceive(Context context, Intent intent) {
    AppWidgetManager mgr = AppWidgetManager.getInstance(context);
    if (intent.getAction().equals(ACCION_INCR)) {
        int widgetId = intent.getIntExtra(
            AppWidgetManager.EXTRA_APPWIDGET_ID,
            AppWidgetManager.INVALID_APPWIDGET_ID);
        String param = intent.getStringExtra(EXTRA_PARAM);
        Toast.makeText(context, "Parámetro:" + param,
            Toast.LENGTH_SHORT).show();
        actualizarWidget(context, widgetId);
    }
    super.onReceive(context, intent);
}
```

## 2.6.5. Añadiendo una actividad de configuración

```
<activity android:name=".ConfiguraWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_CONFIGURE"/>
    </intent-filter>
</activity>
```

```
android:configure="com.example.widgetescritorio.ConfiguraWidget"
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Valor inicial del contador:"
        android:textAppearance="?android:attr/textAppearanceMedium" />
    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" >
        <requestFocus />
    </EditText>
    <Button
        android:id="@+id/buttonOK"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="buttonOK"
        android:text="@android:string/ok" />
</LinearLayout>
```

```
public class ConfiguraWidget extends Activity {
    int widgetId;
    EditText editText;

    @Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.configura_widget);
        editText = (EditText) findViewById(R.id.editText1);
        setResult(RESULT_CANCELED);
        Bundle extras = getIntent().getExtras();
        if (extras == null) {
            finish();
        }
        widgetId = extras.getInt(AppWidgetManager.EXTRA_APPWIDGET_ID,
                                AppWidgetManager.INVALID_APPWIDGET_ID);
        if (widgetId == AppWidgetManager.INVALID_APPWIDGET_ID) {
            finish();
        }
    }
}
```

```
public void buttonOK(View view) {
    int cont;
    try {
        cont = Integer.parseInt(editText.getText().toString());
    } catch (Exception e) {
        Toast.makeText(this, "No es un número", Toast.LENGTH_SHORT).show();
        return;
    }
    SharedPreferences prefs = getSharedPreferences("contadores",
                                                    Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = prefs.edit();
    editor.putInt("cont_" + widgetId, cont);
    editor.commit();
    MiAppWidgetProvider.actualizaWidget(this, widgetId);
    Intent resultValue = new Intent();
    resultValue.putExtra(AppWidgetManager.EXTRA_APPWIDGET_ID, widgetId);
    setResult(RESULT_OK, resultValue);
    finish();
}
```

# CAPÍTULO 3.

## Hilos de ejecución en la interfaz del usuario

Por CARSTEN VOGT  
Traducción de JESÚS TOMÁS

### 3.1. Programación basada en eventos y el hilo de ejecución de usuario

#### 3.1.1. Cola de eventos y bucle de eventos

```
while (true) {  
    if (la cola de eventos está vacía)  
        espera a que llegue un evento  
    evento_a_manejar = primer evento de la cola  
    switch (evento_a_manejar) {  
        case EVENTO_1: ejecuta manejador del tipo EVENTO_1  
        case EVENTO_2: ejecuta manejador del tipo EVENTO_2  
        ...  
    }  
}
```

```
public void onClick(View v) {  
    String buttonText = ((Button) v).getText().toString();  
    if (buttonText.equals("red"))  
        screenLayout.setBackgroundColor(Color.RED);  
    if (buttonText.equals("green"))  
        screenLayout.setBackgroundColor(Color.GREEN);  
    if (buttonText.equals("blue"))  
        screenLayout.setBackgroundColor(Color.BLUE);  
}
```

### 3.1.2. El hilo de la interfaz de usuario de una aplicación Android

```
public class MainActivity extends Activity {
    LinearLayout screenLayout = null;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        screenLayout = (LinearLayout) findViewById(R.id.layoutMain);
        Log.v("MYLOG", "En onCreate():" + Thread.currentThread().getId());
    }

    public void redOnClick(View v) {
        Log.v("MYLOG", "En redOnClick():" + Thread.currentThread().getId());
        screenLayout.setBackgroundColor(Color.RED);
    }

    public void greenOnClick(View v) {
        Log.v("MYLOG", "En greenOnClick():" + Thread.currentThread().getId());
        screenLayout.setBackgroundColor(Color.GREEN);
    }

    public void blueOnClick(View v) {
        Log.v("MYLOG", "En blueOnClick():" + Thread.currentThread().getId());
        screenLayout.setBackgroundColor(Color.BLUE);
    }
}
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layoutMain"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:onClick="redOnClick"
        android:text="red" />
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:onClick="greenOnClick"
        android:text="green" />
    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:onClick="blueOnClick"
        android:text="blue" />
</LinearLayout>
```

```
public void longCalculation(View v) {
```

```

Log.v("MYLOG", "En longCalculation():" +
                                Thread.currentThread().getId());
try {
    Thread.currentThread().sleep(4000);
} catch (InterruptedException e) {}
EditText resultField = (EditText) findViewById(R.id.resultField);
resultField.setText("Resultado " + resultField.getText());
}

```

```

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:onClick="LongCalculation"
    android:text="Cálculo proLongado" />
<EditText
    android:id="@+id/resultField"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />

```

## 3.2. Concurrencia en programación orientada a eventos

### 3.2.1. Hilos para el manejo de eventos

```

public void longCalculation(View v) {
    (new Thread() {
        public void run() {
            1.) código para manejar un evento largo
            2.) mostramos los resultados en la IGU
               o se los mandamos al hilo de la IU
        }
    }).start();
}

```

### 3.2.2. El problema de los hilos en segundo plano: no tienen acceso a la interfaz gráfica de usuario

```

public void longCalculation(View v) {
    (new Thread() {
        public void run() {
            Log.v("MYLOG", "En longCalculation():"
                                +Thread.currentThread().getId());

            try {
                Thread.currentThread().sleep(4000);
            } catch (InterruptedException e) {}
            EditText resultField = (EditText) findViewById(R.id.resultField);
            resultField.setText("Resultado "+resultField.getText());
        }
    }).start();
}

```

## 3.3. La clase AsyncTask

```
abstract class AsyncTask<Params, Progress, Result> {  
    protected abstract Result doInBackground(Params... param);  
    ... mas métodos ...  
}
```

### 3.3.1. Extendiendo AsyncTask

```
class MyAsyncTask extends AsyncTask<Long, Double, Boolean> {  
    protected Boolean doInBackground(Long... a) {  
        ... código para manejar un evento ...  
        ... devolver un valor booleano ...  
    }  
}
```

### 3.3.2. Secuencia de operaciones

```
MyAsyncTask mAsyncTask = new MyAsyncTask();  
mAsyncTask.execute(3093215881333057);
```

```
Boolean doInBackground(Long... n) {  
    boolean isPrime =  
        ... operación larga para ver si n es un número primo ...  
    return isPrime;  
}
```

```
void onPostExecute(Boolean isPrime) {  
    String resultStr;  
    if (isPrime) resultStr = "Is a prime number!";  
    else resultStr = "Is no prime number!";  
    EditText resultField = (EditText) findViewById(R.id.resultField);  
    resultField.setText(resultStr);  
}
```

```
private class MyAsyncTask extends AsyncTask<Long, Double, Boolean> {  
  
    protected Boolean doInBackground(Long... n) {  
        Log.v("MYLOG", "Thread "+Thread.currentThread().getId()+  
            ": doInBackground() starts");  
  
        boolean isPrime = true;  
        long nValue = n[0];  
        if (nValue%2==0)  
            isPrime = false;  
        else {  
            long factor=3;
```

```

        double limit = Math.sqrt(nValue)+0.0001;
        double progressPercentage = 0;
        while (factor<limit) {
            if (nValue%factor==0) {
                isPrime = false;
                break; }
            factor+=2;
            if (factor>limit*progressPercentage/100) {
                publishProgress(progressPercentage/100);
                progressPercentage+=5; }
        }
    }
    Log.v("MYLOG", "Thread "+Thread.currentThread().getId()+
        ": doInBackground() ends");
    return isPrime;
}

protected void onPreExecute() {
    Log.v("MYLOG", "Thread "+Thread.currentThread().getId()+
        ": onPreExecute()");
    EditText resultField = (EditText) findViewById(R.id.resultField);
    resultField.setText("");
}

protected void onProgressUpdate(Double... progress) {
    Log.v("MYLOG", "Thread "+Thread.currentThread().getId()+
        ": onProgressUpdate()");
    EditText resultField = (EditText) findViewById(R.id.resultField);
    resultField.setText((progress[0]*100)+"% completed");
}

protected void onPostExecute(Boolean isPrime) {
    Log.v("MYLOG", "Thread "+Thread.currentThread().getId()+
        ": onPostExecute()");
    EditText resultField = (EditText) findViewById(R.id.resultField);
    resultField.setText(isPrime+"");
}
}

```

```

public void triggerPrimecheck(View v) {
    Log.v("MYLOG", "Thread "+Thread.currentThread().getId()+
        ": triggerPrimecheck() starts");
    EditText inputField = (EditText) findViewById(R.id.inputField);
    long parameter = Long.parseLong(inputField.getText().toString());
    MyAsyncTask mAsyncTask = new MyAsyncTask();
    mAsyncTask.execute(parameter);
    Log.v("MYLOG", "Thread "+Thread.currentThread().getId()+
        ": triggerPrimecheck() ends");
}

```

```

<EditText
    android:id="@+id/inputField"
    android:text="3093215881333057"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/primecheckbutton"
    android:text="Is it prime?"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:onClick="triggerPrimecheck" />
<EditText
    android:id="@+id/resultField"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />

```

```

public class FindPrimeNumbersActivity extends Activity {

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void triggerFindprimes(View v) {
        EditText lowerborderField =
            (EditText) findViewById(R.id.lowerBorder);
        long lowerBorder =
            Long.parseLong(lowerborderField.getText().toString());
        EditText upperborderField =
            (EditText) findViewById(R.id.upperBorder);
        long upperBorder =
            Long.parseLong(upperborderField.getText().toString());
        MyAsyncTask mAsyncTask = new MyAsyncTask();
        mAsyncTask.execute(lowerBorder, upperBorder);
    }

    private class MyAsyncTask extends AsyncTask<Long, Long, String> {

        protected String doInBackground(Long... n) {
            for (long i=n[0]; i<=n[1]; i++) {
                boolean isPrime = true;
                if (i%2==0)
                    isPrime = false;
                else {
                    long factor=3;
                    double limit = Math.sqrt(i)+0.0001;
                    while (factor<limit) {
                        if (i%factor==0) {
                            isPrime = false;
                            break;
                        }
                        factor+=2;
                    }
                }
            }
        }
    }
}

```



```

        if (isPrime)
            publishProgress(i);
    }
    return "END";
}

protected void onPreExecute() {
    EditText resultField =
        (EditText) findViewById(R.id.resultField);
    resultField.setText("");
}

protected void onProgressUpdate(Long... primeNumber) {
    EditText resultField =
        (EditText) findViewById(R.id.resultField);
    String output = resultField.getText()+" "+primeNumber[0];
    resultField.setText(output);
}

protected void onPostExecute(String message) {
    EditText resultField =
        (EditText) findViewById(R.id.resultField);
    String output = resultField.getText()+" "+message;
    resultField.setText(output);
}
}

```

```

<?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" >
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >
        <EditText
            android:id="@+id/lowerBorder"
            android:text="100000000000"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
        <EditText
            android:id="@+id/upperBorder"
            android:text="100000000300"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
    </LinearLayout>
    <Button
        android:id="@+id/findprimesButton"
        android:text="Find primes in interval"

```

```

        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:onClick="triggerFindprimes" />
<EditText
    android:id="@+id/resultField"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
</LinearLayout>

```

## 3.4. Animaciones con SurfaceView

### 3.4.1. Estructura de la aplicación

```

class MySurfaceView extends SurfaceView implements SurfaceHolder.Callback{
    // el hilo para controlar la animation - se define más adelante
    private MyAnimationThread animThread = null;

    ... define atributos como posición y velocidad de los objetos gráficos ...

    public MySurfaceView(Context ctx, AttributeSet attrs, int defStyle, ...){
        super(ctx, attrs, defStyle);
        ... inicializa los atributos ...
        getHolder().addCallback(this); //registra como escuchador de eventos
    }
    // método para dibujar un paso de la animación
    public void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        ... dibuja en canvas, por ejemplo los objetos en sus posiciones ...
    }

    // métodos callback
    public void surfaceCreated(SurfaceHolder holder) {
        // crea y arranca el hilo de la animación (a menos que ya exista)
        if (animThread!=null) return;
        animThread = new MyAnimationThread(getHolder(),...);
        animThread.start();
    }

    public void surfaceChanged(SurfaceHolder holder,
                               int format, int width, int height) { }

    public void surfaceDestroyed(SurfaceHolder holder) {
        animThread.stop = true; // para el hilo de la animación
    }

    // definición de la clase para el hilo de la animación
    private class MyAnimationThread extends Thread {

        public boolean stop = false;
        private SurfaceHolder surfaceHolder;
    }

```

```

... define otros atributos ...

// constructor
public MyAnimationThread(SurfaceHolder surfaceHolder, ...) {
    this.surfaceHolder = surfaceHolder;
    ... inicializa los atributos ...
}

// operaciones del hilo
public void run() {
    while (!stop) {
        ... actualize los atributos,
        como nuevas posiciones de los objetos etc. ...
        Canvas c = null;
        try {
            // obtén el canvas para dibujar
            c = surfaceHolder.lockCanvas(null);
            synchronized (surfaceHolder) {
                // dibuja en el canvas, como los objetos en sus posiciones
                onDraw(c); }
        } catch (Exception e) {
        } finally {
            // muestra el canvas en la pantalla
            if (c != null) surfaceHolder.unlockCanvasAndPost(c); }
        }
    }
}

// actividad que visualiza el SurfaceView
public class MySurfaceViewActivity extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new MySurfaceView(this,null,0,...));
    }
}

```

```

public class BouncingBallActivity extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new BouncingBallView(this,null,0));
    }
}

```

```

class BouncingBallView extends SurfaceView
    implements SurfaceHolder.Callback {
    private BouncingBallAnimationThread bbThread = null;

    public BouncingBallView(Context ctx, AttributeSet attrs, int defStyle){
        super(ctx, attrs, defStyle);
        getHolder().addCallback(this);
    }
}

```

```
public void onDraw(Canvas canvas) {
    super.onDraw(canvas);
}

public void surfaceCreated(SurfaceHolder holder) {
    if (bbThread!=null) return;
    bbThread = new BouncingBallAnimationThread(getHolder());
    bbThread.start();
}

public void surfaceChanged(SurfaceHolder holder,
                           int format, int width, int height) { }

public void surfaceDestroyed(SurfaceHolder holder) {
    bbThread.stop = true;
}
}
```

```
private int xPosition = getWidth()/2;
private int yPosition = getHeight()/2;
private int xDirection = 20;
private int yDirection = 40;
private static int radius = 20;
private static int ballColor = Color.RED;
```

```
public void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    Paint paint = new Paint();
    paint.setColor(Color.BLACK);
    canvas.drawRect(0, 0, getWidth(), getHeight(), paint);
    paint.setColor(ballColor);
    canvas.drawCircle(xPosition, yPosition, radius, paint);
}
```

```
private class BouncingBallAnimationThread extends Thread {
    public boolean stop = false;
    private SurfaceHolder surfaceHolder;

    public BouncingBallAnimationThread(SurfaceHolder surfaceHolder) {
        this.surfaceHolder = surfaceHolder;
    }

    public void run() {
        while (!stop) {
            xPosition += xDirection;
            yPosition += yDirection;
            if (xPosition<0) {
                xDirection = -xDirection; xPosition = radius; }
            if (xPosition>getWidth()-radius) {
                xDirection = -xDirection; xPosition = getWidth()-radius; }
        }
    }
}
```

```
        if (yPosition<0) {
            yDirection = -yDirection; yPosition = radius; }
        if (yPosition>getHeight()-radius) {
            yDirection = -yDirection; yPosition = getHeight()-radius-1; }
        Canvas c = null;
        try {
            c = surfaceHolder.lockCanvas(null);
            synchronized (surfaceHolder) {
                onDraw(c);
            }
        } catch (Exception e) {
        } finally {
            if (c != null) surfaceHolder.unlockCanvasAndPost(c); }
    }
}
```

```
public boolean onTouchEvent(MotionEvent event) {
    if (event.getAction() != MotionEvent.ACTION_DOWN) return false;
    if (xDirection!=0 || yDirection!=0)
        xDirection = yDirection = 0;
    else {
        xDirection = (int) event.getX() - xPosition;
        yDirection = (int) event.getY() - yPosition; }
    return true;
}
```



# CAPÍTULO 4.

## Comunicación con *bluetooth*

Por MIGUEL GARCÍA PINEDA Y  
CARSTEN VOGT

### 4.1. Permisos en *bluetooth*

```
<manifest ...  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />  
<uses-permission android:name="android.permission.BLUETOOTH" />  
...  
</manifest>
```

### 4.2. Configuración *bluetooth*

```
// Adaptador local Bluetooth  
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();  
  
// Si el adaptador es null, no soporta Bluetooth  
if (mBluetoothAdapter == null) {  
    Toast.makeText(this, "Bluetooth is not available",  
        Toast.LENGTH_LONG).show();  
    finish();  
    return;  
}
```

```
// Si el adaptador no esta activo, pide habilitarlo.  
if (!mBluetoothAdapter.isEnabled()) {  
    Intent enableIntent = new In-  
tent(BluetoothAdapter.ACTION_REQUEST_ENABLE);  
    startActivityForResult(enableIntent, REQUEST_ENABLE_BT);  
}
```

```
public void onActivityResult(int requestCode, int resultCode, Intent data)
{
    if(D) Log.d(TAG, "onActivityResult " + resultCode);
    switch (requestCode) {
        case REQUEST_CONNECT_DEVICE_SECURE:
            // Cuando DeviceListActivity devuelve un dispositivo para conectar
            if (resultCode == Activity.RESULT_OK) {
                connectDevice(data, true);
            }
            break;
        case REQUEST_CONNECT_DEVICE_INSECURE:
            // Cuando DeviceListActivity devuelve un dispositivo para conectar
            if (resultCode == Activity.RESULT_OK) {
                connectDevice(data, false);
            }
            break;
        case REQUEST_ENABLE_BT:
            // Devuelve habilitar el dispositivo bluetooth
            if (resultCode == Activity.RESULT_OK) {
                // Adaptador habilitado, llamamos método
                setupChat();
            } else {
                // No podemos habilitar el adaptador
                Log.d(TAG, "BT not enabled");
                Toast.makeText(this, R.string.bt_not_enabled_leaving,
                    Toast.LENGTH_SHORT).show();
                finish();
            }
    }
}
```

Application name: Bluetooth01  
Project name: Bluetooth01  
Package name: com.bluetooth01  
Activity name: Bluetooth01  
Layout name: activity\_bluetooth01  
Resto de parámetros por defecto.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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"
    tools:context=".Bluetooth01" >

    <TextView
```



```

        android:id="@+id/out"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="" >
    </TextView>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="30dp"
        android:layout_marginTop="49dp"
        android:text="TURN_ON" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/button1"
        android:layout_below="@+id/button1"
        android:layout_marginTop="27dp"
        android:text="DISCOVERABLE" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/button2"
        android:layout_below="@+id/button2"
        android:layout_marginTop="28dp"
        android:text="TURN_OFF" />

</RelativeLayout>

```

```

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

```

```

private static final int REQUEST_ENABLE_BT = 0;
private static final int REQUEST_DISCOVERABLE_BT = 0;

```

```

final TextView out = (TextView) findViewById(R.id.out);
final Button button1 = (Button) findViewById(R.id.button1);
final Button button2 = (Button) findViewById(R.id.button2);
final Button button3 = (Button) findViewById(R.id.button3);
final BluetoothAdapter mBluetoothAdapter = BluetoothAdapter
    .getDefaultAdapter();
if (mBluetoothAdapter == null) {
    out.append("Dispositivo BT no soportado!!!");
}

```

```
}
```

```
button1.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        if (!BluetoothAdapter.isEnabled()) {  
            Intent enableBtIntent = new Intent(  
                BluetoothAdapter.ACTION_REQUEST_ENABLE);  
            startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);  
        }  
    }  
});
```

```
button2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View arg0) {  
        if (!BluetoothAdapter.isDiscovering()) {  
            // out.append("MAKING YOUR DEVICE DISCOVERABLE");  
            Context context = getApplicationContext();  
            CharSequence text = "MAKING YOUR DEVICE DISCOVERABLE";  
            int duration = Toast.LENGTH_SHORT;  
            Toast toast = Toast.makeText(context, text, duration);  
            toast.show();  
            Intent enableBtIntent = new Intent(  
                BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);  
            startActivityForResult(enableBtIntent,  
                REQUEST_DISCOVERABLE_BT);  
        }  
    }  
});
```

1. El último apagará la interfaz *bluetooth* del dispositivo móvil.

```
button3.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View arg0) {  
        mBluetoothAdapter.disable();  
        // out.append("TURN_OFF BLUETOOTH");  
        Context context = getApplicationContext();  
        CharSequence text = "TURNING_OFF BLUETOOTH";  
        int duration = Toast.LENGTH_LONG;  
        Toast toast = Toast.makeText(context, text, 15);  
        toast.show();  
    }  
});
```

## 4.3. Búsqueda de dispositivos *bluetooth*

### 4.3.1. Consultar dispositivos vinculados

```
// Get the local Bluetooth adapter
mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

// Get a set of currently paired devices
Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();

// If there are paired devices, add each one to the ArrayAdapter
if (pairedDevices.size() > 0) {
    findViewById(R.id.title_paired_devices).setVisibility(View.VISIBLE);
    for (BluetoothDevice device : pairedDevices) {
        mPairedDevicesArrayAdapter.add(device.getName() + "\n" + device.getAddress());
    }
} else {
    String noDevices = getResources().getText(R.string.none_paired).toString();
    mPairedDevicesArrayAdapter.add(noDevices);
}
```

### 4.3.2. Descubriendo dispositivos

```
// The BroadcastReceiver that listens for discovered devices
private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();

        // When discovery finds a device
        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            // Get the BluetoothDevice object from the Intent
            BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
            mNewDevicesArrayAdapter.add(device.getName() + "\n" + device.getAddress());
        }
    }
};

// Register for broadcasts when a device is discovered
IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
this.registerReceiver(mReceiver, filter);
```

### 4.3.3. Habilitar Descubrimiento

```
private void ensureDiscoverable() {
    if (D) Log.d(TAG, "ensure discoverable");
```

```

        if (mBluetoothAdapter.getScanMode() !=
            BluetoothAdapter.SCAN_MODE_CONNECTABLE_DISCOVERABLE) {
            Intent discoverableIntent = new In-
tent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
            discoverable-
Intent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 300);
            startActivity(discoverableIntent);
        }
    }
}

```

Application name: Bluetooth02  
 Project name: Bluetooth02  
 Package name: com.bluetooth02  
 Activity name: Bluetooth02  
 Layout name: activity\_bluetooth02  
 Resto de parámetros por defecto.

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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"
    tools:context=".Bluetooth02" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="18dp"
        android:layout_marginTop="61dp"
        android:text="Showing Paired Devices:" />

</RelativeLayout>

```

```

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

```

```

TextView textView1;
private static final int REQUEST_ENABLE_BT = 1;
BluetoothAdapter btAdapter;

```

```

textView1 = (TextView) findViewById(R.id.textView1);
// Getting the Bluetooth adapter
btAdapter = BluetoothAdapter.getDefaultAdapter();
textView1.append("\nAdapter: " + btAdapter);

```

```
checkBluetoothState();
```

```
private void checkBluetoothState() {
    // Checks for Bluetooth support and then makes sure it is turned on // If
    // it isn't turned on, request to turn it on
    // List paired devices
    if (btAdapter == null) {
        textView1.append("\nBluetooth NOT supported. Aborting.");
        return;
    } else {
        if (btAdapter.isEnabled()) {
            textView1.append("\nBluetooth is enabled...");

            // Listing paired devices
            textView1.append("\nPaired Devices are:");
            Set<BluetoothDevice> devices = btAdapter.getBondedDevices();
            for (BluetoothDevice device : devices) {
                textView1.append("\n Device: " + device.getName() + ", "
                    + device);
            }
        } else {
            // Prompt user to turn on Bluetooth
            Intent enableBtIntent = new Intent( BluetoothAdapt-
er.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
        }
    }
}
```

```
/* It is called when an activity completes. */
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_ENABLE_BT) {
        CheckBluetoothState();
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
}
```

## 4.4. Conexión de dispositivos

### 4.4.1. Conexión como servidor

```
private class AcceptThread extends Thread {
```

```
private final BluetoothServerSocket mmServerSocket;

public AcceptThread() {
    // Use a temporary object that is later assigned to mmServerSocket,
    // because mmServerSocket is final
    BluetoothServerSocket tmp = null;
    try {
        // MY_UUID is the app's UUID string, also used by the client code
        tmp = mBluetoothAdapter.listenUsingRfcommWithServiceRecord(NAME,
MY_UUID);
    } catch (IOException e) { }
    mmServerSocket = tmp;
}

public void run() {
    BluetoothSocket socket = null;
    // Keep listening until exception occurs or a socket is returned
    while (true) {
        try {
            socket = mmServerSocket.accept();
        } catch (IOException e) {
            break;
        }
        // If a connection was accepted
        if (socket != null) {
            // Do work to manage the connection (in a separate thread)
            manageConnectedSocket(socket);
            mmServerSocket.close();
            break;
        }
    }
}

/** Will cancel the listening socket, and cause the thread to finish */
public void cancel() {
    try {
        mmServerSocket.close();
    } catch (IOException e) { }
}
```

#### 4.4.2. Conexión como cliente

```
private class ConnectThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final BluetoothDevice mmDevice;

    public ConnectThread(BluetoothDevice device) {
        // Use a temporary object that is later assigned to mmSocket,
        // because mmSocket is final
        BluetoothSocket tmp = null;
        mmDevice = device;
    }
}
```

```

// Get a BluetoothSocket to connect with the given BluetoothDevice
try {
    // MY_UUID is the app's UUID string, also used by the server
code
    tmp = device.createRfcommSocketToServiceRecord(MY_UUID);
} catch (IOException e) { }
mmSocket = tmp;
}

public void run() {
    // Cancel discovery because it will slow down the connection
    mBluetoothAdapter.cancelDiscovery();

    try {
        // Connect the device through the socket. This will block
        // until it succeeds or throws an exception
        mmSocket.connect();
    } catch (IOException connectException) {
        // Unable to connect; close the socket and get out
        try {
            mmSocket.close();
        } catch (IOException closeException) { }
        return;
    }

    // Do work to manage the connection (in a separate thread)
    manageConnectedSocket(mmSocket);
}

/** Will cancel an in-progress connection, and close the socket */
public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) { }
}
}

```

#### 4.4.3. Gestión de la conexión

```

private class ConnectedThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;

    public ConnectedThread(BluetoothSocket socket, String socketType) {
        Log.d(TAG, "create ConnectedThread: " + socketType);
        mmSocket = socket;
        InputStream tmpIn = null;
        OutputStream tmpOut = null;
        // Get the BluetoothSocket input and output streams
        try {

```

