

El gran libro de Android avanzado

Jesús Tomás
Vicente Carbonell
Carsten Vogt
Miguel García Pineda
Jordi Bataller Mascarell
Daniel Ferri



El gran libro de Android avanzado

Primera edición, 2013

© 2013 Jesús Tomás Gironés, Vicente Carbonell, Carsten Vogt, Miguel García Pineda,
Jordi Bataller Mascarell y Daniel Ferri

© MARCOMBO, S.A. 2013

Gran Via de les Corts Catalanes, 594

08007 Barcelona

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) si necesita fotocopiar o escanear algún fragmento de esta obra».

ISBN: 978-84-267-2078-8

DL:

Printed in Spain

Índice general

CAPÍTULO 1. Diseño avanzado de interfaces de usuario.....	9
1.1. <i>GridView</i>	9
1.2. <i>Fragments</i>	12
1.2.1. Insertar <i>fragments</i> desde XML	12
1.2.2. Insertar <i>fragments</i> desde código.....	15
1.2.3. Comunicación e intercambio de <i>fragments</i>	16
1.3. <i>La barra de acciones (ActionBar)</i>	19
1.3.1. Añadiendo preferencias de usuario mediante PreferenceFragment.....	21
1.4. <i>Servicios de búsquedas</i>	22
1.5. <i>Animaciones</i>	23
1.5.1. Animaciones de vistas: transiciones entre actividades	23
1.5.1.1. Aplicando animaciones de vistas en Audiolibros	24
1.5.2. Animaciones de propiedades	25
1.5.2.1. El motor básico de animación: <i>ValueAnimator</i>	25
1.5.2.2. Automatizando las animaciones: <i>ObjectAnimator</i>	26
1.5.2.3. Combinando animaciones: <i>AnimatorSet</i>	26
1.5.2.4. Definiendo animaciones en XML	26
1.5.2.5. Nuevas propiedades de la clase <i>View</i>	27
1.5.2.6. Aplicando animaciones de propiedades en Audiolibros	28
CAPÍTULO 2. Diseño personalizado de vistas	29
2.1. <i>Algunos conceptos básicos</i>	29
2.2. <i>Una vista como la composición de varias vistas</i>	29
2.2.1. Creación de escuchadores de eventos	31
2.3. <i>Modificación de vistas existentes</i>	33
2.3.1. Algo más de información sobre <i>TextView</i>	34
2.4. <i>Creación de nuevos atributos XML</i>	35
2.5. <i>Una vista creada desde cero</i>	37
2.5.1. Diseño y dibujo de la vista	37

2.5.2. Gestión de eventos.....	40
2.5.3. Cómo Android dibuja las vistas y obtiene sus tamaños.....	42
2.5.4. Interactuando con otros objetos.....	42
2.6. <i>Creación de widgets de escritorio</i>	43
2.6.1. Pasos a seguir para crear un <i>widget</i>	43
2.6.1.1. Definir las características del <i>widget</i>	43
2.6.1.2. Diseñar el <i>layout</i> del <i>widget</i>	43
2.6.1.3. Crear una clase descendiente de <i>AppWidgetProvider</i>	43
2.6.1.4. Declarar el <i>widget</i> en <i>AndroidManifest</i>	43
2.6.1.5. Crear una actividad para configurarlo.....	44
2.6.2. Creación de un <i>widget</i> de escritorio sencillo.....	44
2.6.3. Actualizando el <i>widget</i> de escritorio.....	45
2.6.4. Actuando ante el evento <i>onClick</i>	46
2.6.5. Añadiendo una actividad de configuración.....	47
CAPÍTULO 3. Hilos de ejecución en la interfaz del usuario.....	49
3.1. <i>Programación basada en eventos y el hilo de ejecución de usuario</i>	49
3.1.1. Cola de eventos y bucle de eventos.....	49
3.1.2. El hilo de la interfaz de usuario de una aplicación Android.....	50
3.2. <i>Concurrencia en programación orientada a eventos</i>	51
3.2.1. Hilos para el manejo de eventos.....	51
3.2.2. El problema de los hilos en segundo plano: no tienen acceso a la interfaz gráfica de usuario.....	52
3.3. <i>La clase AsyncTask</i>	52
3.3.1. Extendiendo <i>AsyncTask</i>	52
3.3.2. Secuencia de operaciones.....	53
3.4. <i>Animaciones con SurfaceView</i>	57
3.4.1. Programación con <i>SurfaceViews</i>	57
CAPÍTULO 4. Comunicación con Bluetooth.....	61
4.1. <i>Diferencias entre Bluetooth e Internet</i>	61
4.2. <i>Los pasos en la comunicación Bluetooth</i>	61
4.2.1. Pasos de programación.....	61
4.3. <i>Algunas clases de utilidad autodefinidas</i>	63

4.3.1. Clase de utilidad para un servidor	63
4.3.2. Clase utilidad para un cliente	64
4.3.3. Hilos de fondo	66
4.3.3.1. Hilo de fondo del servidor	66
4.3.3.2. Hilo de fondo del cliente	70
4.4. Comunicación entre dispositivos Android	73
4.5. La comunicación con los programas en Java SE	75
CAPÍTULO 5. Servicios en la nube	81
5.1. Introducción a los servicios en la nube	81
5.2. Notificaciones push	81
5.2.1. Servicio Google Cloud Messaging	81
5.2.2. Activar Google Cloud Messaging en Google API Console	81
5.2.3. Aplicación cliente Google Cloud Messaging	81
5.2.4. Aplicación servidor Google Cloud Messaging	87
5.3. Almacenamiento en la nube	89
5.3.1. Almacenamiento en Google Drive	89
5.3.2. Google Drive API v2	89
5.3.2.1. Extracción de la huella digital (SHA1)	89
5.3.2.2. Habilitar el servicio Google Drive API	90
5.3.2.3. Autorizar el acceso a Google Drive	90
5.3.2.4. Subir ficheros a Google Drive	92
5.4. Servicio de Backup de Google	95
5.4.1. Fundamentos	95
5.4.2. Declaración del agente de copia de seguridad en Manifest	95
5.4.3. Registro del servicio Android Backup	95
5.4.4. BackupAgent	95
5.4.5. BackupAgentHelper	95
5.4.5.1. Copia de seguridad de SharedPreferences	95
5.4.5.2. Copia de seguridad de archivos de almacenamiento interno	96
5.4.6. Comprobación de la versión al restaurar los datos	96
5.4.7. Solicitud de copia de seguridad y restauración	96
5.4.8. Un ejemplo paso a paso	96

CAPÍTULO 6. Aplicaciones web en Android.....	98
6.1. Introducción a la tecnología web.....	98
6.1.1. Una aplicación web de ejemplo: 3 en Raya	98
6.1.2. Aplicación web <i>online</i> y <i>offline</i>	104
6.2. Uso de WebView.....	104
6.2.1. Mostrar contenido web usando una intención.....	104
6.2.2. mostrar contenido web	105
6.2.3. Aspectos básicos de un WebView	106
6.2.3.1. Evitar el reinicio de la actividad.....	106
6.2.3.2. Abrir los enlaces en el WebView.....	106
6.2.3.3. Opciones de inicio.....	106
6.2.3.4. Barra de progreso	106
6.2.3.5. Navegación	108
6.2.3.6. Controlar el botón «Volver»	109
6.2.3.7. Habilitar alertas JavaScript	110
6.2.3.8. Gestión de errores.....	110
6.2.3.9. Descargas	110
6.2.3.10. Conectividad	112
6.3. Diseño web en Android	113
6.3.1. Área de visualización y escalado	113
6.3.2. Escalado	113
6.3.3. Densidad de pantalla del dispositivo	113
6.4. Aplicaciones híbridas	114
6.5. Alternativas en la programación independiente de la plataforma para móviles	117
6.5.1. Phonegap	117
6.5.2. JQuery Mobile	119
6.5.2.1. Crear una página básica	119
6.5.2.2. Añadir contenido	120
6.5.2.3. Crear una lista	120
6.5.2.4. Añadir un deslizador.....	120
6.5.2.5. Crear un botón	121
6.5.2.6. Temas	121

CAPÍTULO 7. Programación en código nativo	125
7.1. <i>Android NDK</i>	125
7.2. <i>Instalación de Android NDK</i>	125
7.2.1. Instalación Android NDK en Windows	125
7.2.2. Instalación Android NDK en Linux	125
7.3. <i>Funcionamiento y estructura de Android NDK</i>	126
7.3.1. Desarrollo práctico de Android NDK	126
7.3.2. Situación del código fuente nativo	126
7.3.2.1. Fichero <i>Android.mk</i>	126
7.3.2.2. Fichero <i>Application.mk</i> (opcional)	127
7.3.2.3. La herramienta <i>ndk-build</i>	127
7.4. <i>Interfaz entre JAVA y C/C++ (JNI)</i>	128
7.4.1. Librerías de enlace estático y dinámico	128
7.4.2. Tipos fundamentales, referencias y <i>arrays</i>	128
7.4.3. Desarrollo paso a paso de un programa mediante JNI (I)	129
7.4.3.1. Declaración del método nativo y creación del archivo <i>Android.mk</i>	129
7.4.3.2. Creación del fichero de cabecera nativo	130
7.4.3.3. Implementación del método nativo	130
7.4.4. Acceso a métodos Java desde código nativo (<i>JNI callback</i>)	130
7.4.4.1. Métodos de instancia	130
7.4.4.2. Métodos de clase	131
7.4.4.3. Invocar constructores	131
7.5. <i>Rendimiento de aplicaciones con código nativo</i>	134
7.6. <i>Procesado de imagen con código nativo</i>	138
CAPÍTULO 8. Redes sociales: Facebook y Twitter	143
8.1. <i>Android y Facebook</i>	143
8.1.1. Preliminares	143
8.1.1.1. Darse de alta en Facebook como desarrollador	143
8.1.1.2. SDK de Facebook para Android	143
8.1.1.3. Configurando nuestra aplicación	143
8.1.2. Nuestro proyecto Android	143

8.1.3. Aplicación de ejemplo.....	144
8.2. <i>Android y Twitter</i>	150
8.2.1. Preliminares.....	150
8.2.2. Configurando nuestra aplicación.....	150
8.2.3. Aplicación de ejemplo.....	151
CAPÍTULO 9. Ingeniería inversa en Android.....	159
9.1. <i>El formato APK</i>	159
9.2. <i>Decompilando aplicaciones Android</i>	161
9.2.1. La máquina virtual Dalvic	161
9.2.2. Decompilando aplicaciones Android	161
9.3. <i>Modificando aplicaciones Android</i>	161
9.3.1. Modificando recursos binarios de una aplicación.....	161
9.3.2. Modificando recursos XML de una aplicación	162
9.3.3. Modificando el código de una aplicación.....	163
9.4. <i>Ofuscación del código</i>	164
9.5. <i>Obtención de licencias con Google Play</i>	166
9.5.1. Cómo funciona el servicio de licencias.....	166
9.5.2. Como añadir una licencia a nuestra aplicación.....	166
9.6. <i>Cómo evitar que se elimine la verificación de licencia en nuestras aplicaciones</i>	168
9.6.1. Ingeniería inversa en una aplicación con licencia	168
9.6.2. Primera contramedida: ofuscar el código	170
9.6.3. Segunda contramedida: no usar la librería LVL estándar.....	171
9.6.4. Tercera contramedida: verificar que no ha modificado nuestra APK.....	172

CAPÍTULO 1.

Diseño avanzado de interfaces de usuario

Por DANIEL FERRI y JESÚS TOMÁS

1.1. GridView



Ejercicio paso a paso: *Primera versión de Audiolibros con un GridView.*

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:columnWidth="120dp"
    android:numColumns="auto_fit"
    android:verticalSpacing="10dp"
    android:horizontalSpacing="10dp"
    android:gravity="top"
/>
```

1.

```
public class MainActivity extends Activity {
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    GridView gridView = (GridView) findViewById(R.id.gridview);
    gridView.setAdapter(new SelectorAdapter(this));
    gridView.setOnItemClickListener(new OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View v,
                                int position, long id) {
            Toast.makeText(MainActivity.this, "Seleccionado el elemento: "
                                + position, Toast.LENGTH_SHORT).show();
        }
    });
}
}

```

```

public class SelectorAdapter extends BaseAdapter {
    LayoutInflater inflater;
    public static Vector<BookInfo> bookVector;

    public SelectorAdapter(Activity a) {
        inflater = (LayoutInflater) a
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        inicializarVector();
    }

    public int getCount() {
        return bookVector.size();
    }

    public Object getItem(int position) {
        return null;
    }

    public long getItemId(int position) {
        return 0;
    }

    public View getView(int position, View convertView, ViewGroup parent){
        ImageView imageView;
        TextView audiolibroTextView;
        BookInfo bookInfo = bookVector.elementAt(position);
        View view = convertView;
        if (convertView == null) {
            view = inflater.inflate(R.layout.elemento_selector,null);
        }
        audiolibroTextView = (TextView) view.findViewById(R.id.titulo);
        imageView = (ImageView) view.findViewById(R.id.imageView1);
        imageView.setImageResource(bookInfo.resourceImage);
        imageView.setScaleType(ImageView.ScaleType.CENTER_INSIDE);
        audiolibroTextView.setText(bookInfo.name);
        return view;
    }
}

```