```

        tmpIn = socket.getInputStream();
        tmpOut = socket.getOutputStream();
    } catch (IOException e) {
        Log.e(TAG, "temp sockets not created", e);
    }

    mmInStream = tmpIn;
    mmOutStream = tmpOut;
}

public void run() {
    Log.i(TAG, "BEGIN mConnectedThread");
    byte[] buffer = new byte[1024];
    int bytes;
    // Keep listening to the InputStream while connected
    while (true) {
        try {
            // Read from the InputStream
            bytes = mmInStream.read(buffer);
            // Send the obtained bytes to the UI Activity
            mHandler.obtainMessage(BluetoothChat.MESSAGE_READ, bytes,
                                   -1, buffer).sendToTarget();
        } catch (IOException e) {
            Log.e(TAG, "disconnected", e);
            connectionLost();
            // Start the service over to restart listening mode
            BluetoothChatService.this.start();
            break;
        }
    }
}

// Write to the connected OutStream.
// @param buffer The bytes to write
public void write(byte[] buffer) {
    try {
        mmOutStream.write(buffer);
        // Share the sent message back to the UI Activity
        mHandler.obtainMessage(BluetoothChat.MESSAGE_WRITE, -1, -1, buffer)
            .sendToTarget();
    } catch (IOException e) {
        Log.e(TAG, "Exception during write", e);
    }
}

public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of connect socket failed", e);
    }
}
}

```



```
// Message types sent from the BluetoothChatService Handler
public static final int MESSAGE_STATE_CHANGE = 1;
public static final int MESSAGE_READ = 2;
public static final int MESSAGE_WRITE = 3;
public static final int MESSAGE_DEVICE_NAME = 4;
public static final int MESSAGE_TOAST = 5;

// The Handler that gets information back from the BluetoothChatService
private final Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        switch (msg.what) {
            case MESSAGE_STATE_CHANGE:
                if(D) Log.i(TAG, "MESSAGE_STATE_CHANGE: " + msg.arg1);
                switch (msg.arg1) {
                    case MESSAGE_WRITE:
                        byte[] writeBuf = (byte[]) msg.obj;
                        // construct a string from the buffer
                        String writeMessage = new String(writeBuf);
                        mConversationArrayAdapter.add("Me: " + writeMessage);
                        break;
                    case MESSAGE_READ:
                        byte[] readBuf = (byte[]) msg.obj;
                        // construct a string from the valid bytes in the buffer
                        String readMessage = new String(readBuf, 0, msg.arg1);
                        mConversationArrayAdapter.add(mConnectedDeviceName+": " +
                                                    readMessage);
                        break;
                    case MESSAGE_DEVICE_NAME:
                        // save the connected device's name
                        mConnectedDeviceName = msg.getData().getString(DEVICE_NAME);
                        Toast.makeText(getApplicationContext(), "Connected to "
                                + mConnectedDeviceName, Toast.LENGTH_SHORT).show();
                        break;
                    case MESSAGE_TOAST:
                        Toast.makeText(getApplicationContext(), msg.getData().
                                getString(Toast.LENGTH_SHORT).show());
                        break;
                }
            }
        }
    };
};
```

## 4.5. Intercambio de información con Android bluetooth

### 4.5.1. ServidorBT

Application name: ServidorBT

Project name: ServidorBT  
Package name: com.servidorbt  
Activity name: ServidorBT  
Layout name: activity\_servidor\_bt  
Resto de parámetros por defecto.

#### 4.5.1.1. Actividad principal. ServidorBT

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".ServidorBT">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2">
        <Button
            android:id="@+id/btnBluetooth"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:text="" />
    </RelativeLayout>
    <TextView
        android:id="@+id/tvMensaje"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="2" />
    <TextView
        android:id="@+id/tvConexion"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/SinConexion" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <Button
            android:id="@+id/btnEnviar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/Enviar" />
    <EditText
```

```

        android:id="@+id/txtMensaje"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:ems="10">
        <requestFocus />
    </EditText>
</LinearLayout>
<Button
    android:id="@+id/btnSalir"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/Salir" />
</LinearLayout>

```

```

<resources>
    <string name="app_name">ServidorBT</string>
    <string name="action_settings">Settings</string>
    <string name="Enviar">Enviar</string>
    <string name="ActivarBluetooth">Activar bluetooth</string>
    <string name="DesactivarBluetooth">Desactivar bluetooth</string>
    <string name="Confirmar">Confirmar</string>
    <string name="Cancelar">Cancelar</string>
    <string name="Salir">Salir</string>

    <!-- Mensajes -->
    <string name="SinConexion">Conexión actual: SIN CONEXIÓN</string>
    <string name="ConexionActual">Conexión actual:</string>
    <string name="ConectandoA">Conectando a</string>
    <string name="EnviandoMensaje">Enviando mensaje</string>
    <string name="MsgErrorConexion">Error al enviar al mensaje: no hay
    conexiones activas</string>
</resources>

```

```

public class ServidorBT extends Activity implements OnClickListener {
    private static final String TAG = "ServidorBT";
    // Declaramos una constante para lanzar los Intent de activación de
    // Bluetooth
    private static final int REQUEST_ENABLE_BT = 1;
    private static final String ALERTA = "alerta";
    // Declaramos una variable privada para cada control de la actividad
    private Button btnEnviar;
    private Button btnBluetooth;
    private Button btnSalir;
    private EditText txtMensaje;
    private TextView tvMensaje;
    private TextView tvConexion;
    private BluetoothAdapter mAdapter; //Adapter para uso del Bluetooth
    private ServServidorBT servicio; //Servicio de mensajes de Bluetooth
    private BluetoothDevice ultimoDispositivo;

```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_servidor_bt);
    // Invocamos el metodo de configuracion de nuestros controles
    configurarControles();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_servidor_bt, menu);
    return true;
}
```

```
private void configurarControles(){
    // Referenciamos los controles y registramos los listeners
    referenciarControles();
    registrarEventosControles();
    // Por defecto, desactivamos los botones que no puedan utilizarse
    btnEnviar.setEnabled(false);
    // Configuramos el adaptador bluetooth y nos suscribimos a sus eventos
    configurarAdaptadorBluetooth();
    registrarEventosBluetooth();
}

private void referenciarControles(){
    // Referenciamos los elementos de interfaz
    btnEnviar = (Button)findViewById(R.id.btnEnviar);
    btnBluetooth = (Button)findViewById(R.id.btnBluetooth);
    btnSalir = (Button)findViewById(R.id.btnSalir);
    txtMensaje = (EditText)findViewById(R.id.txtMensaje);
    tvMensaje = (TextView)findViewById(R.id.tvMensaje);
    tvConexion = (TextView)findViewById(R.id.tvConexion);
}
```

```
private void configurarAdaptadorBluetooth(){
    // Obtenemos el adaptador Bluetooth. Si es NULL, significara que el
    // dispositivo no posee Bluetooth, por lo que deshabilitamos el boton
    // encargado de activar/desactivar esta característica.
    bAdapter = BluetoothAdapter.getDefaultAdapter();
    if(bAdapter == null){
        btnBluetooth.setEnabled(false);
        return;
    }

    // Comprobamos si el Bluetooth esta activo y cambiamos el texto de
    // los botones dependiendo del estado. Tambien activamos o
    // desactivamos los botones asociados a la conexion
    if(bAdapter.isEnabled()) {
        btnBluetooth.setText(R.string.DesactivarBluetooth);
    } else {
        btnBluetooth.setText(R.string.ActivarBluetooth);
    }
}
```

```

    }
}

```

```

private final BroadcastReceiver bReceiver = new BroadcastReceiver(){
    @Override
    public void onReceive(Context context, Intent intent){
        final String action = intent.getAction();

        // BluetoothAdapter.ACTION_STATE_CHANGED
        // Codigo que se ejecutara cuando el Bluetooth cambie su estado.
        // Manejaremos Los siguientes estados:
        // - STATE_OFF: El Bluetooth se desactiva
        // - STATE_ON: El Bluetooth se activa
        if (BluetoothAdapter.ACTION_STATE_CHANGED.equals(action)){
            final int estado = intent.getIntExtra(BluetoothAdapter.EXTRA_STATE,
                BluetoothAdapter.ERROR);
            switch (estado){
                // Apagado
                case BluetoothAdapter.STATE_OFF: {
                    Log.v(TAG, "onReceive: Apagando");
                    ((Button)findViewById(R.id.btnBluetooth)).setText(R.string.ActivarBluetooth);
                    break;
                }
                // Encendido
                case BluetoothAdapter.STATE_ON: {
                    Log.v(TAG, "onReceive: Encendiendo");
                    ((Button)findViewById(R.id.btnBluetooth)).setText(R.string.DesactivarBluetooth);
                    Intent discoverableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
                    discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION, 120);
                    startActivity(discoverableIntent);
                    break;
                }
                default:
                    break;
            } // Fin switch
        } // Fin if
    } // Fin onReceive
};

```

```

private void registrarEventosBluetooth(){
    // Registramos el BroadcastReceiver que instanciamos previamente para
    // detectar los distintos eventos que queremos recibir
    IntentFilter filtro = new IntentFilter(BluetoothAdapter.ACTION_STATE_CHANGED);
}

```

```

    this.registerReceiver(bReceiver, filtro);
}

```

```

@Override
public void onClick(View v) {
    switch(v.getId()){
        // Codigo ejecutado al pulsar el Button que se va a encargar de
        // enviar los datos al otro dispositivo.
        case R.id.btnEnviar: {
            if(servicio != null){
                servi-
cio.enviar(txtMensaje.getText().toString().getBytes());
                txtMensaje.setText("");
            }
            break;
        }
        // Codigo ejecutado al pulsar el Button que se va a encargar de
        // activar y desactivar el Bluetooth.
        case R.id.btnBluetooth: {
            if(bAdapter.isEnabled()) {
                if(servicio != null)
                    servicio.finalizarServicio();
                bAdapter.disable();
            }
            else {
                // Lanzamos el Intent que mostrara la interfaz de
                // activacion del Bluetooth. La respuesta de este
                // Intent se maneja en el metodo onActivityResult
                Intent enableBtIntent = new In-
tent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
            }
            break;
        }

        case R.id.btnSalir: {
            if(servicio != null)
                servicio.finalizarServicio();
            finish();
            System.exit(0);
            break;
        }
        default:
            break;
    }
}
}

```

```

private void registrarEventosControles()
{
    // Asignamos los handlers de los botones
    btnEnviar.setOnClickListener(this);
    btnBluetooth.setOnClickListener(this);
}

```

```

    btnSalir.setOnClickListener(this);
}

```

```

protected void onActivityResult (int requestCode, int resultCode, Intent
data) {
    switch(requestCode)
    {
        case REQUEST_ENABLE_BT:
        {
            Log.v(TAG, "onActivityResult: REQUEST_ENABLE_BT");
            if(resultCode == RESULT_OK)
            {
                btnBluetooth.setText(R.string.DesactivarBluetooth);
                if(servicio != null)
                {
                    servicio.finalizarServicio();
                    servicio.iniciarServicio();
                }
                else
                    servicio = new ServServidorBT(this, handler, bAdap-
ter);
            }
            break;
        }
        default:
            break;
    }
}

```

```

@Override
public void onDestroy() {
    super.onDestroy();
    this.unregisterReceiver(bReceiver);
    if(servicio != null)
        servicio.finalizarServicio();
}
@Override
public synchronized void onResume() {
    super.onResume();
    if(servicio != null) {
        if(servicio.getEstado() == ServServidorBT.ESTADO_NINGUNO) {
            servicio.iniciarServicio();
        }
    }
}
@Override
public synchronized void onPause() {
    super.onPause();
}

```

## 4.5.1.2. Hilo Servidor. ServServidorBT

```

public class ServServidorBT {
    private static final String TAG = "ServServidorBT";
    private static final boolean DEBUG_MODE = true;
    private final Handler handler;
    private final Context context;
    private final BluetoothAdapter bAdapter;
    public static final String NOMBRE_SEGURO = "ServServidorBTSecure";
    public static UUID UUID_SEGURO = UUID.fromString("12345678-4321-4111-
ADDA-345127542950");
    public static final int ESTADO_NINGUNO = 0;
    public static final int ESTADO_CONECTADO = 1;
    public static final int ESTADO_REALIZANDO_CONEXION = 2;
    public static final int ESTADO_ATENDIENDO_PETICIONES = 3;
    public static final int MSG_CAMBIO_ESTADO = 10;
    public static final int MSG_LEER = 11;
    public static final int MSG_ESCRIBIR = 12;
    public static final int MSG_ATENDER_PETICIONES = 13;
    public static final int MSG_ALERTA = 14;
    private int estado;
    private HiloServidor hiloServidor = null;
    private HiloConexion hiloConexion = null;

```

```

public ServServidorBT(Context context, Handler handler, BluetoothAdapter
adapter){
    debug("ServServidorBT()", "Iniciando metodo");
    this.context = context;
    this.handler = handler;
    this.bAdapter = adapter;
    this.estado = ESTADO_NINGUNO;
}

```

```

private synchronized void setEstado(int estado){
    this.estado = estado;
    handler.obtainMessage(MSG_CAMBIO_ESTADO, estado, -1).sendToTarget();
}

public synchronized int getEstado(){
    return estado;
}

```

```

private class HiloConexion extends Thread {
    private final BluetoothSocket socket; // Socket
    private final InputStream inputStream; // Flujo de entrada
    private final OutputStream outputStream; // Flujo de salida

```



```

public HiloConexion(BluetoothSocket socket) {
    this.socket = socket;
    setName(socket.getRemoteDevice().getName() + " [" +
            socket.getRemoteDevice().getAddress() + "]");
    // Se usan variables temporales debido a que los atributos se
    // declaran como final no sería posible asignarles valor
    // posteriormente si fallara esta llamada
    InputStream tmpInputStream = null;
    OutputStream tmpOutputStream = null;
    // Obtenemos los flujos de entrada y salida del socket.
    try {
        tmpInputStream = socket.getInputStream();
        tmpOutputStream = socket.getOutputStream();
    }
    catch(IOException e){
        Log.e(TAG, "HiloConexion(): Error al obtener flujos de E/S", e);
    }
    inputStream = tmpInputStream;
    outputStream = tmpOutputStream;
}

```

```

// Metodo principal del hilo, encargado de realizar las lecturas
public void run(){
    debug("HiloConexion.run()", "Iniciando metodo");
    byte[] buffer = new byte[1024];
    int bytes;
    setEstado(ESTADO_CONECTADO);
    // Mientras se mantenga la conexión el hilo se mantiene en espera
    // ocupado leyendo del flujo de entrada
    while(true) {
        try {
            // Leemos del flujo de entrada del socket
            bytes = inputStream.read(buffer);
            // Enviamos la inf a la actividad a través del handler.
            // El método handleMessage se encarga de recibir el mensaje
            // y mostrar los datos recibidos en el TextView
            handler.obtainMessage(MSG_LEER, bytes, -1, buf-
fer).sendToTarget();
            sleep(500);
        }
        catch(IOException e) {
            Log.e(TAG, "HiloConexion.run(): Error al realizar la lectura",
e);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

```

public void escribir(byte[] buffer){
    try {
        // Escribimos en el flujo de salida del socket
    }
}

```

```

        outputStream.write(buffer);
        // Enviamos la informacion a la actividad a traves del handler.
        // El metodo handleMessage sera el encargado de recibir el mensaje
        // y mostrar los datos enviados en el Toast
        handler.obtainMessage(MSG_ESCRIBIR, -1, -1, buffer).sendToTarget();
    }
    catch(IOException e) {
        Log.e(TAG, "HiloConexion.escribir(): Error al realizar la escritura", e);
    }
}

```

```

public void cancelarConexion(){
    debug("HiloConexion.cancelarConexion()", "Iniciando metodo");
    try {
        // Forzamos el cierre del socket
        socket.close();
        // Cambiamos el estado del servicio
        setEstado(ESTADO_NINGUNO);
    }
    catch(IOException e) {
        Log.e(TAG, "HiloConexion.cerrarConexion(): Error al cerrar la conexion", e);
    }
}
} // Fin HiloConexion

```

```

private class HiloServidor extends Thread {
    private final BluetoothServerSocket serverSocket;
    public HiloServidor(){
        BluetoothServerSocket tmpServerSocket = null;
        // Creamos un socket para escuchar las peticiones de conexion
        try {
            tmpServerSocket = bAdapter.listenUsingRfcommWithServiceRecord(NOMBRE_SEGURO, UUID_SEGURO);
        } catch(IOException e) {
            Log.e(TAG, "HiloServidor(): Error al abrir el socket servidor", e);
        }
        serverSocket = tmpServerSocket;
    }
}

```

```

public void run() {
    debug("HiloServidor.run()", "Iniciando metodo");
    BluetoothSocket socket = null;
    setName("HiloServidor");
    setEstado(ESTADO_ATENDIENDO_PETICIONES);
    // El hilo se mantendra en estado de espera ocupada aceptando
    // conexiones entrantes siempre y cuando no exista una conexion
}

```

```

// activa. En el momento en el que entre una nueva conexion,
while(estado != ESTADO_CONECTADO) {
    try {
        // Cuando un cliente solicite la conexion se abra el socket.
        socket = serverSocket.accept();
    }
    catch(IOException e) {
        Log.e(TAG, "HiloServidor.run(): Error al aceptar conexiones
entrantes", e);
        break;
    }
    // Si el socket tiene valor sera porque un cliente ha solicitado
    // la conexion
    if(socket != null) {
        // Realizamos un lock del objeto
        synchronized(ServServidorBT.this) {
            switch(estado) {
                case ESTADO_ATENDIENDO_PETICIONES:
                case ESTADO_REALIZANDO_CONEXION: {
                    // Estado esperado, se crea el hilo de conexión
                    // que recibira y enviara los mensajes
                    hiloConexion = new HiloConexion(socket);
                    hiloConexion.start();
                    break;
                }
                case ESTADO_NINGUNO:
                case ESTADO_CONECTADO: {
                    // No preparado o conexion ya realizada. Se
                    // cierra el nuevo socket.
                    try {
                        socket.close();
                    }
                    catch(IOException e) {
                        Log.e(TAG, "HiloServidor.run(): so-
cket.close(). Error al cerrar el socket.", e);
                    }
                    break;
                }
                default:
                    break;
            }
        }
    }
} // Fin while
}

```

```

public void cancelarConexion() {
    try {
        serverSocket.close();
    }
    catch(IOException e) {

```

```

        Log.e(TAG, "HiloServidor.cancelarConexion(): Error al cerrar
el socket", e);
    }
}
} // Fin HiloServidor

```

```

// Inicia el servicio, creando un HiloServidor que se dedicara a atender
// las peticiones de conexion.
public synchronized void iniciarServicio() {
    debug("iniciarServicio()", "Iniciando metodo");
    // Si existe una conexion previa, se cancela
    if(hiloConexion != null) {
        hiloConexion.cancelarConexion();
        hiloConexion = null;
    }
    // Arrancamos el hilo servidor para que empiece a recibir peticiones
    // de conexion
    if(hiloServidor == null) {
        hiloServidor = new HiloServidor();
        hiloServidor.start();
    }
    debug("iniciarServicio()", "Finalizando metodo");
}

public void finalizarServicio() {
    debug("finalizarServicio()", "Iniciando metodo");
    if(hiloConexion != null)
        hiloConexion.cancelarConexion();
    if(hiloServidor != null)
        hiloServidor.cancelarConexion();
    hiloConexion = null;
    hiloServidor = null;
    setEstado(ESTADO_NINGUNO);
}

```

```

public synchronized void realizarConexion(BluetoothSocket socket, Blue-
toothDevice dispositivo) {
    debug("realizarConexion()", "Iniciando metodo");
    hiloConexion = new HiloConexion(socket);
    hiloConexion.start();
}

```

```

// Sincroniza el objeto con el hilo HiloConexion e invoca a su metodo
// escribir() para enviar el mensaje como flujo de salida del socket.
public int enviar(byte[] buffer) {
    debug("enviar()", "Iniciando metodo");
    HiloConexion tmpConexion;
    synchronized(this) {
        if(estado != ESTADO_CONECTADO)
            return -1;
        tmpConexion = hiloConexion;
    }
}

```

```

    tmpConexion.escribir(buffer);
    return buffer.length;
}

```

```

public void debug(String metodo, String msg) {
    if(DEBUG_MODE)
        Log.d(TAG, metodo + ": " + msg);
}

```

#### 4.5.1.3. Obtención de información desde ServServidorBT

```

private final Handler handler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        byte[] buffer = null;
        String mensaje = null;
        // Atendemos al tipo de mensaje
        switch(msg.what) {
            // Mensaje de lectura: se mostrara en el TextView
            case ServServidorBT.MSG_LEER: {
                buffer = (byte[])msg.obj;
                mensaje = new String(buffer, 0, msg.arg1);
                tvMensaje.setText(mensaje);
                break;
            }
            // Mensaje de escritura: se mostrara en el Toast
            case ServServidorBT.MSG_ESCRIBIR: {
                buffer = (byte[])msg.obj;
                mensaje = new String(buffer);
                mensaje = getString(R.string.EnvianoMensaje) + ": " +
mensaje;
                Toast.makeText(getApplicationContext(), mensaje,
Toast.LENGTH_SHORT).show();
                break;
            }
        }
    }
}

```

```

/ Mensaje de cambio de estado
case ServServidorBT.MSG_CAMBIO_ESTADO: {
    switch(msg.arg1) {
        case ServServidorBT.ESTADO_ATENDIENDO_PETICIONES:
            break;
        // CONECTADO: Se muestra el dispositivo al que se ha
        // conectado y se activa el boton de enviar
        case ServServidorBT.ESTADO_CONECTADO: {
            mensaje = getString(R.string.ConexionActual) + " "
+ servicio.getNombreDispositivo();
            Toast.makeText(getApplicationContext(), mensaje,
Toast.LENGTH_SHORT).show();
            tvConexion.setText(mensaje);
        }
    }
}

```

```

        btnEnviar.setEnabled(true);
        break;
    }
    // NINGUNO: Mensaje por defecto. Desactivacion del
    // boton de enviar
    case ServServidorBT.ESTADO_NINGUNO: {
        mensaje = getString(R.string.SinConexion);
        Toast.makeText(getApplicationContext(), mensaje,
Toast.LENGTH_SHORT).show();
        tvConexion.setText(mensaje);
        btnEnviar.setEnabled(false);
        break;
    }
    default:
        break;
    }
    break;
}
// Mensaje de alerta: se mostrara en el Toast
case ServServidorBT.MSG_ALERTA: {
    mensaje = msg.getData().getString(ALERTA);
    Toast.makeText(getApplicationContext(), mensaje,
Toast.LENGTH_SHORT).show();
    break;
}
default:
    break;
}
}
}; // Fin Handler

```

## 4.5.2. ClienteBT

Application name: ClienteBT  
 Project name: ClienteBT  
 Package name: com.clientebt  
 Activity name: ClienteBT  
 Layout name: activity\_cliente\_bt  
 Resto de parámetros por defecto.

### 4.5.2.1. Actividad principal. ClienteBT

```

<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"

```

```

android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".ClienteBT">
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="2" >
    <Button
        android:id="@+id/btnBluetooth"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="" />
</RelativeLayout>
<TextView
    android:id="@+id/tvMensaje"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:text="" />
<ListView
    android:id="@+id/lvDispositivos"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="3" >
</ListView>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <Button
        android:id="@+id/btnBuscarDispositivo"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/BuscarDispositivo" />
    <Button
        android:id="@+id/btnConectarDispositivo"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/MostrarEnlazados" />
</LinearLayout>
<TextView
    android:id="@+id/tvConexion"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/SinConexion" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

```

```

        <Button
            android:id="@+id/btnEnviar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/Enviar" />
        <EditText
            android:id="@+id/txtMensaje"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10" >
            <requestFocus />
        </EditText>
    </LinearLayout>
    <Button
        android:id="@+id/btnSalir"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/Salir" />
</LinearLayout>

```

```

<resources>
    <string name="app_name">ClienteBT</string>
    <string name="action_settings">Settings</string>
    <string name="Enviar">Enviar</string>
    <string name="BuscarDispositivo">Buscar</string>
    <string name="Conectar">Conectar</string>
    <string name="MostrarEnlazados">Enlazados</string>
    <string name="ActivarBluetooth">Activar bluetooth</string>
    <string name="DesactivarBluetooth">Desactivar bluetooth</string>
    <string name="Confirmar">Confirmar</string>
    <string name="Cancelar">Cancelar</string>
    <string name="Salir">Salir</string>

    <!-- Mensajes -->
    <string name="SinConexion">Conexión actual: SIN CONEXIÓN</string>
    <string name="ConexionActual">Conexión actual:</string>
    <string name="ConectandoA">Conectando a</string>
    <string name="IniciandoDescubrimiento">Iniciando búsqueda de dispositi-
vos bluetooth&#8230;</string>
    <string name="ErrorIniciandoDescubrimiento">Error al iniciar búsqueda
de dispositivos bluetooth</string>
    <string name="DetectadoDispositivo">Detectado dispositivo</string>
    <string name="FinBusqueda">Fin de la búsqueda</string>
    <string name="EnviandoMensaje">Enviando mensaje</string>
    <string name="MsgConfirmarConexion">¿Conectar al dispositivo</string>
    <string name="MsgErrorConexion">Error al enviar al mensaje: no hay
conexiones activas</string>
</resources>

```



```

public class ClienteBT extends Activity implements OnClickListener {
    private static final String TAG = "ClienteBT";
    // Declaramos una constante para lanzar los Intent de activacion
    // de Bluetooth
    private static final int REQUEST_ENABLE_BT = 1;
    private static final String ALERTA = "alerta";
    // Declaramos una variable privada para cada control de la actividad
    private Button btnEnviar;
    private Button btnBluetooth;
    private Button btnBuscarDispositivo;
    private Button btnConectarDispositivo;
    private Button btnSalir;
    private EditText txtMensaje;
    private TextView tvMensaje;
    private TextView tvConexion;
    private ListView lvDispositivos;
    private BluetoothAdapter adaptadorBT;
    private ArrayList<BluetoothDevice> arrayDisp;
    private ArrayAdapter arrayAdaptador;
    private ServClienteBT servicio;
    private BluetoothDevice ultimoDispositivo;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_cliente_bt);
    // Invocamos el metodo de configuracion de nuestros controles
    configurarControles();
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_cliente_bt, menu);
    return true;
}

```

```

private void configurarControles() {
    // Instanciamos el array de dispositivos
    arrayDisp = new ArrayList<BluetoothDevice>();
    // Referenciamos los controles y registramos los listeners
    referenciarControles();
    registrarEventosControles();
    // Por defecto, desactivamos los botones que no puedan utilizarse
    btnEnviar.setEnabled(false);
    btnBuscarDispositivo.setEnabled(false);
    btnConectarDispositivo.setEnabled(false);
    // Configuramos el adaptador bluetooth y nos suscribimos a sus eventos
    configurarAdaptadorBluetooth();
    registrarEventosBluetooth();
}

private void referenciarControles() {

```

```

// Referenciamos los elementos de interfaz
btnEnviar = (Button) findViewById(R.id.btnEnviar);
btnBluetooth = (Button) findViewById(R.id.btnBluetooth);
btnBuscarDispositivo = (Button)
findViewById(R.id.btnBuscarDispositivo);
btnConectarDispositivo = (Button)
findViewById(R.id.btnConectarDispositivo);
btnSalir = (Button) findViewById(R.id.btnSalir);
txtMensaje = (EditText) findViewById(R.id.txtMensaje);
tvMensaje = (TextView) findViewById(R.id.tvMensaje);
tvConexion = (TextView) findViewById(R.id.tvConexion);
lvDispositivos = (ListView) findViewById(R.id.lvDispositivos);
}