```
public static void inicializarVector() {
    bookVector = new Vector<BookInfo>();
    bookVector.add(new BookInfo("Kappa", "Akutagawa", R.drawable.kappa,
        "http://www.leemp3.com/leemp3/1/Kappa_akutagawa.mp3"));
    bookVector.add(new BookInfo("Avecilla", "Alas Clarín, Leopoldo",
        R.drawable.avecilla,
        "http://www.leemp3.com/leemp3/Avecilla_alas.mp3"));
    bookVector.add(new BookInfo("Divina Comedia", "Dante",
        R.drawable.divinacomedia,
        "http://www.leemp3.com/leemp3/8/Divina%20Comedia_alighier.mp3"));
    bookVector.add(new BookInfo("Viejo Pancho, El",
        "Alonso y Trelles, José", R.drawable.viejo_pancho,
        "http://www.leemp3.com/leemp3/1/viejo_pancho_trelles.mp3"));
    bookVector.add(new BookInfo("Canción de Rolando", "Anónimo",
        R.drawable.cancion_rolando,
        "http://www.leemp3.com/leemp3/1/Cancion%20de%20Rolando_
            anonimo.mp3"));
    bookVector.add(new BookInfo("Matrimonio de sabuesos", "Agata Christie",
        R.drawable.matrimonio_sabuesos,
        "http://www.dcomg.upv.es/~jtomas/android/audiolibros/
            01.%20Matrimonio%20De%20Sabuesos.mp3"));
    bookVector.add(new BookInfo("La iliada", "Homero",
        R.drawable.iliada,
        "http://www.dcomg.upv.es/~jtomas/android/audiolibros/la-iliada-
            homero184950.mp3"));
}
```

```
public class BookInfo {
    public String name;
    public String autor;
    public int resourceImage;
    public String url;

    public BookInfo(String name, String autor, int resourceImage,
        String url) {
        this.name = name;
        this.autor = autor;
        this.resourceImage = resourceImage;
        this.url = url;
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
```

```

        android:layout_alignBottom="@+id/imageView1"
        android:layout_alignParentLeft="true"
        android:layout_marginBottom="-21dp"
        android:src="@drawable/sombra0" />
<ImageView
    android:id="@+id/imageView1"
    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:layout_alignParentLeft="true"
    android:layout_below="@+id/imageView1"
    android:gravity="center"
    android:lines="2"
    android:text="TextView"
    android:textSize="18dp" />
</RelativeLayout>

```



Preguntas de repaso: *GridView.*

1.2. Fragments

1.2.1. Insertar *fragments* desde XML



Ejercicio paso a paso: *Un primer ejemplo con fragments.*

```

public class SelectorFragment extends Fragment {
    Activity actividad;
    GridView gridview;
    SelectorAdapter adaptador;

    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
        actividad = activity;
    }
}

```

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    View inflatedView = inflater.inflate(R.layout.fragment_selector,
        container, false);
    gridView = (GridView) inflatedView.findViewById(R.id.gridView);
    adaptador = new SelectorAdapter(actividad);
    gridView.setAdapter(adaptador);
    gridView.setOnItemClickListener(new OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View v, int
            position, long id) {
            Toast.makeText(actividad, "Seleccionado el elemento: " +
                position, Toast.LENGTH_SHORT).show();
        }
    });
    return inflatedView;
}
```

```
<?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" >
    <fragment
        android:id="@+id/headlines_fragment"
        android:name="com.example.audiolibros.fragments.SelectorFragment"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2" />
</LinearLayout>
```

```
public class MainActivity extends FragmentActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        SelectorAdapter.inicializarVector();
        setContentView(R.layout.activity_main);
    }
}
```



Ejercicio paso a paso: Implementando un segundo fragment.

```
public class DetalleFragment extends Fragment {
    public static String ARG_POSITION = "position";
    Activity actividad;
```

```

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    actividad = activity;
}

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

    Bundle args = getArguments();
    if (args != null) {
        int position = args.getInt(ARG_POSITION);
        setUpBookInfo(position, inflatedView);
    } else {
        setUpBookInfo(0, inflatedView);
    }
    return inflatedView;
}

private void setUpBookInfo(int position, View view) {
    BookInfo bookInfo = SelectorAdapter.bookVector.elementAt(position);
    TextView textView = (TextView) view.findViewById(
        R.id.textView1);
    TextView audiolibroTextView = (TextView) view.findViewById(
        R.id.textView2);
    ImageView imageView = (ImageView) view.findViewById(
        R.id.imageView1);
    imageView.setImageResource(bookInfo.resourceImage);
    textView.setText(bookInfo.autor);
    audiolibroTextView.setText(bookInfo.name);
    try {
        mediaPlayer.stop();
        mediaPlayer.release();
    } catch (Exception e) {}
}
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main_fragment_detalle"
    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_alignParentLeft="true"
        android:layout_alignParentRight="true"

```

```

        android:layout_alignParentTop="true"
        android:gravity="center"
        android:text="Large Text"
        android:textAlignment="center"
        android:textAppearance="?android:attr/textAppearanceLarge" />
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/textView1"
    android:gravity="center"
    android:text="Large Text"
    android:textAlignment="center"
    android:textAppearance="?android:attr/textAppearanceLarge" />
<ImageView
    android:id="@+id/imageView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView2"
    android:layout_centerHorizontal="true"
    android:src="@drawable/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" />

```

1.2.2. Insertar *fragments* desde código



Vídeo[Tutorial]: *Uso de recursos alternativos en Android.*



Ejercicio paso a paso: *Implementando un segundo fragment.*

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/fragment_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

```

```

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

```

1.2.3. Comunicación e intercambio de *fragments*



Ejercicio paso a paso: *Comunicación e intercambio de fragments.*

```

public interface OnGridViewListener {
    public void onItemSelected(int position);
}

```

```

try {
    mCallback = (OnGridViewListener) activity;
} catch (ClassCastException e) {
    throw new ClassCastException(activity.toString()
        + " ha de implementar OnGridViewListener");
}

```

```

gridview.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View v,
        int position, long id) {
        mCallback.onItemSelected(position);
    }
});
gridview.setOnItemLongClickListener(new OnItemLongClickListener() {
    public boolean onItemLongClick(AdapterView<?> parent, View v,
        final int position, long id) {
        AlertDialog.Builder builder = new AlertDialog.Builder(actividad);
        CharSequence[] items = { "Compartir", "Borrar ", "Insertar" };
        builder.setItems(items, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                switch (which) {
                    case 0:
                        Toast.makeText(actividad,
                            "Compartiendo en redes sociales",
                            Toast.LENGTH_LONG).show();
                        break;
                    case 1:
                        SelectorAdapter.bookVector.remove(position);
                        adaptador.notifyDataSetChanged();

```



```
        break;
    case 2:
        SelectorAdapter.bookVector
            .add(SelectorAdapter.bookVector.elementAt(0));
        adaptador.notifyDataSetChanged();
        break;
    }
}
});
builder.create().show();
return true;
}
});
```

```
@Override
public void onItemSelected(int position) {
    DetalleFragment detalleFragment = (DetalleFragment)
        getSupportFragmentManager().findFragmentById(R.id.detalle_fragment);
    if (detalleFragment != null) {
        detalleFragment.updateBookView(position);
    } else {
        DetalleFragment nuevoFragment = new DetalleFragment();
        Bundle args = new Bundle();
        args.putInt(DetalleFragment.ARG_POSITION, position);
        nuevoFragment.setArguments(args);
        FragmentTransaction transaction = getSupportFragmentManager()
            .beginTransaction();
        transaction.replace(R.id.fragment_container, nuevoFragment);
        transaction.addToBackStack(null);
        transaction.commit();
    }
}
```



Ejercicio paso a paso: Introducción de un MediaPlayer para reproducir Audiolibros.

```
MediaPlayer mediaPlayer;
MediaController mediaController;
```

```
public class DetalleFragment extends Fragment implements OnTouchListener,
    OnPreparedListener, MediaController.MediaPlayerControl
```

```
@Override public boolean onTouch(View view, MotionEvent event) {
    mediaController.show();
    return false;
}
```

```

@Override public void onPrepared(MediaPlayer mediaPlayer) {
    Log.d("Audiolibros", "Entramos en onPrepared de MediaPlayer");
    //mediaPlayer.start();
    mediaController.setMediaPlayer(this);
    mediaController.setAnchorView(actividad
        .findViewById(R.id.main_fragment_detalle));
}

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

@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() {
    return mediaPlayer.getCurrentPosition();
}

@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();
}
```

```
view.setOnClickListener(this);
Uri video = Uri.parse(bookInfo.url);
mediaPlayer = new MediaPlayer();
mediaPlayer.setOnPreparedListener(this);
try {
    mediaPlayer.setDataSource(actividad, video);
    mediaPlayer.prepareAsync();
} catch (IOException e) {
    Log.e("Audiolibros", "ERROR: No se puede reproducir " + video, e);
}
mediaController = new MediaController(actividad);
```



Preguntas de repaso: Fragments.

1.3. La barra de acciones (*ActionBar*)



Vídeo[Tutorial]: *Añadiendo un menú en Android.*



Ejercicio paso a paso: *Añadiendo un ActionBar a nuestra aplicación.*

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/menu_preferencias"
        android:icon="@android:drawable/ic_menu_preferences"
        android:orderInCategory="95"
        android:showAsAction="ifRoom"
        android:title="Preferencias"/>
    <item
        android:id="@+id/menu_ultimo"
        android:icon="@android:drawable/ic_menu_set_as"
        android:orderInCategory="100"
        android:showAsAction="ifRoom|withText"
        android:title="Ir a último"/>
    <item
        android:id="@+id/menu_buscar"
        android:actionViewClass="android.widget.SearchView"
        android:orderInCategory="90"
```

```

        android:icon="@android:drawable/ic_menu_search"
        android:showAsAction="always|collapseActionView"
        android:title="Buscar"/>
    <item
        android:id="@+id/menu_acerca"
        android:orderInCategory="110"
        android:showAsAction="never"
        android:title="Acerca de..."/>
</menu>

```

```

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

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menu_preferencias:
            Toast.makeText(this, "Preferencias", Toast.LENGTH_LONG).show();
            break;
        case R.id.menu_ultimo:
            goToLastVisited();
            break;
        case R.id.menu_buscar:
            break;
        case 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();
            break;
    }
    return false;
}

```

```

public void goToLastVisited() {
    SharedPreferences pref = getSharedPreferences(
        "com.example.audiolibros_internal", MODE_PRIVATE);
    int position = pref.getInt("position", -1);
    if (position >= 0) {
        onItemSelected(position);
    } 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("position", position);
editor.commit();
```

1.3.1. Añadiendo preferencias de usuario mediante PreferenceFragment



Ejercicio paso a paso: Añadiendo preferencias de usuario mediante PreferenceFragment.

```
public class PreferenciasFragment extends PreferenceFragment {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.preferences);
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
xmlns:android="http://schemas.android.com/apk/res/android">
    <CheckBoxPreference
        android:key="pref_autoreproducir"
        android:title="Reproducir automáticamente"
        android:summary="El audio se inicia automáticamente al abrirlo"
        android:defaultValue="true" />
</PreferenceScreen>
```

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

```
Intent i = new Intent(this, PreferenciasActivity.class);
startActivity(i);
```

1.4. Servicios de búsquedas



Ejercicio paso a paso: *Incorporando búsquedas en nuestra aplicación.*

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

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

```
<intent-filter>
    <action android:name="android.intent.action.SEARCH" />
</intent-filter>

<meta-data
    android:name="android.app.searchable"
    android:resource="@xml/searchable" />
```

```
<?xml version="1.0" encoding="utf-8"?>
<searchable xmlns:android="http://schemas.android.com/apk/res/android"
    android:hint="Buscar libro"
    android:label="@string/app_name">
</searchable>
```

```
@Override
protected void onNewIntent(Intent intent) {
    if (intent.getAction().equals(Intent.ACTION_SEARCH)) {
        busqueda(intent.getStringExtra(SearchManager.QUERY));
    }
}

public void busqueda(String query) {
    for (int i = 0; i < SelectorAdapter.bookVector.size(); i++) {
        BookInfo libro = SelectorAdapter.bookVector.elementAt(i);
        if (libro.name.toLowerCase().contains(query.toLowerCase())
            || libro.autor.toLowerCase().contains(query.toLowerCase())) {
```

```

        onItemSelected(i);
    }
}

```

1.5. Animaciones

1.5.1. Animaciones de vistas: transiciones entre actividades



Ejercicio paso a paso: *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>

```

```

<?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);
}
```



Ejercicio paso a paso: *Transiciones definidas en ActivityOptions.*

```
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());
}
```

1.5.1.1. Aplicando animaciones de vistas en Audiolibros



Ejercicio paso a paso: *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:
    Animation anim =AnimationUtils.LoadAnimation(actividad,R.anim.menguar);
    anim.setAnimationListener(SelectorFragment.this);
    v.startAnimation(anim);
    SelectorAdapter.bookVector.remove(position);
    //adaptador.notifyDataSetChanged();
    break;
```

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

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



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



Preguntas de repaso: *Animaciones de vistas.*

1.5.2. Animaciones de propiedades



Vídeo[Tutorial]: *Honeycomb Animation, Chet Haase.*

1.5.2.1. El motor básico de animación: *ValueAnimator*



Ejercicio paso a paso: *Una sencilla animación con ValueAnimator.*

```
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);
    }
}
```

1.5.2.2. Automatizando las animaciones: ObjectAnimator



Ejercicio paso a paso: *Una sencilla animación con ObjectAnimator.*

1.5.2.3. Combinando animaciones: AnimatorSet

1.5.2.4. 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>
```



Ejercicio paso a paso: *Definir una animación con XML.*

```
<?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 textView = (TextView) findViewById(R.id.text_view);
AnimatorSet set = (AnimatorSet) AnimatorInflater.loadAnimator(this,
    R.animator.animacion);
set.setTarget(textView);
set.start();

```

1.5.2.5. Nuevas propiedades de la clase View



Ejercicio paso a paso: *Uso de propiedades para animaciones con vistas.*

```

<?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>

```

1.5.2.6. Aplicando animaciones de propiedades en Audiolibros



Ejercicio paso a paso: 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:
    Animator anim = AnimatorInflater.loadAnimator(actividad,
                                                    R.animator.menguar);
    anim.addListener(SelectorFragment.this);
    anim.setTarget(v);
    anim.start();
    SelectorAdapter.bookVector.remove(position);
    //adaptador.notifyDataSetChanged();
    break;

```

```

if (convertView == null) {
    view = inflater.inflate(R.layout.elemento_selector, null);
} else {
    view = convertView;
    view.setScaleX(1);
    view.setScaleY(1);
}

```



Preguntas de repaso: Animaciones de propiedades.