```

```

private void configurarAdaptadorBluetooth() {
    // Obtenemos el adaptador Bluetooth. Si es NULL, significara que el
    // dispositivo no posee Bluetooth, por lo que deshabilitamos el boton
    // encargado de activar/deshabilitar esta caracteristica.
    adaptadorBT = BluetoothAdapter.getDefaultAdapter();
    if (adaptadorBT == null) {
        btnBluetooth.setEnabled(false);
        return;
    }
    // Comprobamos si el Bluetooth esta activo y cambiamos el texto de
    // los botones dependiendo del estado. Tambien activamos o
    // desactivamos los botones asociados a la conexion
    if (adaptadorBT.isEnabled()) {
        btnBluetooth.setText(R.string.DesactivarBluetooth);
        btnBuscarDispositivo.setEnabled(true);
        btnConectarDispositivo.setEnabled(true);
    } else {
        btnBluetooth.setText(R.string.ActivarBluetooth);
    }
}
}

```

```

private final BroadcastReceiver bReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        final String action = intent.getAction();
        // BluetoothAdapter.ACTION_STATE_CHANGED
        // Codigo que se ejecutara cuando el Bluetooth cambie su estado.
        // Manejaremos los siguientes estados:
        // - STATE_OFF: EL Bluetooth se desactiva
        // - STATE_ON: EL Bluetooth se activa
        if (BluetoothAdapter.ACTION_STATE_CHANGED.equals(action)) {
            final int estado = intent.getIntExtra(BluetoothAdapter.EXTRA_STATE,
                BluetoothAdapter.ERROR);
            switch (estado) {
                // Apagado

```

```

        case BluetoothAdapter.STATE_OFF: {
            Log.v(TAG, "onReceive: Apagando");
            ((Button) findViewById(
                R.id.btnBluetooth)).setText(R.string.ActivarBluetooth);
            ((Button) findViewById(
                R.id.btnBuscarDispositivo)).setEnabled(false);
            ((Button) findViewById(
                R.id.btnConectarDispositivo)).setEnabled(false);
            break;
        }
        // Encendido
        case BluetoothAdapter.STATE_ON: {
            Log.v(TAG, "onReceive: Encendiendo");
            ((Button) findViewById(
                R.id.btnBluetooth)).setText(R.string.DesactivarBluetooth);
            ((Button) findViewById(
                R.id.btnBuscarDispositivo)).setEnabled(true);
            ((Button) findViewById(
                R.id.btnConectarDispositivo)).setEnabled(true);
            break;
        }
        default:
            break;
    } // Fin switch
} // Fin if

```

```

// BluetoothDevice.ACTION_FOUND
// Cada vez que se descubra un nuevo dispositivo por Bluetooth,
// se ejecutara este fragmento de codigo
else if (BluetoothDevice.ACTION_FOUND.equals(action)) {
    if (arrayDisp == null)
        arrayDisp = new ArrayList<BluetoothDevice>();
    BluetoothDevice dispositivo = in-
tent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
    arrayDisp.add(dispositivo);
    String descripcionDispositivo = dispositivo.getName() + " [" +
dispositivo.getAddress() + "]";
    Toast.makeText(getBaseContext(),
getString(R.string.DetectadoDispositivo) + ": " + descripcionDispositivo,
Toast.LENGTH_SHORT).show();
    Log.v(TAG, "ACTION_FOUND: Dispositivo encontrado: " + descrip-
cionDispositivo);
}
// BluetoothAdapter.ACTION_DISCOVERY_FINISHED
// Codigo que se ejecutara cuando el Bluetooth finalice la
// busqueda de dispositivos.
if (BluetoothAdapter.ACTION_DISCOVERY_FINISHED.equals(action)) {
    // Instanciamos un nuevo adapter para el ListView
    arrayAdaptador = new ArrayAdapterBT(getBaseContext(), an-
droid.R.layout.simple_list_item_2, arrayDisp);
    lvDispositivos.setAdapter(arrayAdaptador);
}

```

```

        Toast.makeText(getBaseContext(), R.string.FinBusqueda,
        Toast.LENGTH_SHORT).show();
    }
} // Fin onReceive
}; // Fin BroadcastReceiver

```

#### 4.5.2.2. ArrayAdapterBT

```

public class ArrayAdapterBT extends ArrayAdapter {
    // Contiene el listado de dispositivos
    private List<BluetoothDevice> deviceList;
    // Contexto activo
    private Context context;
    public ArrayAdapterBT(Context context, int textViewResourceId,
    List<BluetoothDevice> objects) {
        super(context, textViewResourceId, objects);
        this.deviceList = objects;
        this.context = context;
    }
    @Override
    public int getCount() {
        if (deviceList != null)
            return deviceList.size();
        else
            return -1;
    }
    @Override
    public Object getItem(int position) {
        return (deviceList == null ? null : deviceList.get(position));
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent){
        if ((deviceList == null) || (context == null))
            return null;
        // Usamos un LayoutInflater para crear las vistas
        LayoutInflater inflater = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        // Creamos una vista a partir de simple_list_item_2, que contiene
        // dos TextView. El primero (text1) lo usaremos para el nombre,
        // mientras que el segundo (text2) lo utilizaremos para la
        // direccion del dispositivo.
        View elemento = inflater.inflate(android.R.layout.simple_list_item_2, parent, false);
        // Referenciamos los TextView
        TextView txtNombre = (TextView) elemento.findViewById(android.R.id.text1);
        TextView txtDireccion = (TextView) elemento.findViewById(android.R.id.text2);
        // Obtenemos el dispositivo del array y obtenemos su nombre y
        // direccion, asociandolos a los dos TextView del elemento
        BluetoothDevice dispositivo = (BluetoothDevice) getItem(position);
        if (dispositivo != null) {
            txtNombre.setText(dispositivo.getName());

```

```

        txtDireccion.setText(dispositivo.getAddress());
    } else {
        txtNombre.setText("ERROR");
    }
    // Devolvemos el elemento con los dos TextView cumplimentados
    return elemento;
}
}

```

#### 4.5.2.3. Hilo Cliente. ServClienteBT

```

public class ServClienteBT {
    public static final String NOMBRE_SEGURO = "ServClienteBTSecure";
    public static final int ESTADO_NINGUNO = 0;
    public static final int ESTADO_CONECTADO = 1;
    public static final int ESTADO_REALIZANDO_CONEXION = 2;
    public static final int ESTADO_ATENDIENDO_PETICIONES = 3;
    public static final int MSG_CAMBIO_ESTADO = 10;
    public static final int MSG_LEER = 11;
    public static final int MSG_ESCRIBIR = 12;
    public static final int MSG_ATENDER_PETICIONES = 13;
    public static final int MSG_ALERTA = 14;
    private static final String TAG = "ServClienteBT";
    private static final boolean DEBUG_MODE = true;
    public static UUID UUID_SEGURO = UUID.fromString("12345678-4321-4111-
ADDA-345127542950");
    private final Handler handler;
    private final Context context;
    private final BluetoothAdapter bAdapter;
    private int estado;
    private HiloCliente hiloCliente = null;
    private HiloConexion hiloConexion = null;
}

```

```

public ServClienteBT(Context context, Handler handler, BluetoothAdapter
adapter){
    debug("BluetoothService()", "Iniciando metodo");
    this.context = context;
    this.handler = handler;
    this.bAdapter = adapter;
    this.estado = ESTADO_NINGUNO;
}

```

```

private synchronized void setEstado(int estado){
    this.estado = estado;
    handler.obtainMessage(MSG_CAMBIO_ESTADO, estado, -1).sendToTarget();
}
public synchronized int getEstado(){
    return estado;
}

```

```

// Hilo encargado de mantener la conexion y realizar las lecturas y
// escrituras de los mensajes intercambiados entre dispositivos.

```

```

private class HiloConexion extends Thread {
    private final BluetoothSocket socket; // Socket
    private final InputStream inputStream; // Flujo de entrada (lecturas)
    private final OutputStream outputStream; // Flujo de salida (escrituras)

    public HiloConexion(BluetoothSocket socket) {
        this.socket = socket;
        setName(socket.getRemoteDevice().getName() + " [" + socket.getRemoteDevice().getAddress() + "]");
        // Se usan variables temporales debido a que los atributos se declaran como final no seria posible asignarles valor posteriormente si fallara esta llamada
        InputStream tmpInputStream = null;
        OutputStream tmpOutputStream = null;
        // Obtenemos los flujos de entrada y salida del socket.
        try {
            tmpInputStream = socket.getInputStream();
            tmpOutputStream = socket.getOutputStream();
        }
        catch(IOException e){
            Log.e(TAG, "HiloConexion(): Error al obtener flujos de E/S", e);
        }
        inputStream = tmpInputStream;
        outputStream = tmpOutputStream;
    }
    // Metodo principal del hilo, encargado de realizar las lecturas
    public void run() {
        debug("HiloConexion.run()", "Iniciando metodo");
        byte[] buffer = new byte[1024];
        int bytes;
        setEstado(ESTADO_CONECTADO);
        // Mientras se mantenga la conexion el hilo se mantiene en espera // ocupada leyendo del flujo de entrada
        while(true) {
            try {
                // Leemos del flujo de entrada del socket
                bytes = inputStream.read(buffer);
                // Enviamos la informacion a la actividad con el handler.
                // El metodo handleMessage sera el encargado de recibir el // mensaje y mostrar los datos recibidos en el TextView
                handler.obtainMessage(MSG_LEER, bytes, -1, buffer).sendToTarget();
                sleep(500);
            }
            catch(IOException e) {
                Log.e(TAG, "HiloConexion.run(): Error al realizar la lectura", e);
            }
            catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

public void escribir(byte[] buffer) {
    try {
        // Escribimos en el flujo de salida del socket
        outputStream.write(buffer);
        // Enviamos la informacion a la actividad con el handler.
        // El metodo handleMessage sera el encargado de recibir el
        // mensaje y mostrar los datos enviados en el Toast
        handler.obtainMessage(MSG_ESCRIBIR, -1, -1, buffer).sendToTarget();
    }
    catch(IOException e) {
        Log.e(TAG, "HiloConexion.escribir(): Error al realizar la escritura", e);
    }
}

public void cancelarConexion() {
    debug("HiloConexion.cancelarConexion()", "Iniciando metodo");
    try {
        // Forzamos el cierre del socket
        socket.close();
        // Cambiamos el estado del servicio
        setEstado(ESTADO_NINGUNO);
    }
    catch(IOException e) {
        Log.e(TAG, "HiloConexion.cerrarConexion(): Error al cerrar la conexion", e);
    }
}
} // Fin HiloConexion

```

```

private class HiloCliente extends Thread {
    private final BluetoothDevice dispositivo;
    private final BluetoothSocket socket;
    public HiloCliente(BluetoothDevice dispositivo) {
        BluetoothSocket tmpSocket = null;
        this.dispositivo = dispositivo;
        // Obtenemos un socket para el dispositivo con el que se quiere
        // conectar
        try {
            tmpSocket = dispositivo.createRfcommSocketToServiceRecord(UUID_SEGURO);
        }
        catch(IOException e) {
            Log.e(TAG, "HiloCliente.HiloCliente(): Error al abrir el socket", e);
        }
        socket = tmpSocket;
    }
}

```

```

public void run() {
    setName("HiloCliente");
    if(bAdapter.isDiscovering())
        bAdapter.cancelDiscovery();

    try {
        socket.connect();
        setEstado(ESTADO_REALIZANDO_CONEXION);
    }
    catch(IOException e) {
        Log.e(TAG, "HiloCliente.run(): socket.connect(): Error realizando
la conexion", e);
        try {
            socket.close();
        }
        catch(IOException inner) {
            Log.e(TAG, "HiloCliente.run(): Error cerrando el socket", in-
ner);
        }
        setEstado(ESTADO_NINGUNO);
    }

    // Reiniciamos el hilo cliente, ya que no lo necesitaremos mas
    synchronized(ServClienteBT.this) {
        hiloCliente = null;
    }

    // Realizamos la conexion
    hiloConexion = new HiloConexion(socket);
    hiloConexion.start();
}

```

```

public void cancelarConexion() {
    debug("cancelarConexion()", "Iniciando metodo");
    try {
        socket.close();
    }
    catch(IOException e) {
        Log.e(TAG, "HiloCliente.cancelarConexion(): Error al cerrar el
socket", e);
    }
    setEstado(ESTADO_NINGUNO);
}
} // Fin HiloCliente

```

```

// Instancia un hilo conector
public synchronized void solicitarConexion(BluetoothDevice dispositivo) {
    debug("solicitarConexion()", "Iniciando metodo");
    // Comprobamos si existia un intento de conexion en curso.
    // Si es el caso, se cancela y se vuelve a iniciar el proceso
    if(estado == ESTADO_REALIZANDO_CONEXION) {

```



```

        if(hiloCliente != null) {
            hiloCliente.cancelarConexion();
            hiloCliente = null;
        }
    }
    // Si existia una conexion abierta, se cierra y se inicia una nueva
    if(hiloConexion != null) {
        hiloConexion.cancelarConexion();
        hiloConexion = null;
    }
    // Se instancia un nuevo hilo conector, encargado de solicitar una
    // conexion al servidor, que sera la otra parte.
    hiloCliente = new HiloCliente(dispositivo);
    hiloCliente.start();
    setEstado(ESTADO_REALIZANDO_CONEXION);
}

```

```

public synchronized void realizarConexion(BluetoothSocket socket, Blue-
toothDevice dispositivo) {
    debug("realizarConexion()", "Iniciando metodo");
    hiloConexion = new HiloConexion(socket);
    hiloConexion.start();
}

```

```

public int enviar(byte[] buffer) {
    debug("enviar()", "Iniciando metodo");
    HiloConexion tmpConexion;
    synchronized(this) {
        if(estado != ESTADO_CONECTADO)
            return -1;
        tmpConexion = hiloConexion;
    }
    tmpConexion.escribir(buffer);
    return buffer.length;
}

```

```

public void finalizarServicio() {
    debug("finalizarServicio()", "Iniciando metodo");
    if(hiloCliente != null)
        hiloCliente.cancelarConexion();
    if(hiloConexion != null)
        hiloConexion.cancelarConexion();
    hiloCliente = null;
    hiloConexion = null;
    setEstado(ESTADO_NINGUNO);
}
public void debug(String metodo, String msg) {
    if(DEBUG_MODE)
        Log.d(TAG, metodo + ": " + msg);
}

```

## 4.5.2.4. Finalización ClienteBT

```
private void registrarEventosBluetooth() {
    // Registramos el BroadcastReceiver que instanciamos previamente para
    // detectar los distintos eventos que queremos recibir
    IntentFilter filtro = new IntentFilter(
        BluetoothAdapter.ACTION_STATE_CHANGED);
    filtro.addAction(BluetoothDevice.ACTION_FOUND);
    filtro.addAction(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);
    this.registerReceiver(bReceiver, filtro);
}
```

```
@Override
public void onClick(View v) {
    switch (v.getId()) {
        // Codigo ejecutado al pulsar el Button que se va a encargar de
        // enviar los datos al otro dispositivo.
        case R.id.btnEnviar: {
            if (servicio != null) {
                servicio.enviar(txtMensaje.getText().toString().getBytes());
                txtMensaje.setText("");
            }
            break;
        }
        // Codigo ejecutado al pulsar el Button que se va a encargar de
        // activar y desactivar el Bluetooth.
        case R.id.btnBluetooth: {
            if (adaptadorBT.isEnabled()) {
                if (servicio != null)
                    servicio.finalizarServicio();
                adaptadorBT.disable();
            } else {
                // Lanzamos el Intent que mostrara la interfaz de
                // activacion del Bluetooth. La respuesta de este Intent
                // se maneja en el metodo onActivityResult
                Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
            }
            break;
        }
        // Codigo ejecutado al pulsar el Button que se va a encargar de
        // descubrir nuevos dispositivos
        case R.id.btnBuscarDispositivo: {
            arrayDisp.clear();
            // Comprobamos si existe un descubrimiento en curso. En caso
            // afirmativo, se cancela.
            if (adaptadorBT.isDiscovering())
                adaptadorBT.cancelDiscovery();
            // Iniciamos la busqueda de dispositivos
            if (adaptadorBT.startDiscovery())
                // Mostramos el mensaje de que el proceso ha comenzado
        }
    }
}
```

```

        Toast.makeText(this, R.string.IniciandoDescubrimiento,
        Toast.LENGTH_SHORT).show();
    }
    else
        Toast.makeText(this,
        R.string.ErrorIniciandoDescubrimiento, Toast.LENGTH_SHORT).show();
        break;
    }
    // Codigo ejecutado al pulsar el Button que se encarga de mostrar
    // todos los dispositivos previamente enlazados al dispositivo
    // actual.
    case R.id.btnConectarDispositivo: {
        Set<BluetoothDevice> dispositivosEnlazados = adapta-
        dorBT.getBondedDevices();
        // Instanciamos un nuevo adapter para el ListView
        arrayDisp = new Arra-
        yList<BluetoothDevice>(dispositivosEnlazados);
        arrayAdaptador = new ArrayAdapterBT(getBaseContext(), an-
        droid.R.layout.simple_list_item_1, arrayDisp);
        lvDispositivos.setAdapter(arrayAdaptador);
        Toast.makeText(getBaseContext(), R.string.FinBusqueda,
        Toast.LENGTH_SHORT).show();
        break;
    }
    case R.id.btnSalir: {
        if (servicio != null)
            servicio.finalizarServicio();
        finish();
        System.exit(0);
        break;
    }
    default:
        break;
}
}
}

```

```

private void registrarEventosControles() {
    // Asignamos Los handlers de Los botones
    btnEnviar.setOnClickListener(this);
    btnBluetooth.setOnClickListener(this);
    btnBuscarDispositivo.setOnClickListener(this);
    btnConectarDispositivo.setOnClickListener(this);
    btnSalir.setOnClickListener(this);
    // Configuramos la lista de dispositivos para que cuando seleccionemos
    // uno de sus elementos realice la conexión al dispositivo
    configurarListaDispositivos();
}

```

```

private void configurarListaDispositivos() {
    lvDispositivos.setOnItemClickListener(new Adapter-
    View.OnItemClickListener() {
        @Override

```

```

    public void onItemClick(AdapterView adapter, View view, int position, long arg) {
        // El ListView tiene un adaptador de tipo
        // BluetoothDeviceArrayAdapter. Invocamos el metodo getItem()
        // del adaptador para recibir el dispositivo bluetooth y
        // realizar la conexion.
        BluetoothDevice dispositivo = (BluetoothDevice) lvDispositivos.getAdapter().getItem(position);
        AlertDialog dialog = crearDialogoConexion(getString(R.string.Conectar),
            getString(R.string.MsgConfirmarConexion) + " " + dispositivo.getName() + "?",
            dispositivo.getAddress());
        dialog.show();
    }
});
}

```

```

private AlertDialog crearDialogoConexion(String titulo, String mensaje, final String direccion) {
    // Instanciamos un nuevo AlertDialog Builder y le asociamos titulo y
    // mensaje
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
    alertDialogBuilder.setTitle(titulo);
    alertDialogBuilder.setMessage(mensaje);
    // Creamos un nuevo OnClickListener para que el boton OK realice la
    // conexion
    DialogInterface.OnClickListener listenerOk = new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            conectarDispositivo(direccion);
        }
    };
    // Creamos un nuevo OnClickListener para el boton Cancelar
    DialogInterface.OnClickListener listenerCancelar = new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            return;
        }
    };
    // Asignamos los botones positivo y negativo a sus respectivos
    // listeners
    alertDialogBuilder.setPositiveButton(R.string.Conectar, listenerOk);
    alertDialogBuilder.setNegativeButton(R.string.Cancelar, listenerCancelar);
    return alertDialogBuilder.create();
}

```

```

public void conectarDispositivo(String direccion) {
    if(adaptadorBT.isEnabled()) {
        Toast.makeText(this, "Conectando a " + direccion,
            Toast.LENGTH_LONG).show();
        if (servicio == null)
        {
            servicio = new ServClienteBT(this, handler, adaptadorBT);
        }
        BluetoothDevice dispositivoRemoto = adaptadorBT.getRemoteDevice(direccion);
        servicio.solicitarConexion(dispositivoRemoto);
        this.ultimoDispositivo = dispositivoRemoto;
    }else{
        Toast.makeText(this, "Imposible conectar: adaptador Bluetooth apagado", Toast.LENGTH_LONG).show();
    }
}

```

```

private final Handler handler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        byte[] buffer = null;
        String mensaje = null;
        // Atendemos al tipo de mensaje
        switch (msg.what) {
            // Mensaje de Lectura: se mostrara en el TextView
            case ServClienteBT.MSG_LEER: {
                buffer = (byte[]) msg.obj;
                mensaje = new String(buffer, 0, msg.arg1);
                tvMensaje.setText(mensaje);
                break;
            }
            // Mensaje de escritura: se mostrara en el Toast
            case ServClienteBT.MSG_ESCRIBIR: {
                buffer = (byte[]) msg.obj;
                mensaje = new String(buffer);
                mensaje = getString(R.string.EnvioMensaje) + ": " +
mensaje;
                Toast.makeText(getApplicationContext(), mensaje,
                    Toast.LENGTH_SHORT).show();
                break;
            }
}

```

```

// Mensaje de cambio de estado
case ServClienteBT.MSG_CAMBIO_ESTADO: {
    switch (msg.arg1) {
        case ServClienteBT.ESTADO_ATENDIENDO_PETICIONES:
            break;
        // CONECTADO: Se muestra el dispositivo al que se ha
        // conectado y se activa el boton de enviar
        case ServClienteBT.ESTADO_CONECTADO: {
            mensaje = getString(R.string.ConexionActual) + " "

```

```

+ servicio.getNombreDispositivo();
    Toast.makeText(getApplicationContext(), mensaje,
    Toast.LENGTH_SHORT).show();
    tvConexion.setText(mensaje);
    btnEnviar.setEnabled(true);
    break;
}
// REALIZANDO CONEXION: Se muestra el dispositivo al
// que se esta conectando
case ServClienteBT.ESTADO_REALIZANDO_CONEXION: {
    mensaje = getString(R.string.ConectandoA) + " " +
ultimoDispositivo.getName() + " [" + ultimoDispositivo.getAddress() + "]";
    Toast.makeText(getApplicationContext(), mensaje,
    Toast.LENGTH_SHORT).show();
    btnEnviar.setEnabled(false);
    break;
}
// NINGUNO: Mensaje por defecto. Desactivacion del
// boton de enviar
case ServClienteBT.ESTADO_NINGUNO: {
    mensaje = getString(R.string.SinConexion);
    Toast.makeText(getApplicationContext(), mensaje,
    Toast.LENGTH_SHORT).show();
    tvConexion.setText(mensaje);
    btnEnviar.setEnabled(false);
    break;
}
default:
    break;
}
break;
}
// Mensaje de alerta: se mostrara en el Toast
case ServClienteBT.MSG_ALERTA: {
    mensaje = msg.getData().getString(ALERTA);
    Toast.makeText(getApplicationContext(), mensaje,
    Toast.LENGTH_SHORT).show();
    break;
}
default:
    break;
}
}
}; // Fin Handler

```

```

protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    switch (requestCode) {
        case REQUEST_ENABLE_BT: {
            Log.v(TAG, "onActivityResult: REQUEST_ENABLE_BT");
            if (resultCode == RESULT_OK) {

```

```

        btnBluetooth.setText(R.string.DesactivarBluetooth);
        if (servicio != null) {
            servicio.finalizarServicio();
        } else
            servicio = new ServClienteBT(this, handler,
adaptadorBT);
    }
    break;
}
default:
    break;
}
}

```

```

@Override
public void onDestroy() {
    super.onDestroy();
    this.unregisterReceiver(bReceiver);
    if (servicio != null)
        servicio.finalizarServicio();
}
@Override
public synchronized void onResume() {
    super.onResume();
}
@Override
public synchronized void onPause() {
    super.onPause();
}
}

```

## 4.6. La comunicación con los programas en Java SE

```

import java.io.*;
import javax.microedition.io.*;
import javax.bluetooth.*;
public class BluetoothSender_JavaSE {
    static Object lock1 = new Object(), lock2 = new Object();
    // Por cuestión de sincronización
    static RemoteDevice partnerDevice = null;
    // El dispositivo al cual enviar los datos
    static String connectionURL = null;
    // URL de conexión de este dispositivo
    public static void main(String args[]) {
    }
}

```

```
static class MyDiscListener implements DiscoveryListener {
    // Ver explicación abajo
    public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
        partnerDevice = btDevice;
        System.out.println("Device discovered: "
            +btDevice.getBluetoothAddress());
        synchronized(lock1) { lock1.notify(); }
    }
    public void servicesDiscovered(
        int transID, ServiceRecord[] servRecord) {
        connectionURL=servRecord[0].getConnectionURL(
            ServiceRecord.AUTHENTICATE_ENCRYPT,false);
        System.out.println(servRecord.length+" service(s) discovered");
        synchronized(lock2) { lock2.notify(); }
    }
    public void serviceSearchCompleted(int transID, int respCode) {}
    public void inquiryCompleted(int discType) {}
}
```

```
try {
    DiscoveryAgent agent =
        LocalDevice.getLocalDevice().getDiscoveryAgent();
    // Agente para descubrir los servicios y dispositivos Bluetooth
    MyDiscListener lis = new MyDiscListener();
    // Discovery listener (ver paso 7)
    agent.startInquiry(DiscoveryAgent.GIAC,lis);
    // Empezar búsqueda de dispositivos
    synchronized(lock1) { try { lock1.wait(); }
        catch (InterruptedException iexc) {} }
    // Esperar hasta que un dispositivo sea encontrado
    // y el método listener haya terminado
    javax.bluetooth.UUID[] uuidSet = new javax.bluetooth.UUID[1];
    uuidSet[0] = new javax.bluetooth.UUID(
        "1234567843214111ADDA345127542950",false);
    // UUID del servicio en el dispositivo Android

    agent.searchServices(null,uuidSet,partnerDevice,lis);
    // Empezar la búsqueda para el servicio especificado
    synchronized(lock2) { try { lock2.wait(); }
        catch (InterruptedException iexc) {} }
    // Esperar hasta que el servicio sea encontrado
    // y el método listener haya terminado
} catch (BluetoothStateException exc) {
    System.out.println("Exception: "+exc.getMessage()); }
System.out.println("Connected to: "+connectionURL);
try {
    StreamConnection sConn =
        (StreamConnection) Connector.open(connectionURL);
    // Abrir la conexión Bluetooth con el servicio y el dispositivo
    OutputStream outputStream = sConn.openOutputStream();
```



```

PrintWriter pwr = new PrintWriter(new OutputStreamWriter(outStream));
pwr.write("Message");
pwr.flush();
System.out.println("Data sent");
try {
    Thread.currentThread().sleep(2000);
    // Dar tiempo para leer los datos
    // (no hay buffer interno!)
} catch (Exception e) {}
pwr.close();
outStream.close();
sConn.close();
System.out.println("Everything closed");
} catch (IOException exc) {
    System.out.println("Exception: "+exc.getMessage()); }