CAPÍTULO 2.

Diseño personalizado de vistas

Por JESÚS TOMÁS

2.1. Algunos conceptos básicos



Vídeo[Tutorial]: *Los atributos de la clase View.*

2.2. Una vista como la composición de varias vistas



Ejercicio paso a paso: *Una vista para introducir una dirección de socket.*

```
<?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>
```



Preguntas de repaso: *Vistas.*

2.2.1. Creación de escuchadores de eventos



Vídeo[Tutorial]: *Escuchadores y manejadores de eventos en Android.*



Ejercicio paso a paso: *Añadir un escuchador de evento a la vista.*

```
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 ser 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();
```



Preguntas de repaso: *Escuchador de eventos.*

2.3. Modificación de vistas existentes



Ejercicio paso a paso: *Un EditText tuneado.*

```
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>
```



Ejercicio paso a paso: *Adaptando la vista a diferentes densidades gráficas.*

2.3.1. Algo más de información sobre TextView



Vídeo[Tutorial]: *Los atributos de la clase TextView.*



Ejercicio paso a paso: *Un EditText con palabras resaltadas.*

```
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);  
            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();
}
```



Preguntas de repaso: *Modificación de las vistas.*

2.4. Creación de nuevos atributos XML



Ejercicio paso a paso: *Añadiendo 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);
}

```



Preguntas de repaso: *Atributos XML.*

2.5. Una vista creada desde cero

2.5.1. Diseño y dibujo de la vista



Ejercicio paso a paso: *La vista ZoomSeekBar.*

```

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;
            }
        }
        v++;
    }
}

```

```

        canvas.drawLine(x, yIni + altoBar, x, y, reglaPaint);
    }
    v += escalaRaya;
}
}

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res/com.example.zoomseekbar"
    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



Ejercicio paso a paso: Reaccionando ante eventos de pantalla táctil en ZoomSeekBar.

```

// 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



Ejercicio paso a paso: *Implementando onMeasure en ZoomSeekBar.*

```

<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



Ejercicio paso a paso: *Implementando métodos get y set en ZoomSeekBar.*

```

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();
    }
}

```



Preguntas de repaso: *Creación de una vista desde cero.*

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:minResiseWidth="40dp"
    android:minResiceHeight="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. Diseñar el *layout* del *widget*

2.6.1.3. Crear una clase descendiente de *AppWidgetProvider*

2.6.1.4. 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>

```

2.6.1.5. 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



Ejercicio paso a paso: *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);
}
```



Ejercicio paso a paso: *Actualizando un widget de escritorio.*

```
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



Ejercicio paso a paso: *Lanzando una actividad al pulsar una vista.*

```

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

```



Ejercicio paso a paso: *Incrementando el contador al pulsar una vista.*

```

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();
    actualizaWidget(context, widgetId);
}
super.onReceive(context, intent);
}

```

2.6.5. Añadiendo una actividad de configuración



Ejercicio paso a paso: *Añadiendo una actividad de configuración.*

```

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

```

```

<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();
}
}

```



Preguntas de repaso: *Widget de escritorio.*

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



Vídeo[Tutorial]: *Events and the UI Thread in Android.*

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



Ejercicio paso a paso: *Cómo el hilo de la IU ejecuta los escuchadores de los botones.*

```

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>
```



Ejercicio paso a paso: *Cómo el hilo de la IU ejecuta escuchadores con un tiempo de ejecución prolongado.*

```
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();
}
```

Comentario [LM1]: la IGU (fem.!).

Comentario [LM2]: de la IU (fem.!).

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



Ejercicio paso a paso: *Un hilo en segundo plano intentando acceder a la IGU.*

```
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();
}
```



Preguntas de repaso: *El hilo de la interfaz de usuario en Android.*

3.3. La clase AsyncTask



Vídeo[Tutorial]: *The UI Thread and the class 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);
}
```



Ejercicio paso a paso: *Una aplicación que controla si un número es primo usando AsyncTask.*

```
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" />

```



Práctica: *Buscar números primos en un intervalo.*



Solución: *Buscar números primos en un intervalo.*

```
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>
```



Preguntas de repaso: *La clase AsyncTask.*

3.4. Animaciones con SurfaceView



Vídeo[Tutorial]: *Animations with SurfaceView.*

3.4.1. Programación con SurfaceViews

```
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); }
            } 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,...));
    }
}
```



Ejercicio paso a paso: *Una animación con SurfaceView.*

```
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);
            }
        } finally {
            if (c != null) surfaceHolder.unlockCanvasAndPost(c); }
        }
    }
}

```



Práctica: *Una pelota que reacciona a eventos de pantalla táctil.*



Solución: *Una pelota que reacciona a eventos de pantalla táctil.*

```

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;
}

```



Preguntas de repaso: *Animaciones con SurfaceView.*

Comentario [LM3]: no hay enlace

CAPÍTULO 4.

Comunicación con Bluetooth

Por CARSTEN VOGT
Traducción de MIGUEL GARCÍA PINEDA

4.1. Diferencias entre Bluetooth e Internet

4.2. Los pasos en la comunicación Bluetooth



Vídeo[Tutorial]: *Bluetooth basics in Android and Java SE.*



Vídeo[Tutorial]: *Bluetooth programming in Android.*

- `read()` y `write()` conocidos desde `java.io`.

4.2.1. Pasos de programación

```
BluetoothAdapter btAdapter;  
btAdapter = BluetoothAdapter.getDefaultAdapter();  
if (btAdapter == null)  
    Error: Device does not support Bluetooth
```

```
if (!btAdapter.isEnabled())  
    Message: Please enable BT and try again
```

```
String serviceName = "BluetoothService_1";
UUID uuid = UUID.fromString("12345678-4321-4111-ADDA-345127542950");
BluetoothServerSocket servSocket;
servSocket =
    btAdapter.listenUsingRfcommWithServiceRecord(serviceName,uuid);
```

```
BluetoothSocket commSocket;
commSocket = servSocket.accept();
```

```
BluetoothAdapter btAdapter;
btAdapter = BluetoothAdapter.getDefaultAdapter();
if (btAdapter == null)
    Error: Device does not support Bluetooth
```

```
if (!btAdapter.isEnabled())
    Message: Please enable BT and try again
```

```
BluetoothDevice partnerDevice = null;
// Get a list of all paired devices
Set<BluetoothDevice> bondedDevs = btAdapter.getBondedDevices();
if (bondedDevs.size() == 0)
    Error: No paired Bluetooth devices
// Search for a device with the name "Devname"
// (this is just an example name)
for (Iterator<BluetoothDevice> it = bondedDevs.iterator();
     it.hasNext();) {
    BluetoothDevice btd = it.next();
    if (btd.getName().equals("Devname") {
        partnerDevice = btd;
        break; }
if (partnerDevice==null)
    Error: No Bluetooth device of this name
```

```
// Use the UUID defined by the server
UUID uuid = UUID.fromString("12345678-4321-4111-ADDA-345127542950");
BluetoothSocket commSocket;
commSocket = partnerDevice.createRfcommSocketToServiceRecord(uuid);
commSocket.connect();
```

```
InputStream inStream;
OutputStream outStream;
InStream = new DataInputStream(commSocket.getInputStream());
outStream = new DataOutputStream(commSocket.getOutputStream());
```

```
byte[] data = ...;
```

```

outStream.write(data);
byte[] buffer = new byte[...];
int noReceived = inStream.read(buffer);