```

```

UUID uuid = new UUID("1234567843214111ADDA345127542950", false);
String hostname = "examplehost";
    // Reemplazar por el nombre del ordenador portátil
final String url = "btspp://localhost:" + uuid +
    ";name=" + hostname + ";authenticate=false;encrypt=false;";
StreamConnectionNotifier host =
    (StreamConnectionNotifier) Connector.open(url);
StreamConnection conn = host.acceptAndOpen();

```

```

DataInputStream din = new DataInputStream(conn.openInputStream());
while (true) {

    byte[] incomingData = new byte[10000];
    din.read(incomingData);
    String s = new String(incomingData);
    System.out.println("\n"+s.trim()+"\n");
}

```



# CAPÍTULO 5.

## Servicios en la nube

Por VICENTE CARBONELL

### 5.1. Aplicación cliente Google Cloud Messaging

#### 5.1.1. Usar Google Play Services en un proyecto

```
apply plugin: 'com.android.application'
...
dependencies {
    compile 'com.google.android.gms:play-services:7.5.0'
}
```

#### 5.1.2. Crear aplicación cliente GCM

```
compile 'com.google.android.gms:play-services:7.+'
```

```
-keep class * extends java.util.ListResourceBundle {
    protected Object[][] getContents();
}
```

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

```
import static org.example.appgcm.UtilidadesGCM.*;

public class ActividadPrincipal extends Activity {
```

```
private final static int PLAY_SERVICES_RESOLUTION_REQUEST = 9000;
Button btnRegistrarUsuario;
Button btnDesregistrarUsuario;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    btnRegistrarUsuario = (Button)
        findViewById(R.id.btnRegistrarUsuario);
    btnDesregistrarUsuario = (Button)
        findViewById(R.id.btnDesregistrarUsuario);
    if (comprobarGooglePlayServices()) {
        mostrarMensaje(this, "Se ha encontrado un Google Play Services
            válido.");
    } else {
        mostrarMensaje(this, "No se ha encontrado un Google Play
            Services válido.");
        finish();
    }
}

@Override
protected void onResume() {
    super.onResume();
    comprobarGooglePlayServices();
}
}
```

```
private boolean comprobarGooglePlayServices() {
    int resultCode = GooglePlayServicesUtil
        .isGooglePlayServicesAvailable(this);
    if (resultCode != ConnectionResult.SUCCESS) {
        if (GooglePlayServicesUtil.isUserRecoverableError(resultCode)) {
            GooglePlayServicesUtil.getErrorDialog(resultCode, this,
                PLAY_SERVICES_RESOLUTION_REQUEST).show();
        } else {
            //Dispositivo no soportado.
            mostrarMensaje(this, "Tienes que tener Google Play Services
                instalado y actualizado.");
            finish();
        }
        return false;
    }
    return true;
}
```

```

public final class UtilidadesGCM {
    private static Handler manejador = new Handler();
    static void mostrarMensaje(final Context context, final String mensaje)
    {
        manejador.post(new Runnable() {
            public void run() {
                Toast.makeText(context, mensaje, Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

```

<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"
/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WAKE_LOCK" />

```

```

<permission android:name="org.example.appgcm.permission.C2D_MESSAGE"
    android:protectionLevel="signature" />
<uses-permission android:name="org.example.appgcm.permission.C2D_MESSAGE"
/>

```

```

<receiver
    android:name="com.google.android.gms.gcm.GcmReceiver"
    android:exported="true"
    android:permission="com.google.android.c2dm.permission.SEND" >
    <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />
        <category android:name="org.example.appgcm" />
    </intent-filter>
</receiver>

```

```

<service
    android:name="org.example.appgcm.MensajelisterService"
    android:exported="false" >
    <intent-filter>
        <action
            android:name="com.google.android.c2dm.intent.RECEIVE"/>
    </intent-filter>
</service>
<service
    android:name="org.example.appgcm.IDListenerService"
    android:exported="false">
    <intent-filter>
        <action android:name="com.google.android.gms.iid.InstanceID"/>
    </intent-filter>
</service>

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".ActividadPrincipal" >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >
        <Button
            android:id="@+id/btnRegistrarUsuario"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="registrarUsuarioGCM"
            android:text="Registrar" />
        <Button
            android:id="@+id/btnDesregistrarUsuario"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="desregistrarUsuarioGCM"
            android:text="Desregistrar" />
    </LinearLayout>
</LinearLayout>

```

```

static final String SERVER_URL =
"http://cursoandroid.hol.es/notificaciones/";
/*Número de Proyecto en Google Console que usa el servicio GCM.
[TIENES SUSTITUIRLO POR EL TUYO] */
static final String SENDER_ID = "1092916126939";
static final String DISPLAY_MESSAGE_ACTION =
"org.example.appgcm.DISPLAY_MESSAGE";

```

### 5.1.3. Registrar dispositivo en GCM

```
public static String idRegistro = "";
```

```

public void registrarUsuarioGCM(View view){
    idRegistro = dameIdRegistro(getApplicationContext());
    if (idRegistro.equals("")) {
        registrarGCM tareaRegistroGCM = new registrarGCM();
        tareaRegistroGCM.execute(getApplicationContext(), null, null);
    } else {
        mostrarMensaje(this, "Dispositivo ya registrado.");
    }
}

```

```

Public static String dameIdRegistro(Context context) {
    final SharedPreferences preferencias = context.getSharedPreferences(
        ActividadPrincipal.class.getSimpleName(), Context.MODE_PRIVATE);
    String auxRegistroId = preferencias.getString("idRegistro", "");
}

```

```

    if (auxRegistroId.equals("")) {
        return "";
    }
    return auxRegistroId;
}

```

```

public static String obtenerTokenGCM(Context context) throws Exception {
    InstanceID instanceID = InstanceID.getInstance(context);
    String token = instanceID.getToken(SENDER_ID,
        GoogleCloudMessaging.INSTANCE_ID_SCOPE, null);
    return token;
}

```

```

public static class registrarGCM extends AsyncTask<Context, Integer,
    String> {
    Context context;
    protected String doInBackground(Context... params) {
        context=params[0];
        String msg = "";
        Boolean registroWeb = false;
        try {
            idRegistro = obtenerTokenGCM(context);
            registroWeb =
                registrarDispositivoEnServidorWeb(context, idRegistro);
            if (registroWeb == true) {
                guardarIdRegistro(context, idRegistro);
                msg = "Dispositivo registrado, idRegistro=" + idRegistro;
            }
        } catch (Exception ex) {
            msg = "Error registro CGM:" + ex.getMessage();
        }
        return msg;
    }

    protected void onPostExecute(String message){
        mostrarMensaje(context,message);
    }
}

```

```

public static Boolean registrarDispositivoEnServidorWeb(Context context,
    final String regId) {
    try {
        Uri.Builder constructorParametros = new Uri.Builder()
            .appendQueryParameter("iddevice", regId)
            .appendQueryParameter("idapp", SENDER_ID);
        String parametros = constructorParametros.build().getEncodedQuery();
        String url = SERVER_URL + "registrar.php";
        URL direccion = new URL(url);
        HttpURLConnection conexion = (HttpURLConnection)

```

```

        direccion.openConnection();
        conexion.setRequestMethod("POST");
        conexion.setRequestProperty("Accept-Language", "UTF-8");
        conexion.setDoOutput(true);
        OutputStreamWriter outputStreamWriter = new
            OutputStreamWriter(conexion.getOutputStream());
        outputStreamWriter.write(parametros.toString());
        outputStreamWriter.flush();
        int respuesta = conexion.getResponseCode();
        if (respuesta==200){
            return true;
        } else {
            return false;
        }
    } catch (IOException e) {
        return false;
    }
}

```

```

public static void guardarIdRegistro(Context context, String regId) {
    final SharedPreferences prefs = context.getSharedPreferences(
        ActividadPrincipal.class.getSimpleName(), Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = prefs.edit();
    editor.putString("idRegistro", regId);
    editor.commit();
}

```

#### 5.1.4. Desregistrar dispositivo en GCM

```

public void desregistrarUsuarioGCM(View v) {
    desregistrarGCM tarea = new desregistrarGCM();
    tarea.execute(getApplicationContext(), null, null);
}

```

```

public static Boolean eliminarTokenGCM(Context context) {
    try {
        InstanceID instanceID = InstanceID.getInstance(context);
        instanceID.deleteToken(SENDER_ID,
            GoogleCloudMessaging.INSTANCE_ID_SCOPE);
        return true;
    } catch (Exception ex) {
        return false;
    }
}

public static class desregistrarGCM extends AsyncTask<Context, Integer,
    String> {
    Context context;
    protected String doInBackground(Context... params) {
        context=params[0];
        Boolean desregistroWeb = false;
    }
}

```



```

String msg = "";
String auxIdRegistro = "";
try {
    auxIdRegistro = dameIdRegistro(context);
    if (auxIdRegistro.equals("")) {
        msg = "El dispositivo no está registrado";
    } else {
        if (eliminarTokenGCM(context)) {
            desregistroWeb = desregistrarDispositivoEnServidorWeb(
                context, auxIdRegistro);
            if (desregistroWeb == true) {
                guardarIdRegistro(context, "");
                msg = "Dispositivo desregistrado";
            } else {
                msg = "Error al desregistrar el dispositivo en el
                    servidor web.";
            }
        } else {
            msg = "Error al desregistrar el dispositivo en GCM.";
        }
    }
} catch (Exception ex) {
    msg = "Error al desregistrar:" + ex.getMessage();
}
return msg;
}

protected void onPostExecute(String message) {
    mostrarMensaje(context, message);
}
}

public static Boolean desregistrarDispositivoEnServidorWeb(
    Context context, final String regId) {
    try {
        Uri.Builder constructorParametros = new Uri.Builder()
            .appendQueryParameter("iddevice", regId)
            .appendQueryParameter("idapp", SENDER_ID);
        String parametros = constructorParametros.build().getEncodedQuery();
        String url = SERVER_URL + "desregistrar.php";
        URL direccion = new URL(url);
        HttpURLConnection conexion = (HttpURLConnection)
            direccion.openConnection();
        conexion.setRequestMethod("POST");
        conexion.setRequestProperty("Accept-Language", "UTF-8");
        conexion.setDoOutput(true);
        OutputStreamWriter outputStreamWriter = new
            OutputStreamWriter(conexion.getOutputStream());
        outputStreamWriter.write(parametros.toString());
        outputStreamWriter.flush();
        int respuesta = conexion.getResponseCode();
        if (respuesta==200){
            mostrarMensaje(context, "Desregistrado en servidor web");
            return true;
        }
    }
}

```

```

    } else {
        return false;
    }
} catch (IOException e) {
    return false;
}
}

```

### 5.1.5. Actualización del identificador desde GCM

```

import org.example.appgcm.UtilidadesGCM.*;

public class IDListenerService extends InstanceIDListenerService {
    @Override
    public void onTokenRefresh() {
        String auxIdRegistro = "";
        Boolean desregistroWeb = false;
        auxIdRegistro =
            UtilidadesGCM.dameIdRegistro(getApplicationContext());
        if (!auxIdRegistro.equals("")) {
            desregistroWeb =
                UtilidadesGCM
                    .desregistrarDispositivoEnServidorWeb(
                        getApplicationContext(), auxIdRegistro);
            if (desregistroWeb == true) {
                registrarGCM tareaRegistroGCM = new
                    UtilidadesGCM.registrarGCM();
                tareaRegistroGCM.execute(null, null, null);
            }
        }
    }
}

```

### 5.1.6. Recepción de mensajes CGM en el dispositivo

```

import static org.example.appgcm.UtilidadesGCM.*;

public class MensajeListenerService extends GcmListenerService {
    @Override
    public void onMessageReceived(String from, Bundle data) {
        String mensaje = data.getString("mensaje");
        mostrarMensaje(getApplicationContext(), mensaje);
    }
}

```

### 5.1.7. Mostrar notificaciones en la barra de estado

```
static public boolean actividadAbierta=false;
```

```

@Override
public void onPause() {

```

```

    super.onPause();
    actividadAbierta = false;
}

```

```

mostrarMensaje(getApplicationContext(), mensaje);

```

```

if (actividadAbierta) {
    mostrarMensaje(getApplicationContext(), mensaje);
} else {
    mostrarAvisoBarraEstado(getApplicationContext(), mensaje);
}

```

```

public static void mostrarAvisoBarraEstado(Context context, String
message) {
    Intent notificationIntent = new Intent(context(getApplicationContext(),
        ActividadPrincipal.class);
    notificationIntent.putExtra("mensaje", message);
    notificationIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP
        | Intent.FLAG_ACTIVITY_SINGLE_TOP);
    Random r = new Random();
    PendingIntent intent = PendingIntent.getActivity(context, r.nextInt(),
        notificationIntent, 0);
    NotificationManager notificationManager = (NotificationManager) context
        .getSystemService(Context.NOTIFICATION_SERVICE);
    Notification notification = new NotificationCompat.Builder(context)
        .setContentTitle(context.getString(R.string.app_name))
        .setContentText(message)
        .setSmallIcon(R.mipmap.ic_launcher)
        .setContentIntent(intent)
        .setWhen(System.currentTimeMillis())
        .setDefaults(Notification.DEFAULT_SOUND |
            Notification.DEFAULT_VIBRATE)
        .setAutoCancel(true).build();
    notificationManager.notify(0, notification);
}

```

```

@Override
protected void onResume() {
    super.onResume();
    actividadAbierta = true;
    comprobarGooglePlayServices();
    if (getIntent().hasExtra("mensaje")) {
        Bundle extras = getIntent().getExtras();
        String mensaje = extras.getString("mensaje");
        mostrarMensaje(ActividadPrincipal.this, mensaje);
        getIntent().removeExtra("mensaje");
    }
}

```

```
@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    setIntent(intent);
}
```

```
<TextView
    android:id="@+id/display"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
```

```
public TextView mDisplay;
```

```
mDisplay = (TextView) findViewById(R.id.display);
```

```
private static Handler manejador = new Handler();
static void mostrarMensaje(final Context context, final String mensaje) {
    ...
}
```

```
static void mostrarMensaje(Context context,String mensaje) {
    Intent intent = new Intent(DISPLAY_MESSAGE_ACTION);
    intent.putExtra("mensaje", mensaje);
    context.sendBroadcast(intent);
}
```

```
registerReceiver(mHandleMessageReceiver,
IntentFilter(DISPLAY_MESSAGE_ACTION));
```

new

```
private final BroadcastReceiver mHandleMessageReceiver = new
BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String nuevoMensaje = intent.getExtras().getString("mensaje");
        mDisplay.append(nuevoMensaje + "\n");
    }
};
```

```
@Override
protected void onDestroy(){
    unregisterReceiver(mHandleMessageReceiver);
    super.onDestroy();
}
```

## 5.2. Aplicación servidor Google Cloud Messaging

```
<?php
//Conexión a la base de datos
mysqli_connect("nombre_servidor", "nombre_usuario", "password") or die
(mysqli_error());
mysqli_select_db("base_datos");
//Insertamos id de registro devuelto por el GCM.
mysql_query("INSERT INTO dispositivos (iddevice) VALUES
('".$_POST["iddevice"]."')") or die(mysqli_error());
mysqli_close();
?>
```

```
<?php
//Conexión a la base de datos
mysqli_connect("nombre_servidor", "nombre_usuario", "password") or die
(mysqli_error());
mysqli_select_db("base_datos");
//Eliminamos el dispositivo basándonos en el id de registro del GCM.
$sql = "DELETE FROM dispositivos WHERE
iddevice='".$_POST["iddevice"]."'" ;
mysqli_query($sql) or die(mysqli_error());
mysqli_close();
?>
```

```
<html>
<head>
</head>
<body>
<form action="notificar.php" name="formulario" method="post">
    Mensaje:<br>
    <textarea name="mensaje" id="mensaje" cols="45"
        rows="5"></textarea><br>
    <input type="submit" name="btnEnviar" id="btnEnviar"
        value="Enviar notificación" />
</form>
</body>
</html>
```

```
<?php
$host = "nombre_servidor";
$user = "nombre_usuario";
$pass = "password";
$database = "base_datos";
//Nombre del paquete de tu aplicación.
$source="org.example.appgcm";

$service="gcm";

//Conexión a la base de datos
```

```

$connection = mysqli_connect ($host, $user, $pass) or die ('Error al
    conectar con el servidor'.mysqli_error());
mysqli_select_db($database) or die ('->>Error seleccionando la base de
    datos'.mysqli_error());

if ( $_POST['mensaje'] != "" ) {
    $message = $_POST['mensaje'];
    //Cambiar por API key de acceso para server del Google Console
    $apiKey = "API_KEY_ACCESO_SERVER";
    $result=mysqli_query("SELECT * FROM dispositivos");
    while($row = mysqli_fetch_assoc ( $result )) {
        //Recuperamos el id de registro del dispositivo en GCM
        $deviceToken = $row['iddevice'];
        //IMPORTANTE: Array con la información que enviará la
        //notificación.
        $data = array(
            'registration_id' => $deviceToken,
            'collapse_key' => 'col_key',
            'data.mensaje' => $message);
        //Fin array mensaje

        //Código para conectar con GCM y enviar notificación.
        // No modificar.
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL,
            "https://android.googleapis.com/gcm/send");
        $headers = array('Authorization:key=' . $apiKey);
        if($headers){
            curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
        }
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
        curl_setopt($ch, CURLOPT_POST, true);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
        $resultado = curl_exec($ch);
        curl_close($ch);
    };
}
?>

```

```

Uri.Builder constructorParametros = new Uri.Builder()
    .appendQueryParameter("iddevice", regId)
    .appendQueryParameter("idapp", SENDER_ID);
String parametros = constructorParametros.build().getEncodedQuery();

String url = SERVER_URL + "registrar.php";
URL direccion = new URL(url);
HttpURLConnection conexion = (HttpURLConnection)
    direccion.openConnection();

```

## 5.3. Almacenamiento en la nube

### 5.3.1. Crear una aplicación Android para Google Drive

#### 5.3.1.1. Extracción de la huella digital (SHA1)

```
keytool -exportcert -alias androiddebugkey -keystore ruta_copiada -list -v
```

Huellas digitales del Certificado:

```
SHA1: 21:45:BD:F6:98:B8:71:50:39:BD:0E:83:F2:06:9B:ED:43:5A:C2:1C
```

#### 5.3.1.2. Autorizar el acceso a Google Drive

```
compile 'com.android.support:appcompat-v7:23.0.1'
compile 'com.google.android.gms:play-services-identity:7.8.0'
compile 'com.google.http-client:google-http-client-gson:1.21.0'
compile 'com.google.api-client:google-api-client-xml:1.21.0'
compile('com.google.api-client:google-api-client-android:1.21.0') {
    exclude group: 'org.apache.httpcomponents'
}
compile('com.google.apis:google-api-services-drive:v3-rev1-1.21.0') {
    exclude group: 'org.apache.httpcomponents'
}
```

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

```
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.USE_CREDENTIALS"/>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Main" >
    <TextView
        android:id="@+id/txtNombreCuenta"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="[Elige una cuenta de Google Drive]" />
    <Button
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="seleccionarCuenta"
        android:text="Seleccionar Cuenta Google Drive" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="hacerFoto"
        android:text="Hacer foto!" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="seleccionarFoto"
        android:text="Seleccionar foto!" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="listarFicheros"
        android:text="Listar!" />
    <TextView
        android:id="@+id/display"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
</LinearLayout>

```

```

public final class UtilidadesDrive {
    static Drive servicio = null;
    static GoogleAccountCredential credencial = null;
    static String nombreCuenta = null;
    static final int PLAY_SERVICES_RESOLUTION_REQUEST = 9000;
    static final String DISPLAY_MESSAGE_ACTION =
        "org.example.googleddriveapp.DISPLAY_MESSAGE";

    private static Handler manejador = new Handler();

    static void mostrarMensaje(final Context context, final String mensaje)
    {
        manejador.post(new Runnable() {
            public void run() {
                Toast.makeText(context, mensaje, Toast.LENGTH_SHORT).show();
            }
        });
    }

    private static Handler carga = new Handler();
    private static ProgressDialog dialogo;

    static void mostrarCarga(final Context context, final String mensaje) {
        carga.post(new Runnable() {
            public void run() {
                dialogo = new ProgressDialog(context);
                dialogo.setMessage(mensaje);
            }
        });
    }
}

```



```

        dialogo.show();
    }
    });
}

static void ocultarCarga(final Context context) {
    carga.post(new Runnable() {
        public void run() {
            dialogo.dismiss();
        }
    });
}

static void mostrarTexto(Context contexto, String mensaje) {
    Intent intent = new Intent(DISPLAY_MESSAGE_ACTION);
    intent.putExtra("mensaje", mensaje);
    contexto.sendBroadcast(intent);
}
}

```

```

import static org.example.googleddriveapp.UtilidadesDrive.*;
public class ActividadPrincipal extends Activity {
    static final int SOLICITUD_SELECCION_CUENTA = 1;
    static final int SOLICITUD_AUTORIZACION = 2;
    static final int SOLICITUD_SELECCIONAR_FOTOGRAFIA = 3;
    static final int SOLICITUD_HACER_FOTOGRAFIA = 4;
    private TextView txtNombreCuenta;
    private static Uri uriFichero;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (comprobarGooglePlayServices()) {
            mostrarMensaje(this,
                "Se ha encontrado Google Play Services válido.");
        } else {
            mostrarMensaje(this,
                "Error Google Play Services: no está instalado o no es
válido.");
            finish();
        }
        credencial =
            GoogleAccountCredential.usingOAuth2(ActividadPrincipal.this,
                Arrays.asList(DriveScopes.DRIVE));
        txtNombreCuenta = (TextView) findViewById(R.id.txtNombreCuenta);
        SharedPreferences prefs = getSharedPreferences("Preferencias",
            Context.MODE_PRIVATE);
        nombreCuenta = prefs.getString("nombreCuenta", null);
        if (nombreCuenta != null) {
            credencial.setSelectedAccountName(nombreCuenta);
            servicio = obtenerServicioDrive(credencial);
            txtNombreCuenta.setText("Cuenta validada: " + nombreCuenta);
        }
    }
}

```

```

    }
}

private boolean comprobarGooglePlayServices() {
    int resultCode = GooglePlayServicesUtil
        .isGooglePlayServicesAvailable(this);
    if (resultCode != ConnectionResult.SUCCESS) {
        if (GooglePlayServicesUtil.isUserRecoverableError(resultCode)) {
            GooglePlayServicesUtil.getErrorDialog(resultCode, this,
                PLAY_SERVICES_RESOLUTION_REQUEST).show();
        } else {
            finish();
        }
        return false;
    }
    return true;
}

public void seleccionarCuenta(View v) {
    nombreCuenta = null;
    PedirCredenciales();
}

private void PedirCredenciales() {
    if (nombreCuenta == null) {
        startActivityForResult(credencial.newChooseAccountIntent(),
            SOLICITUD_SELECCION_CUENTA);
    }
}

@Override
protected void onActivityResult(final int requestCode,
    final int resultCode, final Intent data) {
    switch (requestCode) {
        case SOLICITUD_SELECCION_CUENTA:
            if (resultCode == RESULT_OK && data != null
                && data.getExtras() != null) {
                nombreCuenta = data
                    .getStringExtra(AccountManager.KEY_ACCOUNT_NAME);
                if (nombreCuenta != null) {
                    credencial.setSelectedAccountName(nombreCuenta);
                    servicio = obtenerServicioDrive(credencial);
                    SharedPreferences prefs = getSharedPreferences(
                        "Preferencias", Context.MODE_PRIVATE);
                    SharedPreferences.Editor editor = prefs.edit();
                    editor.putString("nombreCuenta", nombreCuenta);
                    editor.commit();
                    txtNombreCuenta.setText(nombreCuenta);
                } else {
                    txtNombreCuenta
                        .setText("[Elige una cuenta de Google Drive]");
                }
            }
    }
}

```

```

        break;
    case SOLICITUD_HACER_FOTOGRAFIA:
        break;
    case SOLICITUD_SELECCIONAR_FOTOGRAFIA:
        break;
    case SOLICITUD_AUTORIZACION:
        break;
    }
}

private Drive obtenerServicioDrive(GoogleAccountCredential credencial)
{
    return new Drive.Builder(AndroidHttp.newCompatibleTransport(),
        new GsonFactory(), credencial).build();
}
}

```

### 5.3.1.3. Subir ficheros a Google Drive

```
Files().create(File fichero, AbstractInputStreamContent contenido )
```

```
private String idCarpeta="";
```

```
idCarpeta = prefs.getString("idCarpeta", null);
```

```

private void crearCarpetaEnDrive() {
    Thread t = new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                mostrarCarga(ActividadPrincipal.this, "Creando carpeta...");
                File ficheroDrive = new File();
                ficheroDrive.setTitle("Fotografias_Drive");
                ficheroDrive.setMimeType(
                    "application/vnd.google-apps.folder");
                File ficheroSubido =
                    servicio.files().create(ficheroDrive).execute();
                if (ficheroSubido != null) {
                    SharedPreferences prefs =
                        getSharedPreferences("Preferencias",
                            Context.MODE_PRIVATE);
                    SharedPreferences.Editor editor = prefs.edit();
                    editor.putString("idCarpeta", ficheroSubido.getId());
                    editor.commit();
                    idCarpeta=ficheroSubido.getId();
                    mostrarMensaje(ActividadPrincipal.this,

```

```

        "¡Carpeta creada!");
    }
    ocultarCarga(ActividadPrincipal.this);
} catch (UserRecoverableAuthIOException e) {
    ocultarCarga(ActividadPrincipal.this);
    startActivityForResult(e.getIntent(),
        SOLICITUD_AUTORIZACION);
} catch (IOException e) {
    mostrarMensaje(ActividadPrincipal.this, "Error;" +
        e.getMessage());
    ocultarCarga(ActividadPrincipal.this);
    e.printStackTrace();
}
}
});
t.start();
}

```

```

case SOLICITUD_AUTORIZACION:
    if (resultCode == Activity.RESULT_OK) {
        crearCarpetaEnDrive();
    } else {
        mostrarMensaje(this, "El usuario no autoriza usar Google Drive");
    }
    break;

```

```

    crearCarpetaEnDrive();

```

```

public void hacerFoto(View v) {
    if (nombreCuenta == null) {
        mostrarMensaje(this, "Debes seleccionar una cuenta de Google Drive");
    } else {
        String mediaStorageDir =
            Environment.getExternalStoragePublicDirectory(
                Environment.DIRECTORY_PICTURES).getPath();
        String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss",
            Locale.ENGLISH).format(new Date());
        uriFichero = Uri.fromFile(new java.io.File(mediaStorageDir +
            java.io.File.separator + "IMG_" + timeStamp + ".jpg"));
        Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT, uriFichero);
        startActivityForResult(cameraIntent, SOLICITUD_HACER_FOTOGRAFIA);
    }
}

```

```

public void seleccionarFoto(View v) {
    if (nombreCuenta == null) {
        mostrarMensaje(this, "Debes seleccionar una cuenta de Google Drive");
    }
}

```

```

    } else {
        Intent seleccionFotografiaIntent = new Intent();
        seleccionFotografiaIntent.setType("image/*");
        seleccionFotografiaIntent.setAction(Intent.ACTION_PICK);

        startActivityForResult(Intent.createChooser(seleccionFotografiaIntent,
            "Seleccionar fotografía"), SOLICITUD_SELECCIONAR_FOTOGRAFIA);
    }
}

```

```

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.CAMERA" />

```

```

case SOLICITUD_HACER_FOTOGRAFIA:
    if (resultCode == Activity.RESULT_OK) {
        guardarFicheroEnDrive();
    }
    break;

```

```

case SOLICITUD_SELECCIONAR_FOTOGRAFIA:
    if (resultCode == Activity.RESULT_OK) {
        Uri ficheroSeleccionado = data.getData();
        String[] proyeccion = { MediaStore.Images.Media.DATA };
        Cursor cursor = managedQuery(ficheroSeleccionado, proyeccion,
            null, null, null);
        int column_index =
            cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        uriFichero = Uri.fromFile(
            new java.io.File(cursor.getString(column_index)));
        guardarFicheroEnDrive();
    }
    break;

```

```

private void guardarFicheroEnDrive() {
    Thread t = new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                mostrarCarga(ActividadPrincipal.this, "Subiendo imagen...");
                java.io.File ficheroJava = new
                    java.io.File(uriFichero.getPath());
                FileContent contenido = new FileContent("image/jpeg",
                    ficheroJava);
                File ficheroDrive = new File();
                ficheroDrive.setTitle(ficheroJava.getName());
                ficheroDrive.setMimeType("image/jpeg");
                ficheroDrive.setParents(Arrays.asList(
                    new ParentReference().setId(idCarpeta)));
            }
        }
    });
    t.start();
}

```

```

        File ficheroSubido = servicio.files().create(ficheroDrive,
            contenido).execute();
        if (ficheroSubido != null) {
            mostrarMensaje(ActividadPrincipal.this, "¡Foto subida!");
        }
        ocultarCarga(ActividadPrincipal.this);
    } catch (UserRecoverableAuthIOException e) {
        ocultarCarga(ActividadPrincipal.this);
        startActivityForResult(e.getIntent(), SOLICITUD_AUTORIZACION);
    } catch (IOException e) {
        mostrarMensaje(ActividadPrincipal.this, "Error;" + e.getMessage());
        ocultarCarga(ActividadPrincipal.this);
        e.printStackTrace();
    }
}
});
t.start();
}

```

```

public TextView mDisplay;

```

```

registerReceiver(mHandleMessageReceiver, new
    IntentFilter(DISPLAY_MESSAGE_ACTION));
mDisplay = (TextView) findViewById(R.id.display);

```

```

@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    setIntent(intent);
}
private final BroadcastReceiver mHandleMessageReceiver =
    new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String nuevoMensaje = intent.getExtras().getString("mensaje");
            mDisplay.append(nuevoMensaje + "\n");
        }
    };

```

```

public void listarFicheros(View v) {
    if (nombreCuenta == null) {
        mostrarMensaje(this, "Debes seleccionar una cuenta
            de Google Drive");
    } else {
        Thread t = new Thread(new Runnable() {
            @Override
            public void run() {

```

```

try {
    mostrarCarga(ActividadPrincipal.this, "Listando archivos...");
    FileList result = servicio.files().list()
        .setQ("'" + idCarpeta + "' in parents and trashed=false")
        .execute();
    List<File> files = result.getFiles();
    if (files != null) {
        for (File file : files) {
            mostrarTexto(getBaseContext(), file.getName());
        }
    }

    mostrarMensaje(ActividadPrincipal.this,
        "¡Archivos listados!");
    ocultarCarga(ActividadPrincipal.this);
} catch (UserRecoverableAuthIOException e) {
    ocultarCarga(ActividadPrincipal.this);
    startActivityForResult(e.getIntent(),
        SOLICITUD_AUTORIZACION);
} catch (IOException e) {
    mostrarMensaje(ActividadPrincipal.this,
        "Error;" + e.getMessage());
    ocultarCarga(ActividadPrincipal.this);
    e.printStackTrace();
}
}
});
t.start();
}
}

```

## 5.4. Servicio de Backup de Google

### 5.4.1. Declaración del agente de copia de seguridad en Manifest

```

<manifest>

    <application android:label= "@string/app_name"
        android:backupAgent="miAgenteBackup">
        <activity>
            ...
        </activity>
    </application>
</manifest>

```

### 5.4.2. Registro del servicio Android Backup

```

<application android:label="@string/app_name"

```

```
    android:backupAgent="miAgenteBackup">
    ...
    <meta-data android:name="com.google.android.backup.api_key"
        android:value="AEdPqrEAAAAIDaYEVgU6DJnyJdBMU7KLH3kszDXLv_4DIIsEIyQ"
    />
</application>
```

### 5.4.3. BackupAgentHelper

#### 5.4.3.1. Copia de seguridad de SharedPreferences

```
public class miAgenteBackup extends BackupAgentHelper {
    // El nombre del archivo SharedPreferences
    static final String PREFS = "Preferencias";
    // Una clave para identificar unívocamente el conjunto de copia de
    // seguridad de los datos
    static final String PREFS_BACKUP_KEY = "GoogleDrive";
    // Asignar un ayudante y agregarlo al agente de copia de seguridad.
    @Override
    public void onCreate() {
        SharedPreferencesBackupHelper helper =
            new SharedPreferencesBackupHelper(this, PREFS);
        addHelper(PREFS_BACKUP_KEY, helper);
    }
}
```

#### 5.4.3.2. Copia de seguridad de archivos de almacenamiento interno

```
public class miAgenteBackup extends BackupAgentHelper {
    // El nombre de los archivos
    static final String FILE_FOTOGRAFIAS = "fotografías.txt";
    static final String FILE_LUGARES = "lugares.xml";
    // Una clave para identificar unívocamente el conjunto
    // de copia de seguridad de los datos.
    static final String FILES_BACKUP_KEY = "GoogleDrive";
    // Asignar un ayudante y agregarlo al agente de copia de seguridad
    void onCreate() {
        FileBackupHelper helper = new FileBackupHelper(this,
            FILE_FOTOGRAFIAS, FILE_LUGARES);
        addHelper(FILES_BACKUP_KEY, helper);
    }
}
```

### 5.4.4. Un ejemplo paso a paso

```
<application
    android:allowBackup="true"
    android:backupAgent="MisPreferenciasBackupAgent"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
```



```
android:theme="@style/AppTheme" >
```

```
<meta-data
    android:name="com.google.android.backup.api_key"
    android:value="AEdPqrEAAAAIJuQYD05cd9ikBIUDC4oTTIccOHj0Ai9jBTEHCA" />
```

```
public class MisPreferenciasBackupAgent extends BackupAgentHelper {
    static final String PREFERENCES = "Preferencias";
    static final String PREFERENCES_BACKUP_KEY = "GoogleDriveApp";
    @Override
    public void onCreate(){
        SharedPreferencesBackupHelper helper =
            new SharedPreferencesBackupHelper(this, PREFERENCES);
        addHelper(PREFERENCES_BACKUP_KEY, helper);
    }
}
```

```
private BackupManager backupManager;
```

```
backupManager = new BackupManager(this);
```

```
adb shell bmgr enable true
```

```
adb shell bmgr transport
android/com.android.internal.backup.LocalTransport
```

```
adb shell bmgr backup org.example.googleddriveapp
```

```
adb shell bmgr run
```

```
adb uninstall org.example.googleddriveapp
```

```
adb shell bmgr restore org.example.googleddriveapp
```

### 5.4.5. Auto Backup for Apps

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.my.appexample">
    <uses-sdk android:minSdkVersion="23"/>
    <uses-sdk android:targetSdkVersion="23"/>
```

```
<application ...  
    android:fullBackupContent="@xml/mybackupscheme">  
</application>  
...  
</manifest>
```

```
<?xml version="1.0" encoding="utf-8"?>  
<full-backup-content>  
    <exclude domain="file" path="device_info.txt"/>  
</full-backup-content>
```

```
<full-backup-content>  
    <include domain=["file" | "database" | "sharedpref" | "external"  
        | "root"]  
        path="string" />  
    <exclude domain=["file" | "database" | "sharedpref" | "external"  
        | "root"]  
        path="string" />  
</full-backup-content>
```

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    package="com.my.appexample">  
    <uses-sdk android:minSdkVersion="23"/>  
    <uses-sdk android:targetSdkVersion="23"/>  
    <application ...  
        android:allowBackup="false">  
    </application>  
    ...  
</manifest>
```

```
$ adb shell setprop log.tag.BackupXmlParserLogging VERBOSE
```

```
$ adb shell bmgr run
```

```
$ adb shell bmgr fullbackup <PACKAGE>
```

```
$ adb shell bmgr restore <PACKAGE>
```

```
$ adb shell bmgr wipe <TRANSPORT> <PACKAGE>
```

```
$ adb shell bmgr list transports
```



# CAPÍTULO 6.

## Aplicaciones web en Android

Por VICENTE CARBONELL

### 6.1. Introducción a la tecnología web

#### 6.1.1. Una aplicación web de ejemplo: 3 en raya

```
<html>
  <head>
    <link href="estilos.css" rel="stylesheet" type="text/css">
    <script src="funciones.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  </head>
  <body>
    <!-- Pág. Inicio -->
    <div id="inicio" style="width:100%; position:absolute; top:0;
      left:0; visibility:visible;">
      <div class="cabecera">
        <h1 align="center">3 en Raya</h1>
      </div>
      <div style="width:50%; margin-left:auto; margin-right:auto;">
        <table align="center">
          <tr>
            <td><img src=
"http://www.androidcurso.com/images/certificado_upv.jpg"></td>
            <td><button type="button" onClick="iniciarJuego();">
              Jugar</button></td>
          </tr>
          <tr>
            <td align="center"><a href="http://www.androidcurso.com">
              Android Curso</a></td>
            <td><button type="button" onClick="mostrarAyuda();">
              Ayuda</button></td>
          </tr>
        </table>
      </div>
    </div>
  </body>
</html>
```

```

        </table>
    </div>
</div>
<!-- /Pág. Inicio -->
<!-- Pág. Partida -->
<div id="partida" style=
"width:100%; position:absolute; top:0; left:0; visibility:hidden;">
    <div class="cabecera">
        <h1 align="center" id="turno">Turno Jugador 1</h1>
    </div>
    <div id="tabla" align="center">
        <table class="tabla" id="tabla">
            <tbody>
                <tr>
                    <td name="0_0" id="0_0" value=" "
onClick="click_celda('0_0');"></td>
                    <td name="0_1" id="0_1" value=" "
onClick="click_celda('0_1');"></td>
                    <td name="0_2" id="0_2" value=" "
onClick="click_celda('0_2');"></td>
                </tr>
                <tr>
                    <td name="1_0" id="1_0" value=" "
onClick="click_celda('1_0');"></td>
                    <td name="1_1" id="1_1" value=" "
onClick="click_celda('1_1');"></td>
                    <td name="1_2" id="1_2" value=" "
onClick="click_celda('1_2');"></td>
                </tr>
                <tr>
                    <td name="2_0" id="2_0" value=" "
onClick="click_celda('2_0');"></td>
                    <td name="2_1" id="2_1" value=" "
onClick="click_celda('2_1');"></td>
                    <td name="2_2" id="2_2" value=" "
onClick="click_celda('2_2');"></td>
                </tr>
            </tbody>
        </table>
    </div>
    <div>
        <button type="button"
onClick="mostrarInicio();">Inicio</button>
        <button type="button"
onClick="iniciarJuego();">Reiniciar</button>
    </div>
</div>
<!-- /Pág. Partida -->
<!-- Pág. Ayuda -->
<div id="ayuda" style=
"width:100%; position:absolute; top:0; left:0; visibility:hidden;">
    <div class="cabecera">
        <h1 id="turno" align="center">Ayuda</h1>
    </div>

```

```

    </div>
    <div>
        <h3>Reglas:</h3>
        <p>Trata de colocar 3 fichas en línea. Vale ponerlas en
horizontal, vertical o diagonal. Si se rellenan todas las casilla sin que
ningún jugador haya conseguido poner 3 fichas en línea, entonces se
produce un empate.</p>
        <h3>¿Cómo jugar?</h3>
        <p>Empieza el jugador 1, que juega con X, luego le toca tirar
al jugador 2, que juega con O. Cuando se termina la partida puedes volver
a jugar sin ir a la pantalla inicial pulsando el botón reiniciar.</p>
    </div>
    <div>
        <button type="button" onClick="mostrarInicio();">Inicio</button>
    </div>
</div>
<!-- /Pág. Ayuda -->
</body>
</html>

```

```

body {
    font-family: Arial, Helvetica, sans-serif;
}
button {
    background-color:#79bbff;
    -moz-border-radius:6px;
    -webkit-border-radius:6px;
    border-radius:6px;
    border:1px solid #84bbf3;
    display:inline-block;
    color:#ffffff;
    font-family:arial;
    font-size:15px;
    font-weight:bold;
    padding:6px 24px;
    text-decoration:none;
    text-shadow:1px 1px 0px #528ecc;
}
.cabecera {
    margin-left: auto;
    margin-right: auto;
    margin-bottom: 5px;
    background-color: #006699;
    padding-top: 5px;
    padding-right: 15px;
    padding-bottom: 5px;
    padding-left: 15px;
    font-size: 12pt;
    color:#FFF;
}
#turno{
    text-align: center;
    margin-bottom: 30px;
}

```

```
}
.tabla {
    text-align: center;
    border-collapse: collapse;
    border: 1px solid #03476F;
}
.tabla td{
    border: 1px dotted #03476F;
    text-align: center;
}
.tdX{
    border:1px solid #000000;
    color: #369;
    text-align: center;
}
.tdO{
    border:1px solid #000000;
    color: #DF0101;
    text-align: center;
}
```

```
var tamaño = 3;
var turno = "1";
var numJugadas = 0;
var finDelJuego = false;
var nombreJugador1 = "Jugador 1";
var nombreJugador2 = "Jugador 2";

function mostrarInicio() {
    document.getElementById("inicio").style.visibility = 'visible';
    document.getElementById("partida").style.visibility = 'hidden';
    document.getElementById("ayuda").style.visibility = 'hidden';
}

function mostrarPartida() {
    document.getElementById("inicio").style.visibility = 'hidden';
    document.getElementById("partida").style.visibility = 'visible';
    document.getElementById("ayuda").style.visibility = 'hidden';
}

function mostrarAyuda() {
    document.getElementById("inicio").style.visibility = 'hidden';
    document.getElementById("partida").style.visibility = 'hidden';
    document.getElementById("ayuda").style.visibility = 'visible';
}

function iniciarJuego() {
    finDelJuego = false;
    numJugadas = 0;
    turno = "1";
    document.getElementById("turno").innerHTML = "Turno " + nombreJugador1;
    iniciarTablero();
    mostrarPartida();
}
```



```

}

function iniciarTablero() {
    if (document.documentElement.clientWidth <
        document.documentElement.clientHeight) {
        ancho = document.documentElement.clientWidth / 4;
    } else {
        ancho = document.documentElement.clientHeight / 4;
    }
    for ( var i = 0; i < tamano; i++) {
        for ( var j = 0; j < tamano; j++) {
            var cell = document.getElementById(i + "_" + j);
            cell.innerHTML = "";
            cell.style.width = ancho + "px";
            cell.style.height = ancho + "px";
            cell.value = " ";
            cell.className = "";
        }
    }
}

function click_celda(elemento) {
    var casilla = document.getElementById(elemento);
    if (finDelJuego) {
        alert("El juego ya ha terminado. Comienza uno nuevo!");
        return;
    }
    if (casilla.innerHTML != "") {
        alert("Casilla ocupada!");
        return;
    }
    numJugadas++;
    if (turno == "1") {
        casilla.className = "tdX";
        casilla.innerHTML = "X";
        casilla.style.fontSize = (ancho * 0.8) + "px";
        if (buscaGanador('X')) {
            document.getElementById("turno").innerHTML =
                "Fin del Juego: Gana " + nombreJugador1 + "!!";
            alert("Fin del Juego: Gana " + nombreJugador1 + "!!");
            finDelJuego = true;
            return;
        }
    }
    turno = "2";
    if (numJugadas < tamano*tamano) {
        document.getElementById("turno").innerHTML =
            "Turno " + nombreJugador2;
    }
} else {
    casilla.className = "td0";
    casilla.innerHTML = "O";
    casilla.style.fontSize = (ancho * 0.8) + "px";
    if (buscaGanador('O')) {

```

```

        document.getElementById("turno").innerHTML =
            "Fin del Juego: Gana " + nombreJugador2 + "!!";
        alert("Fin del Juego: Gana " + nombreJugador2 + "!!");
        finDelJuego = true;
        return;
    }
    turno = "1";
    document.getElementById("turno").innerHTML =
        "Turno " + nombreJugador1;
}
if (numJugadas >= tamano*tamano) {
    document.getElementById("turno").innerHTML =
        "Fin del Juego: EMPATE!!";
    alert("Fin del Juego: EMPATE!!");
    finDelJuego = true;
    return;
}
}

function casilla(i, j) {
    return document.getElementById(i + '_' + j).innerHTML;
}

function buscaGanador(turno) {
    //verificamos diagonales
    if (casilla(0,0)==turno && casilla(1,1)==turno && casilla(2,2)==turno)
        return true;
    if (casilla(0,2)==turno && casilla(1,1)==turno && casilla(2,0)==turno)
        return true;
    for (n = 0; n < tamano; n++) {
        //verificamos columnas
        if (casilla(n,0)==turno && casilla(n,1)==turno &&
casilla(n,2)==turno)
            return true;
        //verificamos filas
        if (casilla(0,n)==turno && casilla(1,n)==turno &&
casilla(2,n)==turno)
            return true;
    }
}
}

```

### 6.1.2. Aplicación web *online* y *offline*

CACHE MANIFEST

```

index.html
estilos.css
funciones.js
http://www.androidcurso.com/images/certificado_upv.jpg

```

```
<html manifest="3enraya.manifest">
```

## 6.2. Uso de WebView

### 6.2.1. Mostrar contenido web usando una intención

```
package org.example.aplicacionweb;

public class ActividadPrincipal extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Intent intent = new Intent(Intent.ACTION_VIEW);
        Uri uri = Uri.parse("http://cursoandroid.hol.es/appweb/index.html");
        intent.setData(uri);
        startActivity(intent);
    }
}
```

### 6.2.2. Uso de un WebView para mostrar contenido web

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <WebView
        android:id="@+id/webkit"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
</LinearLayout>
```

```
WebView navegador;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    navegador = (WebView) findViewById(R.id.webkit);
    navegador.loadUrl("http://cursoandroid.hol.es/appweb/index.html");
}
```

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
navegador.getSettings().setJavaScriptEnabled(true);
```

```
<uses-permission android:name="android.permission.INTERNET" />
```

### 6.2.2.1. Evitar el reinicio de la actividad

```
<activity
    android:name="org.example.aplicacionweb.ActividadPrincipal"
    android:label="@string/app_name"
    android:configChanges="orientation|keyboardHidden"
    android:screenOrientation="portrait">
```

### 6.2.2.2. Abrir los enlaces en el WebView

```
navegador.setWebViewClient(new WebViewClient() {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url)
    {
        return false;
    }
});
```

### 6.2.2.3. Opciones de inicio

```
navegador.getSettings().setJavaScriptEnabled(true);
navegador.getSettings().setBuiltInZoomControls(false);
```

### 6.2.2.4. Barra de progreso

```
<ProgressBar
    android:id="@+id/barraProgreso"
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
```

```
private ProgressBar barraProgreso;
```

```
barraProgreso = (ProgressBar) findViewById(R.id.barraProgreso);
navegador.setWebChromeClient(new WebChromeClient() {
    @Override
    public void onProgressChanged(WebView view, int progreso) {
        barraProgreso.setProgress(0);
        barraProgreso.setVisibility(View.VISIBLE);
        ActividadPrincipal.this.setProgress(progreso * 1000);
        barraProgreso.incrementProgressBy(progreso);
        if (progreso == 100) {
            barraProgreso.setVisibility(View.GONE);
        }
    }
});
```

```
ProgressDialog dialogo;
```

```
navegador.setWebViewClient(new WebViewClient(){
    @Override
    public void onPageStarted(WebView view, String url, Bitmap favicon) {
        dialogo = new ProgressDialog(ActividadPrincipal.this);
        dialogo.setMessage("Cargando...");
        dialogo.setCancelable(true);
        dialogo.show();
    }

    @Override
    public void onPageFinished(WebView view, String url) {
        dialogo.dismiss();
    }
});
```

### 6.2.2.5. Navegación

```
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <Button
        android:id="@+id/btnDetener"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="detenerCarga"
        android:text="Stop" />
    <Button
        android:id="@+id/btnAnterior"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="irPaginaAnterior"
        android:text="Atrás" />
    <Button
        android:id="@+id/btnSiguiente"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="irPaginaSiguiente"
        android:text="Sig." />
</LinearLayout>
```

```
Button btnDetener, btnAnterior, btnSiguiente;
```

```
btnDetener = (Button) findViewById(R.id.btnDetener);  
btnAnterior = (Button) findViewById(R.id.btnAnterior);  
btnSiguiente = (Button) findViewById(R.id.btnSiguiente);
```

```
public void detenerCarga(View v) {  
    navegador.stopLoading();  
}  
  
public void irPaginaAnterior(View v) {  
    navegador.goBack();  
}  
  
public void irPaginaSiguiente(View v) {  
    navegador.goForward();  
}
```

```
navegador.setWebViewClient(new WebViewClient(){  
    @Override  
    public void onPageStarted(WebView view, String url, Bitmap favicon) {  
        if(!dialogo.isShowing()){  
            dialogo = new ProgressDialog(ActividadPrincipal.this);  
            dialogo.setMessage("Cargando...");  
            dialogo.setCancelable(true);  
            dialogo.show();  
            btnDetener.setEnabled(true);  
        }  
    }  
  
    @Override  
    public void onPageFinished(WebView view, String url) {  
        if(!dialogo.isShowing()){  
            dialogo.dismiss();  
        }  
        btnDetener.setEnabled(false);  
        if (view.canGoBack()) {  
            btnAnterior.setEnabled(true);  
        } else {  
            btnAnterior.setEnabled(false);  
        }  
        if (view.canGoForward()) {  
            btnSiguiente.setEnabled(true);  
        } else {  
            btnSiguiente.setEnabled(false);  
        }  
    }  
});
```

#### 6.2.2.6. Controlar el botón «Volver»

```

@Override
public void onBackPressed() {
    if (navegador.canGoBack()) {
        navegador.goBack();
    }
    else {
        super.onBackPressed();
    }
}

```

### 6.2.2.7. Capturar alertas JavaScript

```

@Override
public boolean onJsAlert(WebView view, String url, String message,
    final JsResult result) {
    new AlertDialog.Builder(ActividadPrincipal.this).setTitle("Mensaje")
        .setMessage(message).setPositiveButton(
            (android.R.string.ok, new AlertDialog.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    result.confirm();
                }
            })).setCancelable(false).create().show();

    return true;
}

```

### 6.2.2.8. Gestión de errores

```

@Override public void onReceivedError(WebView view, int errorCode,
    String description, String failingUrl) {
    AlertDialog.Builder builder =
        new AlertDialog.Builder(ActividadPrincipal.this);
    builder.setMessage(description).setPositiveButton("Aceptar",
        null).setTitle("onReceivedError");
    builder.show();
}

```

### 6.2.2.9. Descargas

```

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

```

```

private class DescargarFichero extends AsyncTask<URL, Integer, Long> {
    private String mensaje;
}

```

```

@Override
protected Long doInBackground(URL... url) {
    String urlDescarga = url[0].toString();
    mensaje = "";
    InputStream inputStream = null;
    try {
        URL direccion = new URL(urlDescarga);
        HttpURLConnection conexion =
            (HttpURLConnection) direccion.openConnection();
        if (conexion.getResponseCode() == HttpURLConnection.HTTP_OK) {
            inputStream = conexion.getInputStream();
            String fileName = android.os.Environment
                .getExternalStorageDirectory().getAbsolutePath() +
                "/descargas";
            File directorio = new File(fileName);
            directorio.mkdirs();
            File file = new File(directorio, urlDescarga.substring(
                urlDescarga.lastIndexOf("/"),
                (urlDescarga.indexOf("?") == -1 ?
                    urlDescarga.length() : urlDescarga.indexOf("?"))));
            FileOutputStream fileOutputStream = new
                FileOutputStream(file);
            byte[] buffer = new byte[1024];
            int len = 0;
            int bytesRead = -1;
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                fileOutputStream.write(buffer, 0, bytesRead);
            }
            fileOutputStream.close();
            inputStream.close();
            mensaje = "Guardado en: " + file.getAbsolutePath();
        } else {
            throw new Exception(conexion.getResponseMessage());
        }
    } catch (Exception ex) {
        mensaje = ex.getClass().getSimpleName() + " " + ex.getMessage();
    }
    finally {
        if (inputStream != null) {
            try {
                inputStream.close();
            } catch (IOException e) {
            }
        }
    }
    return (long) 0;
}

protected void onPostExecute(Long result) {
    AlertDialog.Builder builder = new
        AlertDialog.Builder(ActividadPrincipal.this);
    builder.setTitle("Descarga");
    builder.setMessage(mensaje);
    builder.setCancelable(true);
}

```



```

        builder.create().show();
    }
}

```

```

navegador.setDownloadListener(new DownloadListener() {
    public void onDownloadStart(final String url, String userAgent, String
        contentDisposition, String mimetype, long contentLength) {
        AlertDialog.Builder builder =
            new AlertDialog.Builder(ActividadPrincipal.this);
        builder.setTitle("Descarga");
        builder.setMessage("¿Deseas guardar el archivo?");
        builder.setCancelable(false).setPositiveButton("Aceptar",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    URL urlDescarga;
                    try {
                        urlDescarga = new URL(url);
                        new DescargarFichero().execute(urlDescarga);
                    } catch (MalformedURLException e) {
                        e.printStackTrace();
                    }
                }
            })
        builder.setNegativeButton("Cancelar",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    dialog.cancel();
                }
            })
        builder.create().show();
    }
});

```

### 6.2.2.10. Conectividad

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

```

private boolean comprobarConectividad() {
    ConnectivityManager connectivityManager =
        (ConnectivityManager) this.getSystemService(
            Context.CONNECTIVITY_SERVICE);
    NetworkInfo info = connectivityManager.getActiveNetworkInfo();
    if ((info == null || !info.isConnected() || !info.isAvailable())) {
        Toast.makeText(ActividadPrincipal.this,
            "Oops! No tienes conexión a internet",
            Toast.LENGTH_LONG).show();
        return false;
    }
    return true;
}

```

```
if (comprobarConectividad()) {  
    btnDetener.setEnabled(true);  
} else {  
    btnDetener.setEnabled(false);  
}
```

```
public void irPaginaAnterior(View v) {  
    if (comprobarConectividad()) {  
        navegador.goBack();  
    }  
}
```

```
public void irPaginaSiguiente(View v) {  
    if (comprobarConectividad()) {  
        navegador.goForward();  
    }  
}
```



**Preguntas de repaso:** *Uso de WebView.*

## 6.3. Diseño web en Android

### 6.3.1. Área de visualización y escalado