```

```

commSocket.close();

```

4.3. Algunas clases de utilidad autodefinidas

4.3.1. Clase de utilidad para un servidor

```

class BluetoothServerCV {
    private BluetoothAdapter btAdapter;
    // Bluetooth adapter for the device this app is running on
    private String serviceName;
    // Name of the service offered (for SDP entry)
    private UUID uuid;
    // UUID of the service offered (for SDP entry)
    private BluetoothServerSocket servSocket;
    // Server socket offered; an external client can connect to it
    private BluetoothSocket commSocket;
    // Socket for the communication
    // after the connection has been established
    private InputStream inStream;
    // InputStream of commSocket
    private OutputStream outStream;
    // OutputStream of commSocket
    /* Constructor. A call will wait for an incoming connection request
    of a client and then open a Bluetooth connection to this client,
    creating a new communication socket commSocket for this connection.
    Parameters:
        serviceName    - Name of the service offered
                        (if null, some default value will be used).
        uuidStringParam - UUID identifying the service offered
                        (if null, some default value will be used). */
    public BluetoothServerCV(String serviceName, String uuidString) {
        btAdapter = BluetoothAdapter.getDefaultAdapter();
        if (btAdapter == null) {
            Error: Bluetooth is not supported; return; }
        if (!btAdapter.isEnabled()) {
            Error: Bluetooth is not switched on; return; }
        if (serviceName==null||serviceName.equals(""))
            this.serviceName = some default name;
        else
            this.serviceName = serviceName;
        if (uuidString==null||uuidString.equals(""))
            this.uuid = UUID.fromString(some default UUID string);
        // UUID Strings have the format xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx
        // where the x's are hexadecimal digits
    }
}

```

```

// and y is a hexadecimal digit between 8 and F.
else
    this.uuid = UUID.fromString(uuidString);
try {
    // Create a server socket
    servSocket = btAdapter.listenUsingRfcommWithServiceRecord(
        this.serviceName, this.uuid);
    // Wait for a client connect() and accept it then
    commSocket = servSocket.accept();
    // Close the server socket
    // as no more connection requests shall be accepted
    servSocket.close();
    // Get input and output streams of the new communication socket
    inStream = new DataInputStream(commSocket.getInputStream());
    outStream = new DataOutputStream(commSocket.getOutputStream());
} catch (Exception e) {
    Error message; return; }
}
// Two methods to send data through commSocket
public void send(String data) {
    send(data.getBytes()); }
public void send(byte[] data) {
    try {
        outStream.write(data);
    } catch (IOException e) {
        Error message; return; }
}
/* A method to receive data through commSocket. The received data will
be returned in the buffer parameter. The return value of the method
will be the number of received bytes or -1 in case of an error. */
public int receive(byte[] buffer) {
    try {
        int numberOfBytesReceived = inStream.read(buffer);
        return numberOfBytesReceived;
    } catch (IOException e) {
        Error message; return -1; }
}
// A method to close commSocket
public void close() {
    try {
        commSocket.close();
    } catch (IOException e) {
        Error message; return; }
}
}
}

```

4.3.2. Clase utilidad para un cliente

```

class BluetoothConnectionToServerCV {

```



```

private BluetoothAdapter btAdapter;
// Bluetooth adapter for the device this app is running on
private BluetoothDevice partnerDevice;
// The partner (server) device to communicate with
private UUID uuid;
// UUID (Universally Unique Identifier) identifying the server
private BluetoothSocket commSocket;
// Socket connected to the server
// and to be used for the communication
private InputStream inStream;
// InputStream of commSocket
private OutputStream outStream;
// OutputStream of commSocket
/* Constructor. A call will open a Bluetooth connection to a server.
Afterwards, the attribute commSocket references the client socket.

Parameters:
    partnerDeviceName - Name of the device to connect to
                        (if null or no device of this name is
                         available, the first device in the list of
                         paired devices will be used).
    uuidStringParam   - UUID identifying the service of the partner
                        (if null, some default value is used). */
public BluetoothConnectionToServerCV(
    String partnerDeviceName, String uuidString) {
    btAdapter = BluetoothAdapter.getDefaultAdapter();
    if (btAdapter == null) {
        Error: Bluetooth is not supported; return; }
    if (!btAdapter.isEnabled()) {
        Error: Bluetooth is not switched on; return; }
    partnerDevice = determinePartnerDevice(partnerDeviceName);
    if (partnerDevice==null) {
        Error: No partner device available; return; }
    if (uuidString==null||uuidString.equals(""))
        this.uuid = UUID.fromString(some default UUID string);
    // UUID Strings have the format xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx
    // where the x's are hexadecimal digits
    // and y is a hexadecimal digit between 8 and F.
    else
        this.uuid = UUID.fromString(uuidString);
    try {
        // Create a socket
        commSocket =
            partnerDevice.createRfcommSocketToServiceRecord(uuid);
        // Connect it to the server
        commSocket.connect();
        // Get the input and output streams of commSocket
        inStream = new DataInputStream(commSocket.getInputStream());
        outStream = new DataOutputStream(commSocket.getOutputStream());
    } catch (Exception e) {
        Error message; return; }
}
/* A helper method to determine the partner device to communicate with:

```

- If there are no paired devices, null is returned.
- If there is a paired device with the name 'devname', this device is returned.
- If there are paired devices but none with the name 'devname', the first entry in the list of paired devices is returned. */

```

private BluetoothDevice determinePartnerDevice(String devname) {
    Set<BluetoothDevice> bondedDevs = btAdapter.getBondedDevices();
    if (bondedDevs.size()==0)
        return null;
    for (Iterator<BluetoothDevice> it = bondedDevs.iterator();
         it.hasNext();) {
        BluetoothDevice btd = it.next();
        if (btd.getName().equals(devname))
            return btd;
    }
    return bondedDevs.iterator().next();
}

// Two methods to send data through commSocket
public void send(String data) {
    send(data.getBytes()); }
public void send(byte[] data) {
    try {
        outputStream.write(data);
    } catch (IOException e) {
        Error message; return; }
}

/* A method to receive data through commSocket. The received data will
be returned in the buffer parameter. The return value of the method
will be the number of received bytes or -1 in case of an error. */
public int receive(byte[] buffer) {
    try {
        int numberOfBytesReceived = inputStream.read(buffer);
        return numberOfBytesReceived;
    } catch (IOException e) {
        Error message; return -1; }
}

// A method to close commSocket
public void close() {
    try {
        commSocket.close();
    } catch (IOException e) {
        Error message; return; }
}
}