```
<meta name="viewport"  
    content="  
        height = [pixel_value | device-height] ,  
        width = [pixel_value | device-width ] ,  
        initial-scale = float_value ,  
        minimum-scale = float_value ,  
        maximum-scale = float_value ,  
        user-scalable = [yes | no] ,  
        target-densitydpi = [dpi_value | device-dpi |  
            high-dpi | medium-dpi | low-dpi]  
    " />
```

```
<meta name="viewport" content="width=400" />
```

```
<meta name="viewport" content="width=device-width" />
```

### 6.3.2. Densidad de pantalla del dispositivo

```
<meta name="viewport" content="target-densitydpi=device-dpi, width=device-width" />
```

```
<!DOCTYPE HTML5>
<html lang="es">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1, target-densitydpi=medium-dpi">
```

```
<link rel="stylesheet" media="screen and (-webkit-device-pixel-ratio:1.5)"
href="hdpi.css" />
```

```
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
    WebView.setWebContentsDebuggingEnabled(true);
}
```

## 6.4. Aplicaciones híbridas

```
public class InterfazComunicacion {
    Context mContext;

    InterfazComunicacion(Context c) {
        mContext = c;
    }
    @JavascriptInterface
    public void mensaje(String contenido){
        Toast.makeText(mContext, contenido, Toast.LENGTH_SHORT).show();
    }
}
```

```
final InterfazComunicacion miInterfazJava = new InterfazComunicacion(this);
```

```
navegador.addJavascriptInterface(miInterfazJava, "jsInterfazNativa");
```

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/inicio"
```

```

        android:icon="@android:drawable/ic_menu_save"
        android:title="Inicio">
    </item>
</menu>

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.inicio, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.inicio:
            navegador.loadUrl("javascript:mostrarInicio()");
            break;
    }
    return true;
}

```

```

<item
    android:id="@+id/jugador1"
    android:icon="@android:drawable/ic_menu_info_details"
    android:title="Jug.1">
</item>

```

```

case R.id.jugador1:
    nombreJugador("1");
    break;

```

```

public void nombreJugador(final String jugador) {
    AlertDialog.Builder alert = new
        AlertDialog.Builder(ActividadPrincipal.this);
    alert.setTitle("Nombre jugador" + jugador);
    alert.setMessage("Nombre:");
    final EditText nombre = new EditText(this);
    alert.setView(nombre);
    alert.setPositiveButton("Guardar",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int whichButton) {
                Editable valor = nombre.getText();
                navegador.loadUrl("javascript:cambiaNombreJugador(\"" + jugador +
                    "\",\"" + valor.toString() + "\");");
                SharedPreferences prefs = getSharedPreferences("Preferencias",
                    Context.MODE_PRIVATE);
                SharedPreferences.Editor editor = prefs.edit();
                editor.putString("jugador" + jugador, valor.toString());
                editor.commit();
            }
        }
    );
}

```

```
});
alert.setNegativeButton("Cancelar",
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int whichButton) {
        }
    });
alert.show();
}
```

```
SharedPreferences prefs = getSharedPreferences("Preferencias",
    Context.MODE_PRIVATE);

final String nombreJ1;
nombreJ1 = prefs.getString("jugador1", null);
```

```
if (nombreJ1 != null) {
    navegador.loadUrl("javascript:cambiaNombreJugador(\"1\", \"\" +
        nombreJ1 + "\");");
}
```

```
function cambiaNombreJugador(numJugador, nombreJugador) {
    if (numJugador=="1") {
        nombreJugador1=nombreJugador;
    } else {
        nombreJugador2=nombreJugador;
    }
}
```

## 6.5. Alternativas en la programación independiente de la plataforma para móviles

### 6.5.1. JQuery Mobile

```
<link rel="stylesheet"
    href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css"/>
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
```

#### 6.5.1.1. Crear una página básica

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>Hola mundo</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href=
    "http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css"/>
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<script src=
    "http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js">
</script>
</head>
<body>
<div data-role="page">
    <div data-role="header">
        <h1>Cabecera</h1>
    </div>
    <div data-role="content">
        <p>Contenido</p>
    </div>
</div>
</body>
</html>
```

```
<script>
$(document).bind('mobileinit',function(){
    $.mobile.changePage.defaults.changeHash = false;
    $.mobile.hashListeningEnabled = false;
    $.mobile.pushStateEnabled = false;
});
</script>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <WebView
        android:id="@+id/webkit"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
</LinearLayout>
```

```
WebView navegador;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    navegador = (WebView) findViewById(R.id.webkit);
    navegador.getSettings().setJavaScriptEnabled(true);
    navegador.getSettings().setBuiltInZoomControls(false);
    navegador.loadUrl("file:///android_asset/index.html");
}
```

```
<activity
    android:name="org.example.ejemplojquerymobile.ActividadPrincipal"
    android:label="@string/app_name"
    android:configChanges="orientation|keyboardHidden">
```

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css"/>
    <script src="jquery-1.11.3.min.js"></script>
    <script>
        $(document).bind('mobileinit',function(){
            $.mobile.changePage.defaults.changeHash = false;
            $.mobile.hashListeningEnabled = false;
            $.mobile.pushStateEnabled = false;
        });
    </script>
    <script src="jquery.mobile-1.4.5.min.js"></script>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
```

### 6.5.1.2. Crear una lista

```
<ul data-role="listview" data-inset="true" data-filter="true">
    <li><a href="#">Samsung</a></li>
    <li><a href="#">Apple</a></li>
    <li><a href="#">Blackberry</a></li>
    <li><a href="#">HTC</a></li>
    <li><a href="#">Nokia</a></li>
</ul>
```

```
<WebView
    android:id="@+id/display"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
```

```
static WebView mDisplay;
```

```
mDisplay = (WebView) findViewById(R.id.display);
mDisplay.getSettings().setJavaScriptEnabled(true);
mDisplay.getSettings().setBuiltInZoomControls(false);
mDisplay.loadUrl("file:///android_asset/index.html");
```

```
<!DOCTYPE html>
```

```

<html>
<head>
<meta name="viewport" content="user-scalable=no,width=device-width"
/>
<link rel="stylesheet"
href=
"http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css"
/>
<script src="funciones.js"></script>
<script src="http://code.jquery.com/jquery-1.11.3.min.js"></script>
<script
src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
</head>

<body>
<div data-role="page">
<div data-role="content">
<ul id="lista" data-role="listview" data-inset="true"></ul>
</div>
</div>
</body>
</html>

```

```

function add(fichero, imagen){
    var listItem = '<li>' +
fichero + '</li>';
    $("#lista").append(listItem);
    $('#lista').listview('refresh');
}

function vaciar(){
    $("#lista").empty();
    $('#lista').listview('refresh');
}

```

```

static void addItem(final Context context, final String fichero,
final String imagen) {
    carga.post(new Runnable() {
        public void run() {
            mDisplay.loadUrl(
                "javascript:add(\"" + fichero + "\",\"" + imagen + "\");");
        }
    });
}

static void vaciarLista(final Context context) {
    carga.post(new Runnable() {
        public void run() {
            mDisplay.loadUrl("javascript:vaciar()");
        }
    });
}

```



```
vaciarLista(getBaseContext());
```

```
mostrarTexto(getBaseContext(), fichero.getOriginalFilename());
```

```
addItem(ActividadPrincipal.this, fichero.getOriginalFilename(),  
fichero.getThumbnailLink());
```

### 6.5.1.3. Añadir un deslizador

```
<form>  
  <label for="slider-0">Valora de 0 a 100:</label>  
  <input type="range" name="slider" id="slider-0" value="50"  
    min="0" max="100"/>  
</form>
```

### 6.5.1.4. Crear un botón

```
<a href="http://www.androidcurso.com" data-role="button" data-icon="star">  
Curso Android</a>
```

### 6.5.1.5. Temas

```
<div data-role="page"> <!-- página -->
```

```
<div data-role="page" data-theme="b"> <!-- página -->
```

```
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <link rel="stylesheet" href="jquery.mobile-1.4.5.min.css"/>  
  <script src="jquery-1.11.3.min.js"></script>  
  <script>  
    $(document).bind('mobileinit',function(){  
      $.mobile.changePage.defaults.changeHash = false;  
      $.mobile.hashListeningEnabled = false;  
      $.mobile.pushStateEnabled = false;  
    });  
  </script>  
  <script src="jquery.mobile-1.4.5.min.js"></script>  
  <link href="estilos.css" rel="stylesheet" type="text/css">  
  <script src="funciones.js"></script>
```

```
</head>
```

```
<!-- Pág. Inicio -->
<div data-role="page" data-theme="b" id="inicio">
  <div data-role="header">
    <h1>3 en Raya</h1>
  </div>
  <div data-role="content">
    <a href="#partida" data-role="button" data-transition="pop"
      onClick="iniciarJuego();">Jugar</a>
    <a href="#ayuda" data-role="button"
      data-display="overlay">Ayuda</a>
  </div>
  <div data-role="footer" data-position="fixed">
    <h3>DEU UPV</h3>
  </div>
</div>
<!-- /Pág. Inicio -->
```

```
<!-- Pág. Partida -->
<div data-role="page" data-theme="b" id="partida">
  <div data-role="header">
    <a href="#inicio" data-icon="home">Inicio</a>
    <h1>3 en Raya</h1>
    <a href="#" data-icon="refresh" class="ui-btn-active"
      onClick="iniciarJuego();">Reiniciar</a>
  </div>
  <div data-role="content">
    <div id="tabla" align="center" style="padding:10px">
      <table class="tabla" id="tabla">
        <tbody>
          <tr>
            <td name="0_0" id="0_0" value=" "
              onClick="click_celda('0_0');"></td>
            <td name="0_1" id="0_1" value=" "
              onClick="click_celda('0_1');"></td>
            <td name="0_2" id="0_2" value=" "
              onClick="click_celda('0_2');"></td>
          </tr>
          <tr>
            <td name="1_0" id="1_0" value=" "
              onClick="click_celda('1_0');"></td>
            <td name="1_1" id="1_1" value=" "
              onClick="click_celda('1_1');"></td>
            <td name="1_2" id="1_2" value=" "
              onClick="click_celda('1_2');"></td>
          </tr>
          <tr>
            <td name="2_0" id="2_0" value=" "
              onClick="click_celda('2_0');"></td>
            <td name="2_1" id="2_1" value=" "
              onClick="click_celda('2_1');"></td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
```

```

        <td name="2_2" id="2_2" value=" "
            onClick="click_celda('2_2');"></td>
    </tr>
</tbody>
</table>
</div>
</div>

<div data-role="footer" data-position="fixed">
    <h1 id="turno">Turno Jugador 1</h1>
</div>
</div>
<!-- /Pág. Partida -->

```

```

</div>
<!-- /Pág. Inicio -->

```

```

<!-- Pág. Ayuda -->
<div data-role="panel" data-theme="b" id="ayuda">
    <h3>Reglas:</h3>
    <p>Trata de colocar 3 fichas en línea. Vale ponerlas en horizontal,
    vertical o diagonal. Si se rellenan todas las casillas sin que ningún
    jugador haya conseguido poner 3 fichas en línea, entonces se produce un
    empate.</p>
    <h3>¿Cómo jugar?</h3>
    <p>Empieza el jugador 1, que juega con X, luego le toca tirar al
    jugador 2, que juega con O. Cuando se termina la partida puedes volver a
    jugar sin ir a la pantalla inicial pulsando el botón reiniciar.</p>
    <a href="#ayuda" data-role="button" data-display="overlay">Cerrar</a>
</div>
<!-- /Pág. Ayuda -->

```

```

function mostrarInicio(){
    $.mobile.changePage("#inicio");
}

```



# CAPÍTULO 7.

## Programación en código nativo

Por MIGUEL GARCÍA PINEDA

### 7.1. Instalación de Android NDK

#### 7.1.1. Un primer ejemplo con Android NDK

```
defaultConfig {  
    applicationId "com.example.SanAngeles"  
    minSdkVersion 4  
    targetSdkVersion 4  
  
    ndk {  
        moduleName "sanangeles"  
    }  
  
    sourceSets.main {  
        jni.srcDirs = []  
        jniLibs.srcDir "src/main/libs"  
    }  
}
```

### 7.2. Funcionamiento y estructura de Android NDK

```
static {
```

```
System.LoadLibrary ("Fichero");  
}
```

### 7.2.1.1. Fichero *Android.mk*

```
LOCAL_PATH := $(call my-dir)  
  
include $(CLEAR_VARS)  
  
LOCAL_MODULE := sanangeles  
  
LOCAL_CFLAGS := -DANDROID_NDK \  
                -DDISABLE_IMPORTGL  
  
LOCAL_SRC_FILES := \  
    importgl.c \  
    demo.c \  
    app-android.c \  
  
LOCAL_LDLIBS := -lGLSv1_CM -ldl -llog  
  
include $(BUILD_SHARED_LIBRARY)
```

```
LOCAL_PATH := $(call my-dir)
```

```
include $(CLEAR_VARS)
```

```
LOCAL_MODULE := sanangeles
```

```
LOCAL_CFLAGS := -DANDROID_NDK \  
                -DDISABLE_IMPORTGL
```

```
LOCAL_SRC_FILES := \  
    importgl.c \  
    demo.c \  
    app-android.c \  

```

```
LOCAL_LDLIBS := -lGLSv1_CM -ldl -llog
```

```
include $(BUILD_SHARED_LIBRARY)
```

### 7.2.1.2. Fichero *Application.mk* (opcional)

```
APP_ABI := all
APP_PLATFORM := android-14
```

### 7.2.1.3. NDKLa herramienta *ndk-build*

```
cd $DIR_JNI_PROYECTO
ndk-build
```

```
ndk-build.cmd NDK_APPLICATION_MK=Application.mk
```

## 7.3. Interfaz entre JAVA y C/C++ (JNI)

### 7.3.1. Tipos fundamentales, referencias y *arrays*

```
const jbyte* GetStringUTFChars(JNIEnv* env, jstring string, jboolean*
                                                                    isCopy);
```

```
const jbyte* str = (*env)->GetStringUTFChars(env, text, NULL);
if (str==NULL)
    return NULL;
```

```
void ReleaseStringUTFChars(JNIEnv* env, jstring string, const char*
                                                                    utf_buffer);
```

```
const jchar* GetStringChars(JNIEnv* env, jstring string, jboolean*
                                                                    isCopy);
void ReleaseStringChars(JNIEnv* env, jstring string, const jchar*
                                                                    chars_buffer);
```

```
jstring NewStringUTF(JNIEnv* env, const char* bytes);
```

```
jstring NewString(JNIEnv* env, const jchar* ubuffer, jsize length);
```

```
jsize GetArrayLength(JNIEnv* env, jarray array);
```

```
jtype* GetTypeArrayElements(JNIEnv* env, jtypeArray array, jtype*  
                             isCopy);
```

```
jtypeArray NewTypeArray(JNIEnv* env, jsize length);
```

### 7.3.2. Desarrollo paso a paso de un programa mediante JNI (I)

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    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"  
    tools:context=".HolaMundoNDK" >  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/hello_world" />  
</RelativeLayout>
```

#### 7.3.2.1. Declaración del método nativo y creación del archivo *Android.mk*

```
static {  
    System.LoadLibrary("libreria");  
}  
public native String nombreMetodo();
```

```
public class HolaMundoNDK extends Activity {  
    static {  
        System.LoadLibrary ("holamundondk");  
    }  
    public native String dameDatos();  
  
    @Override protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_hola_mundo_ndk);  
        setTitle(dameDatos());  
    }  
}
```



```
}

```

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE := holamundondk
LOCAL_SRC_FILES := com_holamundondk_HolaMundoNDK.c

include $(BUILD_SHARED_LIBRARY)

```

### 7.3.2.2. Creación del fichero de cabecera nativo

```
defaultConfig {
    ...

    ndk {
        moduleName "holamundondk"
    }

    sourceSets.main {
        jni.srcDirs = []
        jniLibs.srcDir "src/main/libs"
    }
}

```

### 7.3.2.3. Implementación del método nativo

```
#include "com_holamundondk_HolaMundoNDK.h"
JNIEXPORT jstring Java_com_holamundondk_HolaMundoNDK_dameDatos (JNIEnv *
env, jobject this) {
    return (*env)->NewStringUTF(env, "App nativa");
}

```

## 7.3.3. Acceso a métodos Java desde código nativo (JNI *callback*)

### 7.3.3.1. Métodos de instancia

```
jmethodID GetMethodID(JNIEnv* env, jclass class, const char* name,
                        const char* signature);

```

```
void CallVoidMethod(JNIEnv* env, jobject object, jmethodID methodID, ...);
jboolean CallBooleanMethod(JNIEnv* env, jobject object,
                           jmethodID methodID, ...);
jbyte CallByteMethod(JNIEnv* env, jobject object, jmethodID methodID,
                    ...);
jshort CallShortMethod(JNIEnv* env, jobject object, jmethodID methodID,
                      ...);
jchar CallCharMethod(JNIEnv* env, jobject object, jmethodID methodID,
                    ...);
jint CallIntMethod(JNIEnv* env, jobject object, jmethodID methodID,
                  ...);
jlong CallLongMethod(JNIEnv* env, jobject object, jmethodID methodID,
                    ...);
jfloat CallFloatMethod(JNIEnv* env, jobject object, jmethodID methodID,
                      ...);
jdouble CallDoubleMethod(JNIEnv* env, jobject object, jmethodID methodID,
                         ...);
jobject CallObjectMethod(JNIEnv* env, jobject object, jmethodID methodID,
                        ...);
```

### 7.3.3.2. Métodos de clase

```
jmethodID GetStaticMethodID(JNIEnv* env, jclass class, const char* name,
                           const char* signature);
```

```
void CallStaticVoidMethod(JNIEnv* env, jclass class, jmethodID methodID,
                          ...);
jbyte CallStaticByteMethod(JNIEnv* env, jclass class, jmethodID methodID,
                           ...);
jshort CallStaticShortMethod(JNIEnv* env, jclass class, jmethodID
                             methodID, ...);
jchar CallStaticCharMethod(JNIEnv* env, jclass class, jmethodID methodID,
                           ...);
jint CallStaticIntMethod(JNIEnv* env, jclass class, jmethodID methodID,
                         ...);
jlong CallStaticLongMethod(JNIEnv* env, jclass class, jmethodID methodID,
                           ...);
jfloat CallStaticFloatMethod(JNIEnv* env, jclass class, methodID methodID,
                             ...);
jdouble CallStaticDoubleMethod(JNIEnv* env, jclass class, methodID
                               methodID, ...);
jobject CallStaticObjectMethod(JNIEnv* env, jclass class, methodID
                               methodID, ...);
```

### 7.3.3.3. Invocar constructores

```

...
<Button
    android:id="@+id/button1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_below="@+id/textView1"
    android:layout_marginTop="88dp"
    android:onClick="button1"
    android:text="@string/button1" />
<Button
    android:id="@+id/button0"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/button1"
    android:layout_alignLeft="@+id/button1"
    android:layout_marginBottom="42dp"
    android:onClick="button0"
    android:text="@string/button0" />
<TextView
    android:id="@+id/output"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/button0"
    android:layout_alignLeft="@+id/button0"
    android:gravity="center" />
...

```

```

public class HolaMundoNDK extends Activity implements Runnable {
    private TextView salida;
    private Handler handler;
    public native String dameDatos();
    public native String funcion1(String message);
    public native void funcion2();

    static {
        System.loadLibrary ("holamundondk");
    }

    @Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hola_mundo_ndk);
        setTitle(dameDatos());
        salida = (TextView)super.findViewById(R.id.output);
        this.handler = new Handler();
    }

    @Override public void onResume() {
        super.onResume();
    }
}

```

```
        this.handler.post(this);
    }
    @Override public void onPause() {
        super.onPause();
        this.handler.removeCallbacks(this);
    }

    public void button0(View v){
        salida.setText(funcion1("testString"));
    }

    public void button1(View v){
        funcion2();
    }

    public void funcion3Callback() {
        String message =
            "funcion3Callback llamada por la funcion2 nativa";
        salida.setText(message);
    }

    public void run() {}
}
```

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE := holamundondk
LOCAL_SRC_FILES := com_holamundondk_HolaMundoNDK.c
LOCAL_LDLIBS := -llog
LOCAL_CFLAGS := -Werror

include $(BUILD_SHARED_LIBRARY)
```

```
defaultConfig {
    ...

    ndk {
        moduleName "holamundondk"
    }

    sourceSets.main {
        jni.srcDirs = []
        jniLibs.srcDir "src/main/libs"
    }
}
```

```
/*
 * Class:      com_holamundondk_HolaMundoNDK
```

```

* Method:    funcion1
* Signature: (Ljava/lang/String;)Ljava/lang/String;
*/
JNIEXPORT jstring JNICALL Java_com_holamundondk_HolaMundoNDK_funcion1
    (JNIEnv *, jobject, jstring);
/*
* Class:      com_holamundondk_HolaMundoNDK
* Method:     funcion2
* Signature:  ()V
*/
JNIEXPORT void JNICALL Java_com_holamundondk_HolaMundoNDK_funcion2
    (JNIEnv *, jobject);

```

```

#include "com_holamundondk_HolaMundoNDK.h"
#include <android/log.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define LOG_TAG    "HolaMundoNDK"
#define LOGI(...)  __android_log_print(ANDROID_LOG_INFO, LOG_TAG, __VA_ARGS__)
#define LOGE(...)  __android_log_print(ANDROID_LOG_ERROR, LOG_TAG, __VA_ARGS__)

JNIEXPORT jstring JNICALL Java_com_holamundondk_HolaMundoNDK_dameDatos (JNIEnv *
                                                                    env, jobject thiz) {
    return (*env)->NewStringUTF(env, "App nativa");
}

jstring JNICALL Java_com_holamundondk_HolaMundoNDK_funcion1(JNIEnv* env,
                                                            jobject thiz, jstring message) {
    const char *nativeString = (*env)->GetStringUTFChars(env, message, 0);
    LOGI("funcion1 llamada! Parametro entrante: %s", nativeString);
    (*env)->ReleaseStringUTFChars(env, message, nativeString);
    return (*env)->NewStringUTF(env, "Llamada nativa JNI realizada!");
}

void JNICALL Java_com_holamundondk_HolaMundoNDK_funcion2(JNIEnv* env,
                                                            jobject thiz) {
    LOGI("funcion2 llamada!");
    jclass clazz = (*env)->GetObjectClass(env, thiz);
    if (!clazz) {
        LOGE("callback_handler: FALLO object Class");
        goto failure;
    }
    jmethodID method = (*env)->GetMethodID(env, clazz, "funcion3Callback",
                                                                    "()V");

    if (!method) {
        LOGE("callback_handler: FALLO metodo ID");
        goto failure;
    }
    (*env)->CallVoidMethod(env, thiz, method);
}

```

```
failure: return;  
}
```

## 7.4. Rendimiento de aplicaciones con código nativo

```
<?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"  
    android:gravity="center_horizontal">  
    <TextView  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:text="Dalvik vs. Nativo"  
        android:gravity="center"  
        android:textSize="40sp"  
        android:layout_margin="10dp" />  
    <LinearLayout  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content">  
        <EditText  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:id="@+id/ValorEntrante"  
            android:hint="Valor"  
            android:textSize="30sp"  
            android:inputType="number"  
            android:layout_margin="10dp">  
        </EditText>  
        <Button  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:id="@+id/botonLanzar"  
            android:text="Lanzar"  
            android:textSize="30sp"  
            android:layout_margin="10sp">  
        </Button>  
    </LinearLayout>  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:id="@+id/Resultado"  
        android:text="RESULTADO"  
        android:textSize="20sp">  
    </TextView>  
</LinearLayout>
```

```

public class FibonacciNDK extends Activity implements OnClickListener {
    TextView Resultado;
    Button botonLanzar;
    EditText ValorEntrante;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        ValorEntrante = (EditText) findViewById(R.id.ValorEntrante);
        Resultado = (TextView) findViewById(R.id.Resultado);
        botonLanzar = (Button) findViewById(R.id.botonLanzar);
        botonLanzar.setOnClickListener(this);
    }

    public static long fibonacciDalvikR(long n) {
        if (n <= 0)
            return 0;
        if (n == 1)
            return 1;
        return fibonacciDalvikR(n - 1) + fibonacciDalvikR(n - 2);
    }

    public static long fibonacciDalvikI(long n) {
        long previous = -1;
        long result = 1;
        for (long i = 0; i <= n; i++) {
            long sum = result + previous;
            previous = result;
            result = sum;
        }
        return result;
    }

    static {
        System.loadLibrary("fibonacci");
    }

    public static native long fibonacciNativoR(int n);

    public static native long fibonacciNativoI(int n);

    public void onClick(View view) {
        int input = Integer.parseInt(ValorEntrante.getText().toString());
        long start1, start2, stop1, stop2;
        long result;
        String out = "";
        // Dalvik - Recursivo
        start1 = System.currentTimeMillis();
        result = fibonacciDalvikR(input);
        stop1 = System.currentTimeMillis();
        out += String.format("Dalvik recursiva - Valor: %d Tiempo: (%d msec)",
                             result, stop1 - start1);
        // Dalvik - Iterativo
        start2 = System.currentTimeMillis();
    }
}

```

```

        result = fibonacciDalvikI(input);
        stop2 = System.currentTimeMillis();
        out += String.format("\nDalvik iterativa-Valor: %d Tiempo: (%d msec)",
                                result, stop2 - start2);

// Nativo - Recursivo
        start1 = System.currentTimeMillis();
        result = fibonacciNativoR(input);
        stop1 = System.currentTimeMillis();
        out += String.format("\nNativo recursivo-Valor: %d Tiempo: (%d msec)",
                                result, stop1 - start1);

// Nativo - Iterativo
        start2 = System.currentTimeMillis();
        result = fibonacciNativoI(input);
        stop2 = System.currentTimeMillis();
        out += String.format("\nNativo iterativo-Valor: %d Tiempo: (%d msec)",
                                result, stop2 - start2);

        Resultado.setText(out);
    }
}

```

```

LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE := fibonacci
LOCAL_SRC_FILES := fibonacci.c

include $(BUILD_SHARED_LIBRARY)

```

```

defaultConfig {
    ...

    ndk {
        moduleName "fibonacci"
    }

    sourceSets.main {
        jni.srcDirs = []
        jniLibs.srcDir "src/main/libs"
    }
}

```

```

...
/*
 * Class:      com_fibonaccindk_FibonacciNDK
 * Method:    fibonacciNativoR
 * Signature: (I)J
 */
JNIEXPORT jlong JNICALL Java_com_fibonaccindk_FibonacciNDK_fibonacciNativoR
    (JNIEnv *, jclass, jint);

```



```

/*
 * Class:      com_fibonaccindk_FibonacciNDK
 * Method:     fibonacciNativoI
 * Signature: (I)J
 */
JNIEXPORT jlong JNICALL Java_com_fibonaccindk_FibonacciNDK_fibonacciNativoI
    (JNIEnv *, jclass, jint);

#ifdef __cplusplus
}
#endif
#endif

```

```

#include "com_fibonaccindk_FibonacciNDK.h"
jint fibonacciNativoR(jint n) {
    if(n<=0) return 0;
    if(n==1) return 1;
    return fibonacciNativoR(n-1) + fibonacciNativoR(n-2);
}

jint fibonacciNativoI(jint n) {
    jint previous = -1;
    jint result = 1;
    jint i=0;
    jint sum=0;
    for (i = 0; i <= n; i++) {
        sum = result + previous;
        previous = result;
        result = sum;
    }
    return result;
}

JNIEXPORT jlong JNICALL Java_com_fibonaccindk_FibonacciNDK_fibonacciNativoR
    (JNIEnv *env, jclass obj, jint n) {
    return fibonacciNativoR(n);
}

JNIEXPORT jlong JNICALL Java_com_fibonaccindk_FibonacciNDK_fibonacciNativoI
    (JNIEnv *env, jclass obj, jint n) {
    return fibonacciNativoI(n);
}

```

## 7.5. Procesado de imagen con código nativo

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"

```

```

        android:orientation="vertical" >
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:scaleType="centerCrop"
            android:layout_gravity="center_vertical|center_horizontal"
            android:id="@+id/ivDisplay"/>
        <LinearLayout
            android:orientation="horizontal"
            android:gravity="center"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >
            <Button
                android:id="@+id/btnReset"
                style="?android:attr/buttonStyleSmall"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Original"
                android:onClick="onResetImagen" />
            <Button
                android:id="@+id/btnConvert"
                style="?android:attr/buttonStyleSmall"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Grises"
                android:onClick="onConvertirGrises" />
        </LinearLayout>
    </LinearLayout>

```

```

public class ImgProcesadoNDK extends Activity {
    private String tag = "ImgProcesadoNDK";
    private Bitmap bitmapOriginal = null;
    private Bitmap bitmapGrises = null;
    private ImageView ivDisplay = null;

    static {
        System.loadLibrary("imgprocesadondk");
    }

    public native void convertirGrises(Bitmap bitmapIn, Bitmap bitmapOut);

    @Override public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Log.i(tag, "Imagen antes de modificar");
        ivDisplay = (ImageView) findViewById(R.id.ivDisplay);
        BitmapFactory.Options options = new BitmapFactory.Options();
        // Asegurar que la imagen tiene 24 bits de color
        options.inPreferredConfig = Config.ARGB_8888;
        bitmapOriginal = BitmapFactory.decodeResource(this.getResources(),
            R.drawable.sampleimage, options);

        if (bitmapOriginal != null)
            ivDisplay.setImageBitmap(bitmapOriginal);
    }
}

```

```

public void onResetImagen(View v) {
    Log.i(tag, "Resetea Imagen");
    ivDisplay.setImageBitmap(bitmapOriginal);
}

public void onConvertirGrises(View v) {
    Log.i(tag, "Conversion a escala de grises");
    bitmapGrises = Bitmap.createBitmap(bitmapOriginal.getWidth(),
                                       bitmapOriginal.getHeight(), Config.ARGB_8888);
    convertirGrises(bitmapOriginal, bitmapGrises);
    ivDisplay.setImageBitmap(bitmapGrises);
}
}

```

```

LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)
LOCAL_MODULE := imgprocesadondk
LOCAL_SRC_FILES := imgprocesadondk.c
LOCAL_LDLIBS := -llog -ljnigraphics

include $(BUILD_SHARED_LIBRARY)

```

```
APP_PLATFORM=android-8
```

```

defaultConfig {
    ...

    ndk {
        moduleName "imgprocesadondk"
    }

    sourceSets.main {
        jni.srcDirs = []
        jniLibs.srcDir "src/main/libs"
    }
}

```

```

#include "com_imgprocesadondk_ImgProcesadoNDK.h"
#include <android/log.h>
#include <android/bitmap.h>

#define LOG_TAG "libimgprocesadondk"
#define LOGI(...) __android_log_print(ANDROID_LOG_INFO, LOG_TAG, __VA_ARGS__)
#define LOGE(...) __android_log_print(ANDROID_LOG_ERROR, LOG_TAG, __VA_ARGS__)

typedef struct
{
    uint8_t red;
    uint8_t green;
    uint8_t blue;
}

```

```

    uint8_t alpha;
} rgba;

/*Conversion a grises por pixel*/
JNIEXPORT void JNICALL Java_com_imgprocesadondk_ImgProcesadoNDK_convertirGrises
    (JNIEnv * env, jobject obj, jobject bitmapcolor, jobject bitmapgris)
{
    AndroidBitmapInfo infocolor;
    void* pixelscolor;
    AndroidBitmapInfo infogris;
    void* pixelgris;
    int ret;
    int y;
    int x;
    LOGI("convertirGrises");
    if ((ret = AndroidBitmap_getInfo(env, bitmapcolor, &infocolor)) < 0) {
        LOGE("AndroidBitmap_getInfo() failed ! error=%d", ret);
        return;
    }

    if ((ret = AndroidBitmap_getInfo(env, bitmapgris, &infogris)) < 0) {
        LOGE("AndroidBitmap_getInfo() failed ! error=%d", ret);
        return;
    }

    LOGI("imagen color :: ancho %d;alto %d;avance %d;formato %d;flags %d",
        infocolor.width, infocolor.height, infocolor.stride,
        infocolor.format, infocolor.flags);
    if (infocolor.format != ANDROID_BITMAP_FORMAT_RGBA_8888) {
        LOGE("Bitmap no es formato RGBA_8888 !");
        return;
    }

    LOGI("imagen color :: ancho %d;alto %d;avance %d;formato %d;flags %d",
        infogris.width, infogris.height, infogris.stride,
        infogris.format, infogris.flags);
    if (infogris.format != ANDROID_BITMAP_FORMAT_RGBA_8888) {
        LOGE("Bitmap no es formato RGBA_8888 !");
        return;
    }

    if ((ret = AndroidBitmap_lockPixels(env, bitmapcolor, &pixelscolor))
        < 0) {
        LOGE("AndroidBitmap_lockPixels() failed ! error=%d", ret);
    }

    if ((ret = AndroidBitmap_lockPixels(env, bitmapgris, &pixelgris)) < 0) {
        LOGE("AndroidBitmap_lockPixels() fallo ! error=%d", ret);
    }

    // modificacion pixeles en el algoritmo de escala grises
    for (y=0;y<infocolor.height;y++) {
        rgba * line = (rgba *) pixelscolor;
        rgba * grisline = (rgba *) pixelgris;
        for (x=0;x<infocolor.width;x++) {

```

```
        float output = (line[x].red + line[x].green + line[x].blue)/3;
        if (output > 255) output = 255;
        grisline[x].red = grisline[x].green = grisline[x].blue =
                                (uint8_t) output;
        grisline[x].alpha = line[x].alpha;
    }
    pixelscolor = (char *)pixelscolor + infocolor.stride;
    pixelgris = (char *) pixelgris + infogris.stride;
}
LOGI("unlocking pixels");
AndroidBitmap_unlockPixels(env, bitmapcolor);
AndroidBitmap_unlockPixels(env, bitmapgris);
}
```

```
outputRed = (inputRed * .393) + (inputGreen *.769) + (inputBlue * .189)
outputGreen = (inputRed * .349) + (inputGreen *.686) + (inputBlue * .168)
outputBlue = (inputRed * .272) + (inputGreen *.534) + (inputBlue * .131)
```



# CAPÍTULO 8.

## Redes sociales: Facebook y Twitter

Por JORDI BATALLER MASCARELL

### 8.1. Android y Facebook

#### 8.1.1.1. Configurando nuestra nueva aplicación (en Facebook)

```
keytool -exportcert -alias androiddebugkey -keystore  
%HOMEPATH%\android\debug.keystore | openssl sha1 -binary | openssl  
base64
```

Escriba la contraseña del almacén de claves: android

```
keytool -list -printcert -jarfile miaplicacion.apk | grep "SHA1: " |  
cut -d " " -f 3 | xxd -r -p | openssl base64
```

```
keytool -list -printcert -jarfile [path_to_your_apk] | grep -Po  
"(?<=SHA1:) .*" | xxd -r -p | openssl base64
```

```
PackageInfo info;
```

```
try {
    info = getPackageManager().getPackageInfo("com.you.name", // nombre
paquete
        PackageManager.GET_SIGNATURES);
    for (Signature signature : info.signatures) {
        MessageDigest md;
        md = MessageDigest.getInstance("SHA");
        md.update(signature.toByteArray());
        String something = new String(Base64.encode(md.digest(), 0));
        // Log.d("KeyHash:", Base64.encodeToString(md.digest(), Base64.DEFAULT));

        Log.e("hash key", something);
    }
} catch (NameNotFoundException e1) {
    Log.e("name not found", e1.toString());
} catch (NoSuchAlgorithmException e) {
    Log.e("no such an algorithm", e.toString());
} catch (Exception e) {
    Log.e("exception", e.toString());
}
```

## 8.1.2. Nuestro proyecto Android

```
repositories {
    mavenCentral()
}
```

```
compile 'com.facebook.android:facebook-android-sdk:4.7.0'
```

### 8.1.2.1. Configurando el código de nuestra aplicación

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.cuandrav.pruebafacebooksdk1">

    <!-- 1. añadir estos dos permisos -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission an-
droid:name="android.permission.ACCESS_NETWORK_STATE" />
```



```

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">

    <!-- 2. añadir por Facebook -->
    <meta-data android:name="com.facebook.sdk.ApplicationId" an-
droid:value="@string/facebook_app_id" />

    <!-- nuestra actividad -->
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name"
        android:theme="@style/AppTheme.NoActionBar">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER"
/>

        </intent-filter>
    </activity>

    <!-- 3. añadir por Facebook: para hacer login -->
    <activity android:name="com.facebook.FacebookActivity" />

</application>

</manifest>

```

### 8.1.3. Aplicación de ejemplo (Graph API)

```

com.facebook.login.widget.LoginButton
    android:id="@+id/login_button"
    android:layout_width="wrap_content"
    android:layout_height="121dp"
    facebook:com_facebook_confirm_logout="false"
    facebook:com_facebook_tooltip_mode="never_display"
/>

```

```
FacebookSdk.sdkInitialize(this.getApplicationContext());
```

```
loginButtonOficial.setPublishPermissions("publish_actions");

LoginManager
.getInstance().registerCallback(this.elCallbackManagerDeFacebook,
    new FacebookCallback<LoginResult>() {
        @Override
        public void onSuccess(LoginResult loginResult) { ...

this.elCallbackManagerDeFacebook.onActivityResult(requestCode,
resultCode, data);

Profile profile = Profile.getCurrentProfile();

AccessToken acessToken = AccessToken.getCurrentAccessToken();

LoginManager.getInstance().loginWithPublishPermissions(this,
Arrays.asList("publish_actions"));

LoginManager.getInstance().logout();

Bundle params = new Bundle();
params.putString("message", textoQueEnviar);

GraphRequest request = new GraphRequest(
    AccessToken.getCurrentAccessToken(),
    "/me/feed",
    params,
    HttpMethod.POST,
    new GraphRequest.Callback() {
        public void onCompleted(GraphResponse response) {
            Toast.makeText(THIS, "Publicación realizada: " +
textoQueEnviar, Toast.LENGTH_LONG).show();

        }
    }
);
request.executeAsync();
```

```

Bundle params = new Bundle();
params.putByteArray("source", byteArray); // bytes de la imagen
params.putString("caption", comentario); // comentario

// si se quisiera publicar una imagen de internet: pa-
rams.putString("url",
// "{image-url}");

GraphRequest request = new GraphRequest(
    AccessToken.getCurrentAccessToken(),
    "/me/photos",
    params,
    HttpMethod.POST,
    new GraphRequest.Callback() {
        public void onCompleted(GraphResponse response) {
            Toast.makeText(THIS, "" + byteArray.length + " Foto
enviada: " + response.toString(), Toast.LENGTH_LONG).show();
        }
    }
);

request.executeAsync();

POST graph.facebook.com/{ user-
id}/feed?message={message}&access_token={access-token}

```

```

package org.cuandrav.pruebafacebooksdk1;

//-----
// imports
//-----
import com.facebook.AccessToken;
import com.facebook.CallbackManager;
import com.facebook.FacebookCallback;
import com.facebook.FacebookException;
import com.facebook.FacebookSdk;
import com.facebook.GraphRequest;
import com.facebook.GraphResponse;
import com.facebook.HttpMethod;
import com.facebook.Profile;
import com.facebook.login.LoginManager;
import com.facebook.login.LoginResult;
import com.facebook.login.widget.LoginButton;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;

```

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.util.Log;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONObject;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.Arrays;

//-----
// class MainActivity
//-----
public class MainActivity extends AppCompatActivity {

    // -----
    // elementos gráficos
    // -----
    private TextView elTextoDeBienvenida;
    private Button botonHacerLogin;
    private Button botonLogout;
    private Button botonEnviarFoto;
    private TextView textoConElMensaje;
    private Button botonCompartir;

    //
    // boton oficial de Facebook para login/logout
    //
    LoginButton loginButtonOficial;

    //
    // gestiona los callbacks al FacebookSdk desde el método onActivityResult() de una actividad
    //
    private CallbackManager elCallbackManagerDeFacebook;

    //-----
    // puntero a this para los callback
    //-----
    private final Activity THIS = this;

    //-----
    // onCreate ()
```

```
//-----
@Override
protected void onCreate(Bundle savedInstanceState) {
    //
    // inicializar super
    //
    super.onCreate(savedInstanceState);

    //
    Log.d("cuandrav.onCreate()", " .onCreate() llamado");

    //
    // cosas de Facebook
    //

    //
    //
    // inicializar FacebookSDK
    //
    FacebookSdk.sdkInitialize(this.getApplicationContext());

    //
    // pongo el contenido visual de la actividad (hacer antes que
findViewById (
    // y después de inicializar FacebookSDK)
    //
    this.setContentView(R.layout.activity_main);

    //
    // botón oficial de "login en Facebook"
    //

    // obtengo referencia
    loginButtonOficial = (LoginButton)
findViewById(R.id.login_button);

    // declaro los permisos que debe pedir al ser pulsado
    // ver lista en: https://developers.facebook.com/docs/facebook-
login/permissions
    loginButtonOficial.setPublishPermissions("publish_actions");

    //loginButtonOficial.setReadPermissions("public_profile"); // si
pones uno, no puedes poner el otro

    //
    // crear callback manager de Facebook
    //
    this.elCallbackManagerDeFacebook = CallbackMana-
ger.Factory.create();

    // registro un callback para saber cómo ha ido el login
    LoginMana-
ger.getInstance().registerCallback(this.elCallbackManagerDeFacebook,
```

```

        new FacebookCallback<LoginResult>() {
            @Override
            public void onSuccess(LoginResult loginResult) {
                // App code
                Toast.makeText(THIS, "Login onSuccess()",
                    Toast.LENGTH_LONG).show();
                actualizarVentanita();
            }

            @Override
            public void onCancel() {
                Toast.makeText(THIS, "Login onCancel()",
                    Toast.LENGTH_LONG).show();
                actualizarVentanita();
            }

            @Override
            public void onError(FacebookException exception) {
                // App code
                Toast.makeText(THIS, "Login onError(): " + excep-
                    tion.getMessage(),
                        Toast.LENGTH_LONG).show();
                actualizarVentanita();
            }
        });

        //
        // otras cosas
        //

        //
        // obtengo referencias a mis otros widgets en el layout
        //
        elTextoDeBienvenida=(TextView)
        findViewById(R.id.elTextoDeBienvenida);
        botonHacerLogin = (Button) findViewById(R.id.boton_hacerLogin);
        botonLogOut = (Button) findViewById(R.id.boton_LogOut);
        botonEnviarFoto = (Button) findViewById(R.id.boton_EnviarFoto);
        textoConElMensaje = (TextView) findViewById(R.id.txt_mensajeFB);
        botonCompartir = (Button) findViewById(R.id.boton_EnviarAFB);

        //
        //
        //
        this.actualizarVentanita();

        //
        Log.d("cuandrav.onCreate", "final .onCreate() ");

    } // ()

    // -----

```

```

// -----
@Override
protected void onActivityResult(final int requestCode, final int resultCode, final Intent data) {
    // se llama cuando otra actividad que hemos arrancado termina y nos devuelve el control
    // tal vez, devolviéndonos algun resultado (resultCode, data)
    Log.d("cuandrav.onActivityResultResu", "llamado");

    //
    // avisar a super
    //
    super.onActivityResult(requestCode, resultCode, data);

    //
    // avisar a Facebook (a su callback manager) por si le afecta
    //
    this.elCallbackManagerDeFacebook.onActivityResult(requestCode, resultCode, data);
}

// -----
// -----
private void actualizarVentanita() {

    Log.d("cuandrav.actualizarVent", "empiezo");
    //
    // obtengo el access token para ver si hay sesión
    //
    AccessToken accessToken = this.obtenerAccessToken();

    if (accessToken == null) {

        Log.d("cuandrav.actualizarVent", "no hay sesion, deshabilito");
        //
        // sesion con facebook cerrada
        //
        this.botonHacerLogin.setEnabled(true);
        this.botonLogout.setEnabled(false);
        this.textoConElMensaje.setEnabled(false);
        this.botonCompartir.setEnabled(false);
        this.botonEnviarFoto.setEnabled(false);
        this.elTextoDeBienvenida.setText("haz login");

        return;
    }

    //
    // sí hay sesión
    //

```

```

        Log.d("cuandrav.actualizarVent", "hay sesion habilito");

        this.botonHacerLogin.setEnabled(false);
        this.botonLogOut.setEnabled(true);
        this.textoConElMensaje.setEnabled(true);
        this.botonCompartir.setEnabled(true);
        this.botonEnviarFoto.setEnabled(true);

        //
        // averiguo los datos básicos del usuario acreditado
        //
        Profile profile = Profile.getCurrentProfile();
        if (profile != null) {
            this.textoConElMensaje.setText(profile.getName());
        }

        //
        // otra forma de averiguar los datos básicos:
        // hago una petición con "graph api" para obtener datos del usua-
        rio acreditado
        //
        this.obtenerPublicProfileConRequest_async(
            // como es asíncrono he de dar un callback
            new GraphRequest.GraphJSONObjectCallback() {
                @Override
                public void onCompleted(JSONObject datosJSON, Graph-
Response response) {
                    //
                    // muestro los datos
                    //
                    String nombre= "nombre desconocido";
                    try {
                        nombre = datosJSON.getString("name");
                    } catch (org.json.JSONException ex) { }
                    elTextoDeBienvenida.setText(nombre);
                }
            });

    } // ()

    // -----
    // -----
    private boolean sePuedePublicar() {
        //
        // compruebo la red
        //
        if (!this.hayRed()) {
            Toast.makeText(this, "¿no hay red? No puedo publicar",
Toast.LENGTH_LONG).show();
            return false;
        }

        //

```



```

        // compruebo permisos
        //
        if (! this.tengoPermisoParaPublicar()) {
            Toast.makeText(this, "{no tengo permisos para publicar? Los
            pido.", Toast.LENGTH_LONG).show();
            // pedirPermisoParaPublicar();
            LoginManager.getInstance().loginWithPublishPermissions(this,
            Arrays.asList("publish_actions"));

            return false;
        }

        return true;
    }

    // -----
    // -----
    private AccessToken obtenerAccessToken() {
        return AccessToken.getCurrentAccessToken();
    }

    // -----
    // -----
    private boolean tengoPermisoParaPublicar() {
        AccessToken accessToken = this.obtenerAccessToken();
        return accessToken != null && ac-
        cessToken.getPermissions().contains("publish_actions");
    }

    // -----
    // -----
    private boolean hayRed() {
        ConnectivityManager connectivityManager = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo activeNetworkInfo = connectivityManager
            .getActiveNetworkInfo();
        return activeNetworkInfo != null && activeNetworkIn-
        fo.isConnected();

        // http://stackoverflow.com/questions/15091591/post-on-facebook-
        // wall-without-showing-dialog-on-android
        // comprobar que estamos conetados a internet, antes de hacer el
        login con
        // facebook. Si no: da problemas.
    } // ()

    // -----
    // -----
    public void enviarTextoAFacebook_async (final String textoQueEnviar) {
        //
        // si no se puede publicar no hago nada
        //
        if ( ! sePuedePublicar() ) {

```

```

        return;
    }

    //
    // hago la petición a través del API Graph
    //
    Bundle params = new Bundle();
    params.putString("message", textoQueEnviar);

    GraphRequest request = new GraphRequest(
        AccessToken.getCurrentAccessToken(),
        "/me/feed",
        params,
        HttpMethod.POST,
        new GraphRequest.Callback() {
            public void onCompleted(GraphResponse response) {
                Toast.makeText(THIS, "Publicación realizada: " +
textoQueEnviar, Toast.LENGTH_LONG).show();
            }
        }
    );

    request.executeAsync();

} // ()

// -----
// -----
public void enviarFotoAFacebook_async (Bitmap image, String comen-
tario) {

    //
    // si no se puede publicar no hago nada
    //
    if ( ! sePuedePublicar() ) {
        return;
    }

    //
    // convierto el bitmap a array de bytes
    //
    ByteArrayOutputStream stream = new ByteArrayOutputStream();
    image.compress(Bitmap.CompressFormat.PNG, 100, stream);
    //image.recycle ();
    final byte[] byteArray = stream.toByteArray();
    try {
        stream.close();
    } catch (IOException e) { }

    //
    // hago la petición a través del Graph API

```

```

//
Bundle params = new Bundle();
params.putByteArray("source", byteArray); // bytes de la imagen
params.putString("caption", comentario); // comentario

// si se quisiera publicar una imagen de internet:
params.putString("url", "{image-url}");

GraphRequest request = new GraphRequest(
    AccessToken.getCurrentAccessToken(),
    "/me/photos",
    params,
    HttpMethod.POST,
    new GraphRequest.Callback() {
        public void onCompleted(GraphResponse response) {
            Toast.makeText(THIS, "" + byteArray.length + "
Foto enviada: " + response.toString(), Toast.LENGTH_LONG).show();
            //textoConElMensaje.setText(response.toString());
        }
    }
);

request.executeAsync();

} // ()

// -----
// -----
public void boton_Login_pulsado(View quien) {
    //
    // compruebo la red
    //
    if (!this.hayRed()) {
        Toast.makeText(this, "¿No hay red? No puedo abrir sesión",
            Toast.LENGTH_LONG).show();
    }

    //
    // login
    //
    LoginManager.getInstance().loginWithPublishPermissions(this, Ar-
rays.asList("publish_actions"));

    //
    // actualizar
    //
    this.actualizarVentanita();
} // ()

// -----

```

```
// -----
public void boton_Logout_pulsado(View quien) {
    //
    // compruebo la red
    //
    if (!this.hayRed()) {
        Toast.makeText(this, "¿No hay red? No puedo cerrar sesión",
            Toast.LENGTH_LONG).show();
    }

    //
    // logout
    //
    LoginManager.getInstance().logout();

    //
    // actualizar
    //
    this.actualizarVentanita();
} // ()

// -----
// -----
private void obtenerPublicProfileConRequest_async (GraphRe-
quest.GraphJSONObjectCallback callback) {

    //
    // obtengo access token y compruebo que hay sesión
    //
    AccessToken accessToken = obtenerAccessToken();
    if (accessToken == null) {
        Toast.makeText(THIS, "no hay sesión con Facebook",
            Toast.LENGTH_LONG).show();
        return;
    }

    //
    // monto la petición: /me
    //
    GraphRequest request = GraphRequest.newMeRequest(accessToken, ca-
llback);
    Bundle params = new Bundle ();
    params.putString("fields", "id, name");
    request.setParameters(params);

    //
    // la ejecuto (asíncronamente)
    //
    request.executeAsync();
}

// -----
// -----
```

```

public void boton_enviarTextoAFB_pulsado(View quien) {
    //
    // cojo el mensaje que ha escrito el usuario
    //

    String mensaje = "msg:" + this.textoConElMensaje.getText() + " :";
    + System.currentTimeMillis();

    //
    // borro lo escrito
    //
    this.textoConElMensaje.setText("");

    //
    // cierro el soft-teclado
    //
    InputMethodManager imm = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
    imm.hideSoftInputFromWindow(this.textoConElMensaje.getWindowToken(), 0);

    //
    // llamo al método que publica
    //
    enviarTextoAFacebook_async(mensaje);
}

// -----
// -----
public void boton_EnviarFoto_pulsado(View quien) {

    //
    // cojo una imagen directamente de los recursos
    // para publicarla
    //
    Bitmap image = BitmapFactory.decodeResource(
        getResources(), R.drawable.com_facebook_logo);

    //
    // llamo al método que envía la imagen
    //
    enviarFotoAFacebook_async(image, "comentario de la foto");
} // ()
} // class

```

#### 8.1.4. Aplicación de ejemplo (Share Dialog)

```
<string name="facebook_app_id">874076352708325</string>
```

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.cuandrav.pruebafacebooksdk2">

    <!-- 1. añadir estos dos permisos -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission an-
droid:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <!-- 2. añadir por Facebook -->
        <meta-data android:name="com.facebook.sdk.ApplicationId" an-
droid:value="@string/facebook_app_id" />

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        <!-- 3. añadir por Facebook -->
        <activity android:name="com.facebook.FacebookActivity" />

        <!-- 4. añadir por Facebook -->
        <!-- para poder enviar fotos con Share Dialog. Fijaos dónde hay
que poner el Facebook App ID -->
        <!-- ver: https://developers.facebook.com/docs/android/getting-
started, sección
        Sending Images or Videos -->
        <provider an-
droid:authorities="com.facebook.app.FacebookContentProvider874076352708325
"
            android:name="com.facebook.FacebookContentProvider"
            android:exported="true" />
        </application>
</manifest>

```

```
this.elShareDialog = new ShareDialog(this);
```

```

boolean respuesta = ShareDialog.canShow(ShareLinkContent.class);

boolean respuesta = ShareDialog.canShow(SharePhotoContent.class);

this.elShareDialog.show(content);

ShareLinkContent content = new ShareLinkContent.Builder()
    .setContentType(Uri.parse("https://developers.facebook.com"))
    .setContentDescription("me ha gustado esta página")
    .build();

ShareLinkContent content = new ShareLinkContent.Builder().build();

Bitmap Image = ...
SharePhoto photo = new SharePhoto.Builder()
    .setBitmap(image)
    .build();
SharePhotoContent content = new SharePhotoContent.Builder()
    .addPhoto(photo)
    .build();

```

```

package org.cuandrav.pruebafacebooksdk2;

import ...