```

4.3.3. Hilos de fondo

4.3.3.1. Hilo de fondo del servidor

```

class BluetoothServerConcurrentCV {
    private BluetoothManagerThreadCV btMgr;
    // The thread managing all calls (class definition see below)
    private BluetoothServerCV btServ;
    // The BluetoothServerCV object managed by the thread
    /* Constructor. A call will create and start a thread. This thread will
    create a BluetoothServerCV object and manage read/write requests for
    it. Parameters:
        serviceName      - Name of the offered service
                           (if null, some default value will be used).
        uuidStringParam  - UUID identifying the offered service
                           (if null, some default value will be used). */
    public BluetoothServerConcurrentCV(
        String serviceName, String uuidString) {
        btMgr = new BluetoothManagerThreadCV(serviceName,uuidString);
        btMgr.start();
    }
    // Two public methods to send data (return value = success)
    public boolean send(String data) {
        return send(data.getBytes());
    }
    public boolean send(byte[] data) {
        // Prepare the send order (as a Message object containing the data)
        Bundle bdl = new Bundle();
        bdl.putString("Operation","Send");
        bdl.putByteArray("Data", data);
        Message msg = new Message();
        msg.setData(bdl);
        // Submit the order to the managing thread
        boolean success = btMgr.execute(msg);
        return success;
    }
    /* A public method to receive data.
    Parameters:
        handler - A handler of the thread that calls this method. The
                  handler will handle the received data for this thread:
                  After a successful read() from the Bluetooth socket, the
                  btMgr thread will call handleMessage() of this handler
                  with the received data. The parameter of this call will be
                  a Message object containing a bundle with two entries:
                      ("NumberOfBytes",int): The number of valid bytes trans-
                                              ferred in the byte array below.
                      ("Data",byte[]): A byte array with the received data.
        maxNo   - The maximum number of bytes that can be received.
    Return value: Success of the operation. */
    public boolean receive(BluetoothCallbackHandlerCV handler, int maxNo) {
        // Prepare the receive order (as a Message object containing
        // a handler to be executed on the received data)
        Bundle bdl = new Bundle();
        bdl.putString("Operation","Receive");
        bdl.putSerializable("Handler",handler);
        bdl.putInt("MaxNo",maxNo);
        Message msg = new Message();

```

```

msg.setData(bdl);
// Submit the order to the managing thread
boolean success = btMgr.execute(msg);
return success;
}

/* A second public method to receive data. It differs from the first
method in the parameter 'requestedNoOfBytes': Only after receiving
this number of bytes, the bytes received will be returned to the
caller. This might require a whole sequence of read operations
on the socket. */
public boolean receiveFully(
    BluetoothCallbackHandlerCV handler, int requestedNoOfBytes) {
    Bundle bdl = new Bundle();
    bdl.putString("Operation", "ReceiveFully");
    bdl.putSerializable("Handler", handler);
    bdl.putInt("RequestedNumberOfBytes", requestedNoOfBytes);
    Message msg = new Message();
    msg.setData(bdl);
    boolean success = btMgr.execute(msg);
    return success;
}

// A method to stop the btMgr thread. btMgr will close the socket.
public void close() {
    btMgr.cancel();
}

// Class definition for the managing thread
private class BluetoothManagerThreadCV extends Thread {
    private BluetoothOrderHandlerCV orderHandler;
    // Handler handling orders to this thread
    private String serviceName;
    // Name of the offered service
    private String uuidString;
    // UUID identifying the offered service
    BluetoothManagerThreadCV(String serviceName, String uuidString) {
        this.serviceName=serviceName;
        this.uuidString=uuidString;
    }
    // Run loop of the thread, realized by a Looper
    public void run() {
        // Create a message queue for orders to this thread
        Looper.prepare();
        // Create a handler handling orders to this thread
        orderHandler = new BluetoothOrderHandlerCV();
        // Create the communication endpoint
        btServ = new BluetoothServerCV(serviceName, uuidString);
        // Start the loop to take and handle orders
        Looper.loop();
    }
}

/* A method used to submit execution orders to this thread.
An order is defined by a Message object containing a bundle.
The bundle contains a String component "Operation" defining the
operation to be executed and a second component for the data.
Name and type of these data depend on the operation. */

```

```

public boolean execute(Message msg) {
    // If the orderHandler does not exist yet, wait at most
    // five seconds to give it a chance (a somewhat dirty hack).
    for (int i=0;i<10;i++)
        if (orderHandler==null)
            try { sleep(500); } catch (Exception e) {};
    // If the connection is still not open, give up.
    if (orderHandler==null) return false;
    orderHandler.sendMessage(msg);
    return true;
}
// A method to close the connection and stop the looping thread
public void cancel() {
    btServ.close();
    orderHandler.getLooper().quit();
}
// Class definition for the handler
// that shall execute the orders submitted to this thread
private class BluetoothOrderHandlerCV extends Handler {
    public void handleMessage(Message msg) {
        Bundle msgDataBundle = msg.getData();
        String operation = msgDataBundle.getString("Operation");
        // Execute a "Send" order
        if (operation.equals("Send"))
            btServ.send(msgDataBundle.getByteArray("Data"));
        // Execute a "Receive" order (see above comments on
        // receive() in BluetoothServerConcurrentCV)
        if (operation.equals("Receive")) {
            BluetoothCallbackHandlerCV callbackHandler =
                (BluetoothCallbackHandlerCV)
                    msgDataBundle.getSerializable("Handler");
            int maxNo = msgDataBundle.getInt("MaxNo");
            byte receivedData[] = new byte[maxNo];
            int noReceived = btServ.receive(receivedData);
            Bundle replyBundle = new Bundle();
            replyBundle.putInt("NumberOfBytes", noReceived);
            replyBundle.putByteArray("Data", receivedData);
            Message replyMessage = new Message();
            replyMessage.setData(replyBundle);
            callbackHandler.sendMessage(replyMessage);
        }
        // Execute a "ReceiveFully" order
        if (operation.equals("ReceiveFully")) {
            BluetoothCallbackHandlerCV callbackHandler =
                (BluetoothCallbackHandlerCV)
                    msgDataBundle.getSerializable("Handler");
            int reqNumberOfBytes =
                msgDataBundle.getInt("RequestedNumberOfBytes");
            byte receivedData[] = new byte[reqNumberOfBytes];
            int noReceived = 0;
            do {
                byte buffer[] = new byte[10000];
                int rec = btServ.receive(buffer);