//-----
// class MainActivity
//-----
public class MainActivity extends AppCompatActivity {

    //
    // elementos graficos
    //

    private Button boton2;
    private Button boton3;

    private Button boton1;
    private TextView textoEntrada1;
    private TextView textoSalida1;

    //
    // gestiona los callbacks al FacebookSdk desde el método onActivityResult() de una actividad

```

```

//
private CallbackManager elCallbackManagerDeFacebook;

//
// shareDialog
//
private ShareDialog elShareDialog;

//
// puntero a this para los callback
//
private final Activity THIS = this;

// -----
// -----
private void conseguirReferenciasAElementosGraficos () {
    boton1 = (Button) findViewById(R.id.button1);
    boton2 = (Button) findViewById(R.id.button2);
    boton3 = (Button) findViewById(R.id.button3);

    textoEntrada1 = (TextView) findViewById(R.id.textoEntrada1);
    textoSalida1 = (TextView) findViewById(R.id.textoSalida1);
} // ()

// -----
// -----
@Override
protected void onCreate(Bundle savedInstanceState) {
    //
    // inicializar super
    //
    super.onCreate(savedInstanceState);

    //
    Log.d("cuandrav.onCreate()", " .onCreate() llamado");

    //
    //
    // inicializar FacebookSDK
    //
    FacebookSdk.sdkInitialize(this.getApplicationContext());

    //
    // pongo el contenido visual de la actividad (hacer antes que
findViewById ()
    // y después de inicializar FacebookSDK)
    //
    this.setContentView(R.layout.activity_main);

    conseguirReferenciasAElementosGraficos();

```



```

        //
        // crear callback manager de Facebook
        //
        this.elCallbackManagerDeFacebook = CallbackManager.Factory.create();

        //
        // crear objeto share dialog
        //
        this.elShareDialog = new ShareDialog(this);

        this.elShareDialog.registerCallback(this.elCallbackManagerDeFacebook, new
        FacebookCallback<Sharer.Result>() {

            @Override
            public void onSuccess(Sharer.Result result) {
                Toast.makeText(THIS, "Sharer onSuccess()",
                Toast.LENGTH_LONG).show();
            }

            @Override
            public void onCancel() {}

            @Override
            public void onError(FacebookException error) {
                Toast.makeText(THIS, "Sharer onError(): " +
                error.toString(), Toast.LENGTH_LONG).show();
            }

        });

    }

    // -----
    // -----
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        this.elCallbackManagerDeFacebook.onActivityResult(requestCode,
        resultCode, data);
    }

    // -----
    // -----
    private void publicarMensajeConIntent () {
        Intent shareIntent = new Intent();
        shareIntent.setAction(Intent.ACTION_SEND);

        startActivityForResult(Intent.createChooser(shareIntent, "Share"),
        1234); //requestId;
    } //

```

```

// -----
// -----
public void boton1_pulsado(View quien) {
    Log.d("cuandrav.boton1_pulsado", " llamado ");
    textoSalida1.setText("boton1_pulsado");
    this.publicarMensajeConIntent();
} // ()

// -----
// -----
private boolean puedoUtilizarShareDialogParaPublicarMensaje () {
    return puedoUtilizarShareDialogParaPublicarLink();
}

// -----
private boolean puedoUtilizarShareDialogParaPublicarLink () {
    return ShareDialog.canShow(ShareLinkContent.class);
} // ()

// -----
// -----
private boolean puedoUtilizarShareDialogParaPublicarFoto () {
    return ShareDialog.canShow(SharePhotoContent.class);
} // ()

// -----
// -----
public void boton2_pulsado(View quien) {
    Log.d("cuandrav.boton2_pulsado", " llamado ");

    textoSalida1.setText("boton2_pulsado");

    //
    // llamar al metodo para publicar
    //
    this.publicarMensajeConShareDialog();
} // ()

// -----
// -----
private void publicarMensajeConShareDialog () {

    // https://developers.facebook.com/docs/android/share -> Using the
    // Share Dialog

    if ( ! puedoUtilizarShareDialogParaPublicarMensaje() ) {
        Log.d("cuandrav.boton2_pul()", " ¡¡¡ No puedo utilizar share
        dialog !!!");

        return;
    }

```

```

    }

    //
    // llamar a share dialog
    // aunque utilizamos ShareLinkContent, al no poner link
    // publica un mensaje
    //

    ShareLinkContent content = new ShareLinkContent.Builder().build();

    this.elShareDialog.show(content);
} // ()

// -----
public void boton3_pulsado(View quien) {
    Log.d("cuandrav.boton3_pulsado", " llamado ");

    textoSalida1.setText("boton3_pulsado");

    //
    // llamar al metodo para publicar foto
    //
    this.publicarFotoConShareDialog();
} // ()

// -----
// -----
private void publicarFotoConShareDialog () {

    // https://developers.facebook.com/docs/android/share -> Using the
    // Share Dialog

    if ( ! puedoUtilizarShareDialogParaPublicarFoto() ) {
        return;
    }

    //
    // cojo una imagen directamente de los recursos
    // para publicarla
    //
    Bitmap image = BitmapFactory.decodeResource(
        getResources(), R.drawable.com_facebook_logo);

    //
    // monto la petición
    //
    SharePhoto photo = new SharePhoto.Builder()
        .setBitmap(image)
        .build();
    SharePhotoContent content = new SharePhotoContent.Builder()
        .addPhoto(photo)

```

```
        .build();

        this.elShareDialog.show(content);

    } // ()

} // class
```

## 8.2. Android y Twitter

### 8.2.1. Configurando Fabric en nuestra aplicación de Android Studio

```
private static final String TWITTER_KEY = "WOHF727DwGJBxy8uPtJxA";
private static final String TWITTER_SECRET = "kc40JcVSuxZeWRgq";
```

### 8.2.2. Aplicación de Ejemplo

```
<meta-data
    android:name="io.fabric.ApiKey"
    android:value="93e32eale8b8d08c046cbd05a67472cf054efc70" />
</application>

<uses-permission android:name="android.permission.INTERNET" />

<com.twitter.sdk.android.core.identity.TwitterLoginButton
    android:id="@+id/twitter_login_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

private static final String TWITTER_KEY = "WOHF727DwGJBxy8uPtJxA";
private static final String TWITTER_SECRET = "kc40JcVSux";

TwitterAuthConfig authConfig = new TwitterAuthConfig(TWITTER_KEY,
TWITTER_SECRET);
Fabric.with(this, new Twitter(authConfig);

botonEnviarATwitter = (Button) findViewById(
    R.id.boton_EnviarATwitter);

botonLoginTwitter.setCallback(new Callback<TwitterSession>() {
```

```

        @Override
        public void success(Result<TwitterSession> result) {
            Toast.makeText(THIS, "Autenticado en twitter: " + re-
sult.data.getUserName(), Toast.LENGTH_LONG).show();
        }

        @Override
        public void failure(TwitterException e) {
            Toast.makeText(THIS, "Fallo en autenticación: " +
e.getMessage(), Toast.LENGTH_LONG).show();
        }
    });

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        botonLoginTwitter.onActivityResult(requestCode, resultCode, da-
ta);
    }

    TwitterSession session = Twit-
ter.getSessionManager().getActiveSession();
    StatusesService statusesService = Twit-
ter.getApiClient(sesion).getStatusesService();

    statusesService.update(textoQueEnviar, null, null, null, null, null,
null, null, null, new Callback<Tweet>() {
        @Override
        public void success(Result<Tweet> result) {
            Toast.makeText(THIS, "Tweet publicado: "+ re-
sult.response.getReason().toString(), Toast.LENGTH_LONG).show();
        }

        @Override
        public void failure(TwitterException e) {
            Toast.makeText(THIS, "No se pudo publicar el tweet: "+
e.getMessage(), Toast.LENGTH_LONG).show();
        }
    });

    TweetComposer.Builder builder = new TweetComposer.Builder(this)
        .text("just setting up my Fabric.");
    //image(myImageUri);
    builder.show();

    private void enviarImagen_async () {
        File photo = null;
        try {
            // 1. Abrimos el fichero con la imagen

```

```
// que queremos enviar. Como necesitamos un path (para Typed-
File)
// debe estar fuera de /res o /assets porque estos
// estan dentro del .apk y NO tiene path
photo = new File ("/storage/sdcard0/DCIM/100ANDRO/ok.png");

Log.d ("pruebaImagen", " can read: " + photo.canRead());
Log.d("pruebaImagen", " absolute path: " + photo.getAbsolutePath());
Log.d("pruebaImagen", " length: " + photo.length());
} catch (Exception e) {
    Log.d("pruebaImagen", " excepcion: " + e.getMessage());
    return;
}
// 2. ponemos el fichero en un TypedFile
TypedFile typedFile = new TypedFile("image/png", photo);
Log.d("pruebaImagen", " cargando: " + typedFile.toString()+ " - "
+ typedFile.fileName());
// 3. obtenemos referencia al media service
MediaService ms = Twitter.getApiClient( obtenerSesionDeTwitter()
).getMediaService();
// 4. con el media service: enviamos la foto a Twitter
ms.upload(
    typedFile, // foto que enviamos
    null, null,
    new Callback<Media>() {
        @Override
        public void success(Result<Media> mediaResult) {
            // he tenido éxito:
            Log.d("cuandrav.pruebaImagen", " ms.upload -> success: " +
mediaResult.response.getReason() +
" " + mediaResult.data.mediaId);
            Toast.makeText(THIS, "imagen publicada: " +
mediaResult.response.toString(), Toast.LENGTH_LONG);

            // 5. como he tenido éxito, la foto está en twitter, pero no en el
            timeline
            // he de escribir un tweet referenciando la foto

            // 6. obtengo referencia al status service
            StatusesService statusesService =
                TwitterCore.getInstance().getApiClient(obtenerSesionDeTwitter())
                    .getStatusesService();

            // 7. publico un tweet
            statusesService.update("prueba de enviar imagen
"+System.currentTimeMillis() , // mensaje del tweet
```

```

    null, false, null, null, null, true, false,
    "+" + mediaResult.data.mediaId, // string con los identificadores (hasta
    4, separado por coma) de las imágenes
    // que quiero que aparezcan en este tweet. El mediaId referencia
    a la foto que acabo de subir previamente

    new Callback<Tweet>() {
        @Override
        public void success(Result<Tweet> result) {
            Toast.makeText(THIS, "Tweet publicado: " +
            result.response.getReason().toString(), Toast.LENGTH_LONG).show();
        }

        @Override
        public void failure(TwitterException e) {
            Toast.makeText(THIS, "No se pudo publicar el tweet: "
            "+" e.getMessage(), Toast.LENGTH_LONG).show();
        }
    });
}

@Override
public void failure(TwitterException e) {
    Toast.makeText(THIS, "No se pudo publicar el tweet: " +
    e.getMessage(), Toast.LENGTH_LONG).show();
    Log.d("pruebaImagen", " failure: del upload: " +
    e.getMessage());
}
});
} // ()

```

# CAPÍTULO 9.

## Ingeniería inversa en Android

Por JESÚS TOMÁS

### 9.1. El formato APK

```
Manifest-Version: 1.0
Created-By: 1.6.0_29 (Sun Microsystems Inc.)

Name: res/drawable/facebook_icon.png
SHA1-Digest: 9jTbeG4b8nJZlZss16UPfagqlUs=

Name: res/drawable/com_facebook_picker_list_longpressed.9.png
SHA1-Digest: /uU9+qqC9MjC0cwR3L+DskCy9xM=

Name: res/drawable/square_d.xml
SHA1-Digest: kl+o4e30jU9gil/l+A/AUGBcI8U=

Name: res/drawable-ldpi/warply_notifications.png
SHA1-Digest: qaM+ngxBdXq0EfBbTdLffAJ0D70=

...
```

```
Signature-Version: 1.0
SHA1-Digest-Manifest-Main-Attributes: ngANqza5/whaYZFfNQhzhT/iRBA=
Created-By: 1.6.0_29 (Sun Microsystems Inc.)
SHA1-Digest-Manifest: cbc183Z5qhSYsrmSxoegFDp5I3c=

Name: res/drawable/facebook_icon.png
```



```
SHA1-Digest: PDgDHoip/6HBjLGwRLz06l35sy4=
Name: res/drawable/com_facebook_picker_list_longpressed.9.png
SHA1-Digest: 2CGPL/jp5oPTQ2ltrIIWxbJ07UM=
Name: res/drawable/square_d.xml
SHA1-Digest: 5UTm9fULZG3sQPFpSsTpofL7Yro=
Name: res/drawable-ldpi/warply_notifications.png
SHA1-Digest: 4KFSugOGx0l271gbgNaqEUvi8Fk=
...
```

```
C:\>"C:\Program Files (x86)\Java\jre7\bin\keytool" -printcert -file
APAL.RSA
Propietario: O=Etermax
Emisor: O=Etermax
Número de serie: 4e861907
Válido desde: Fri Sep 30 21:31:19 CEST 2011 hasta: Sun Sep 06 21:31:19
CEST 2111
Huellas digitales del Certificado:
    MD5: 05:0A:16:21:5C:61:4C:76:9D:3E:7C:38:65:5F:61:ED
    SHA1: 13:A3:A4:55:F0:C1:26:26:42:2C:52:C4:54:4C:DC:B5:A2:27:52:25
    SHA256: 6D:56:0E:33:C6:C6:9C:A4:6F:37:98:8B:DF:42:CC:8D:E2:BC:BB:E8:
        1D:87:36:D6:42:44:18:02:67:D1:F3:EE
Nombre del Algoritmo de Firma: SHA1withRSA
Versión: 3
```

## 9.2. Decompilando aplicaciones Android

### 9.2.1. La máquina virtual Dalvik/ART

```
public static long sumArray(int[] arr) {
    long sum = 0;
    for (int i : arr) {
        sum += i;
    }
    return sum;
}
```

### 9.2.2. Decompilando aplicaciones Android

```
C:\C9>dex2jar\d2j-dex2jar classes.dex
dex2jar classes.dex -> classes-dex2jar.jar
```

## 9.3. Modificando aplicaciones Android

### 9.3.1. Modificando recursos XML de una aplicación

```
...
<activity android:theme="@*android:style/Theme.NoTitleBar.Fullscreen"
    android:label="@string/app_name"
    android:name="com.etermax.apalabrados.ui.SplashActivity_"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
...
```

```
...
<string name="new_game">Nueva partida</string>
...
```

```
Please make your decision:12
Building Apk
"An Error Occured, Please Check The Log (option 21)"
Presione una tecla para continuar . . .
```

```
I: Building resources...
C:\C9\APK-Multi-Tool\other\..\projects\com.etermax.apalabrados.lite.apk
\res\values\styles.xml:166: error: Error retrieving parent for item: No
resource found that matches the given name 'Widget.AppCompat.Base'
C:\C9\APK-Multi-Tool\other\..\projects\com.etermax.apalabrados.lite.apk
\res\values\styles.xml:206: error: Error retrieving parent for item: No
resource found that matches the given name 'Widget.AppCompat.Base'
...
5 errores más que hacen referencia a 'Widget.AppCompat.Base'
...
C:\C9\APK-Multi-Tool\other\..\projects\com.etermax.apalabrados.lite.apk
\res\values\styles.xml:313: error: Error retrieving parent for item: No
resource found that matches the given name Widget.AppCompat.Base.DropDownItem
C:\C9\APK-Multi-Tool\other\..\projects\com.etermax.apalabrados.lite.apk\
res\values\styles.xml:346: error: Error retrieving parent for item: No
resource found that matches the given name 'Widget.AppCompat.Light.Base'
C:\C9\APK-Multi-Tool\other\..\projects\com.etermax.apalabrados.lite.apk\
res\values\styles.xml:365: error: Error retrieving parent for item: No
resource found that matches the given name 'TextAppearance.AppCompat.Base'
C:\C9\APK-Multi-Tool\other\..\projects\com.etermax.apalabrados.lite.apk
\res\values\styles.xml:592: error: Error retrieving parent for item: No
resource found that matches the given name 'Theme.AppCompat.Base'.
...
```

```
<style name="Widget.AppCompat.Base.ActionBar">
...
</style>
...
```

```
<style name="Widget.AppCompat.Base"/>
<style name="Widget.AppCompat.Base.ActionBar">
...
</style>
...
```

```
<style name="Widget"/>
<style name="Widget.AppCompat"/>
<style name="Widget.AppCompat.Base"/>
<style name="Widget.AppCompat.Base.ActionBar">
    <item name="actionOverflowButtonStyle">...</item>
    <item name="actionButtonStyle">...</item>
...
</style>
...
```

### 9.3.2. Modificando el código de una aplicación

```
package com.etermax.gamescommon.login.ui;
...
public abstract class BaseSplashActivity extends FragmentActivity
{
    public static final String LAUNCH_DATA = "launch_data";
    protected static final int LOGIN_REQUEST;
    protected static int SPLASH_DURATION = 2000;
    ...
}
```

```
.class public abstract(1)
Lcom/etermax/gamescommon/login/ui/(2)BaseSplashActivity(3);
.super Landroid/support/v4/app/FragmentActivity;(4)
.source "BaseSplashActivity.java"

# static fields
.field public static final LAUNCH_DATA:Ljava/lang/String; = "launch_data"
.field protected static final LOGIN_REQUEST:I(5)
.field protected static SPLASH_DURATION:I(6)

# direct methods
.method static constructor <clinit>()V(7)
    .locals 1
```

```

.prologue
.line 35
const/16 v0, 0x7d0
sput v0, Lcom/etermax/gamescommon/login/ui/BaseSplashActivity;>
                                                    SPLASH_DURATION:I

    return-void
.end method

...

```

```

package com.etermax.gamescommon.login.ui;(2)

import android.content.Intent;

...
public abstract class(1) BaseSplashActivity(3) extends FragmentActivity(4)
{
    protected static final int LOGIN_REQUEST;(5)
    protected static int SPLASH_DURATION(6) = 2000;

    ...

```

## 9.4. Ofuscación del código

```

...
.field protected static SPLASH_DURATION:I

# direct methods
.method static constructor <clinit>()V
    .locals 1
    .prologue
    .line 35
    ...

```

```

private String cadena = "Hola Mundo";

public String bienvenida (){
    return cadena;
}

public void noUsada() {}

protected void onCreate(Bundle paramBundle) {
    ...
    Toast.makeText(this, bienvenida(),Toast.LENGTH_LONG).show();
}

```

```
C:\C9>dex2jar\d2j-dex2jar classes_no_ofusc.dex
```

```
dex2jar classes_no_ofusc.dex -> classes_no_ofusc-dex2jar.jar
```

```
-optimizationpasses 5
-dontusemixedcaseclassnames
-dontskipnonpubliclibraryclasses
-dontpreverify
-verbose
-optimizations
!code/simplification/arithmetic,!field/*,!class/merging/*

-keep public class * extends android.app.Activity
-keep public class * extends android.app.Application
-keep public class * extends android.app.Service
-keep public class * extends android.content.BroadcastReceiver
-keep public class * extends android.content.ContentProvider
-keep public class * extends android.app.backup.BackupAgentHelper
-keep public class * extends android.preference.Preference
-keep public class com.android.vending.licensing.ILicensingService

-keepclasseswithmembernames class * {
    native <methods>;
}

-keepclasseswithmembers class * {
    public <init>(android.content.Context, android.util.AttributeSet);
}

-keepclasseswithmembers class * {
    public <init>(android.content.Context, android.util.AttributeSet, int);
}

-keepclassmembers class * extends android.app.Activity {
    public void *(android.view.View);
}

-keepclassmembers enum * {
    public static **[] values();
    public static ** valueOf(java.lang.String);
}

-keep class * implements android.os.Parcelable {
    public static final android.os.Parcelable$Creator *;
}
```

```
public class MainActivity extends ActionBarActivity → e {

    private String cadena → n = "Hola Mundo";

    public String bienvenida → i(){
        return cadena → n;
    }
}
```

```

public void noUsada() {}

protected void onCreate(Bundle paramBundle) {
    super.onCreate(paramBundle);
    setContentView(2130903063);
    Toast.makeText(this, bienvenida → i (), 1).show();;
}
...

```

## 9.5. Obtención de licencias con Google Play

### 9.5.1. Como añadir una licencia a nuestra aplicación

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity" >
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="comprobarLicencia"
        android:text="Comprobar Licencia" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="entrar"
        android:text="Entrar" />
</LinearLayout>

```

```

public class MainActivity extends Activity implements LicenseCheckerCallback {
    private static final String CLAVE_PUBLICA_LICENCIA = "Tu clave pública";
    // Genera 20 bytes aleatorios, y reemplazalos por los siguientes.
    private static final byte[] SALT = new byte[] {-46, 65, 30, -128, -103,
        -57, 74, -64, 51, 88, -95, -45, 77, -117, -36, -113, -11, 32, -64, 89};
    LicenseChecker comprobarLicencia;
    boolean permitir = false;
    ProgressDialog dialogo;

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

        String idDispositivo = Secure.getString(getContentResolver(),
            Secure.ANDROID_ID);

        ServerManagedPolicy politica = new ServerManagedPolicy(this,
            new AESObfuscator(SALT, getPackageName(), idDispositivo));
        comprobarLicencia = new LicenseChecker(this, politica,

```

```

        CLAVE_PUBLICA_LICENCIA);
        dialogo = new ProgressDialog(this);
        dialogo.setTitle("comprobando licencia");
        dialogo.setIndeterminate(true);
    }

    public void comprobarLicencia(View view) {
        dialogo.show();
        comprobarLicencia.checkAccess(this);
    }

    public void entrar(View view) {
        if (permitir) {
            Toast.makeText(this, "Entrando en aplicación",
                           Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(this, "Licencia no válida",
                           Toast.LENGTH_LONG).show();
        }
    }

    @Override public void allow(int reason) {
        permitir = true;
        Toast.makeText(this, "Licencia correcta: " + reason,
                       Toast.LENGTH_LONG).show();
        dialogo.dismiss();
    }

    @Override public void dontAllow(int reason) {
        permitir = false;
        Toast.makeText(this, "Licencia no válida: " + reason,
                       Toast.LENGTH_LONG).show();
        dialogo.dismiss();
    }

    @Override public void applicationError(int errorCode) {
        Toast.makeText(this, "Error al comprobar licencia: " + errorCode,
                       Toast.LENGTH_LONG).show();
        dialogo.dismiss();
    }

    @Override protected void onDestroy() {
        super.onDestroy();
        comprobarLicencia.onDestroy();
    }
}

```

## 9.6. Cómo evitar que se elimine la verificación de licencia en nuestras aplicaciones

### 9.6.1. Ingeniería inversa en una aplicación con licencia

```
.class public Lorg/tomas/girones/jesus/obtencionlicencia/MainActivity;
.super Landroid/app/Activity;
.source "MainActivity.java"

# interfaces
.implements Lcom/google/android/vending/licensing/LicenseCheckerCallback;

# static fields
.field private static final CLAVE PUBLICA LICENCIA:Ljava/lang/String; = ...
.field private static final SALT:[B

# instance fields
.field comprobarLicencia:Lcom/google/android/vending/licensing/
                                     LicenseChecker;
.field dialogo:Landroid/app/ProgressDialog;
.field permitir:Z

# direct methods
.method static constructor <clinit>()V
    .locals 1
    .prologue
    .line 18
    const/16 v0, 0x14
    new-array v0, v0, [B
    fill-array-data v0, :array_0
    sput-object v0, Lorg/.../obtencionlicencia/MainActivity;->SALT:[B
    .line 19
    return-void
    ...
```

```
@Override public void allow(int reason) {
    permitir = true;
    ...

# virtual methods
.method public allow(I)V
    .locals 3
    .parameter "reason"
    .prologue
    const/4 v2, 0x1
    .line 54
    iput-boolean v2, p0, Lorg/tomas/girones/jesus/obtencionlicencia/
                                     MainActivity; ->permitir:Z
    ...
```



```
Toast.makeText(this, "Licencia correcta: "+reason, Toast.LENGTH_LONG)
    .show();
```

```
...
.line 55
new-instance v0, Ljava/lang/StringBuilder;
const-string v1, "Licencia correcta: "
invoke-direct {v0, v1}, Ljava/lang/StringBuilder;.<init>
    (Ljava/lang/String;)V
invoke-virtual {v0, p1}, Ljava/lang/StringBuilder;.>
    append(I)Ljava/lang/StringBuilder;
move-result-object v0
invoke-virtual {v0}, Ljava/lang/StringBuilder;.>toString()
    Ljava/lang/String;
move-result-object v0
invoke-static {p0, v0, v2}, Landroid/widget/Toast;.>
    makeText(Landroid/content/Context;Ljava/lang/CharSequence;I)
    Landroid/widget/Toast;
move-result-object v0
.line 56
invoke-virtual {v0}, Landroid/widget/Toast;.>show()V
...
```

```
...
    dialogo.dismiss();
}
```

```
...
.line 57
iget-object v0, p0, Lorg/tomas/girones/jesus/obtencionlicencia/
    MainActivity;.> dialogo:Landroid/app/ProgressDialog;
invoke-virtual {v0}, Landroid/app/ProgressDialog;.>dismiss()V
.line 58
return-void
.end method
```

2.

```
@Override public void dontAllow(int reason) {
    permitir = false;
    ...
}
```

```
# virtual methods
.method public dontAllow(I)V
    .locals 2
    .parameter "reason"
    .prologue
    const/4 v0, 0x0
    .line 62
```

```
input-boolean v0, p0, Lorg/tomas/girones/jesus/obtencionlicencia/
    MainActivity; ->permitir:Z
...
```

### 9.6.2. Primera contramedida: ofuscar el código

```
.class public Lorg/tomas/girones/jesus/obtencionlicencia/MainActivity;
.super Landroid/app/Activity;

# interfaces
.implements La/a/a/a/a/m;

# static fields
.field private static final d:[B

# instance fields
.field a:La/a/a/a/a/i;
.field b:Z
.field c:Landroid/app/ProgressDialog;

# direct methods
.method static constructor <clinit>()V
    .locals 1
    const/16 v0, 0x14
    new-array v0, v0, [B
    fill-array-data v0, :array_0
    sput-object v0, Lorg/.../obtencionlicencia/MainActivity;->d:[B
    return-void
...
```

```
.class public interface abstract La/a/a/a/a/m;
.super Ljava/lang/Object;

# virtual methods
.method public abstract a(I)V
.end method

.method public abstract b(I)V
.end method

.method public abstract c(I)V
.end method
```

### 9.6.3. Segunda contramedida: no usar la librería LVL estándar

```
public void dontAllow(int reason);
```

```
public String dontAllow(String s, short reason);
```

```
public String noHaceNada(String s, short reason);
```

```
public static final int ERROR_INVALID_PACKAGE_NAME = 1;
public static final int ERROR_NON_MATCHING_UID = 2;
public static final int ERROR_NOT_MARKET_MANAGED = 3;
public static final int ERROR_CHECK_IN_PROGRESS = 4;
public static final int ERROR_INVALID_PUBLIC_KEY = 5;
public static final int ERROR_MISSING_PERMISSION = 6;
```

```
validator.getCallback().dontAllow(Policy.RETRY);
```

```
validator.getCallback().dontAllow("", (short) Policy.RETRY);
```

```
@Override public void dontAllow(int reason) {
```

```
@Override public String dontAllow(String s, short reason) {
```

```
return s+reason;
```

```
.class public interface abstract La/a/a/a/a/m;
.super Ljava/lang/Object;

# virtual methods
.method public abstract a(Ljava/lang/String;)Ljava/lang/String;
.end method

.method public abstract a(Ljava/lang/String;S)Ljava/lang/String;
.end method

.method public abstract a(I)V
.end method

.method public abstract b(I)V
.end method
```

### 9.6.4. Tercera contramedida: verificar que no ha modificado nuestra APK

```
public boolean verificarFirma() {  
    try {  
        Signature[] firmas = getPackageManager().getPackageInfo(  
            getPackageName(), PackageManager.GET_SIGNATURES).signatures;  
        int i = firmas[0].hashCode();  
        Toast.makeText(this, i+", "+firmas[0].toCharsString(),  
            Toast.LENGTH_LONG).show();  
        return i == -633674321; //Reemplaza este valor  
    } catch (NameNotFoundException e) {  
        return false;  
    }  
}
```

```
public void entrar(View view) {  
    if (permitir) {  
        if (verificarFirma()) {  
            Toast.makeText(this, "Entrando en aplicación",  
                Toast.LENGTH_LONG).show();  
        } else {  
            Toast.makeText(this, "Firma cambiada", Toast.LENGTH_LONG).show();  
        }  
    } else {  
        Toast.makeText(this, "Licencia no válida", Toast.LENGTH_LONG).show();  
    }  
}
```