```

```

        for (int i=0;i<rec;i++)
            receivedData[noReceived+i] = buffer[i];
        noReceived += rec;
    } while (noReceived<reqNumberOfBytes);
    Bundle replyBundle = new Bundle();
    replyBundle.putInt("NumberOfBytes", noReceived);
    replyBundle.putByteArray("Data", receivedData);
    Message replyMessage = new Message();
    replyMessage.setData(replyBundle);
    callbackHandler.sendMessage(replyMessage);
}
}
}
}
}
/* Class definition for the callback handler of the calling thread.
The handler is used to notify this thread about the result of a
Bluetooth operation. It is especially called after completion of a
concurrently executed Bluetooth read() operation to transfer the
received data to the receiver thread - the read() operation will
activate the handleMessage() method of the handler. The parameter of
this call will be a Message object containing a bundle with two
entries:
    ("NumberOfBytes",int): The number of valid bytes
                           transferred in the byte array below.
    ("Data",byte[]): A byte array with the received data.
Therefore, the programmer of the receiving thread must define
a subclass of BluetoothCallbackHandlerCV overwriting the
void handleMessage(Message msg) method. This method defines the
operations the receiving thread wants to execute on the received
data. */
public static abstract class BluetoothCallbackHandlerCV
    extends Handler implements Serializable { }
}

```

4.3.3.2. Hilo de fondo del cliente

```

class BluetoothConnectionToServerConcurrentCV {
    private BluetoothManagerThreadCV btMgr;
    // The thread managing this Bluetooth connection
    private BluetoothConnectionToServerCV btConn;
    // The Bluetooth connection that is managed by the thread
    /* Constructor. A call will create and start a thread. This thread will
    create a BluetoothConnectionToServerCV object and manage read/write
    requests for it.
    Parameters:
        partnerDeviceName - Name of the device to connect to
                           (if null or no device of this name is
                           available, the first device in the list of
                           paired devices will be used).
        uuidStringParam    - UUID identifying the service of the partner
                           (if null, some default value will be used). */
    public BluetoothConnectionToServerConcurrentCV(
        String partnerDeviceName, String uuidString) {

```

```

    btMgr = new BluetoothManagerThreadCV(partnerDeviceName,uuidString);
    btMgr.start();
}
// Two methods to send data (return value = success)
public boolean send(String data) {
    return send(data.getBytes());
}
public boolean send(byte[] data) {
    // Prepare the send order (as a Message object containing the data)
    Bundle bdl = new Bundle();
    bdl.putString("Operation","Send");
    bdl.putByteArray("Data", data);
    Message msg = new Message();
    msg.setData(bdl);
    // Submit the order to the managing thread
    boolean success = btMgr.execute(msg);
    return success;
}
/* A method to receive data. The parameter is a handler defined by the
calling thread to handle the received data. After a successful
read() from the Bluetooth socket, the management thread will call
handleMessage() of this handler with the received data.
The parameter of this call will be a Message object containing a
bundle with two entries:
    ("NumberOfBytes",int): The number of valid bytes transferred
                           in the byte array below.
    ("Data",byte[]): A byte array with the received data.
The return value indicates whether the operation has been
successful. */
public boolean receive(BluetoothCallbackHandlerCV handler) {
    // Prepare the receive order (as a Message object containing
    // a handler to be executed on the received data)
    Bundle bdl = new Bundle();
    bdl.putString("Operation","Receive");
    bdl.putSerializable("Handler",handler);
    Message msg = new Message();
    msg.setData(bdl);
    // Submit the order to the managing thread
    boolean success = btMgr.execute(msg);
    return success;
}
// A method to stop the thread and close the communication socket
public void close() {
    btMgr.cancel();
}
// Class definition for the managing thread
private class BluetoothManagerThreadCV extends Thread {
    private BluetoothOrderHandlerCV orderHandler;
    // Handler with the handleMessage() method
    // handling orders to this thread
    private String partnerDeviceName;
    // Name of the device to connect to
    private String uuidString;

```

```

// UUID identifying the service to connect to
BluetoothManagerThreadCV(
    String partnerDeviceName, String uuidString) {
    this.partnerDeviceName=partnerDeviceName;
    this.uuidString=uuidString;
}
// Run loop of the thread, realized by a Looper
public void run() {
    // Create a message queue for orders to this thread
    Looper.prepare();
    // Create a handler handling orders to this thread
    orderHandler = new BluetoothOrderHandlerCV();
    // Open a connection to the server
    btConn =
        new BluetoothConnectionToServerCV(partnerDeviceName,uuidString);
    // Start the loop to take and handle orders
    Looper.loop();
}
/* A method used to submit execution orders to this thread.
An order is defined by a Message object containing a bundle.
The bundle contains a String component "Operation" defining the
operation to be executed and a second component for the data. The
name and the type of this component depend on the operation. */
public boolean execute(Message msg) {
    // If the orderHandler does not exist yet,
    // wait at most five seconds to give it a chance
    for (int i=0;i<10;i++)
        if (orderHandler==null)
            try { sleep(500); } catch (Exception e){};
    // If the connection is still not open, give up
    if (btConn==null) return false;
    orderHandler.sendMessage(msg);
    return true;
}
// A method to close the connection and stop the looping thread
public void cancel() {
    btConn.close();
    orderHandler.getLooper().quit();
}
// Class definition for the handler
// that shall execute the orders submitted to this thread
private class BluetoothOrderHandlerCV extends Handler {
    public void handleMessage(Message msg) {
        Bundle msgDataBundle = msg.getData();
        String operation = msgDataBundle.getString("Operation");
        // Execute a "Send" order
        if (operation.equals("Send"))
            btConn.send(msgDataBundle.getBytes("Data"));
        // Execute a "Receive" order (see above comments on receive())
        // in BluetoothConnectionToServerConcurrentCV
        if (operation.equals("Receive")) {
            BluetoothCallbackHandlerCV callbackHandler =
                (BluetoothCallbackHandlerCV)

```



```

        msgDataBundle.getSerializable("Handler");
        byte receivedData[] = new byte[1024];
        int noReceived = btConn.receive(receivedData);
        Bundle replyBundle = new Bundle();
        replyBundle.putInt("NumberOfBytes", noReceived);
        replyBundle.putByteArray("Data", receivedData);
        Message replyMessage = new Message();
        replyMessage.setData(replyBundle);
        callbackHandler.sendMessage(replyMessage);
    }
}
}
}
/* Class definition for the callback handler of the calling thread
   (for detailed comments see class BluetoothServerConcurrentCV) */
public static abstract class BluetoothCallbackHandlerCV
    extends Handler implements Serializable { }
}

```

4.4. Comunicación entre dispositivos Android



Ejercicio paso a paso el ejercicio: *Un mensaje de texto a través de Bluetooth.*

```

public class BluetoothSender extends Activity {
    private BluetoothConnectionToServerConcurrentCV btConn;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        String devname = "PartnerDeviceName";
        String uuid = "36AE13EE-7CC3-4ABC-A060-B5E4D4317904";
        btConn =
            new BluetoothConnectionToServerConcurrentCV(devname,uuid);
    }
    public void send(View v) {
        EditText et = (EditText) findViewById(R.id.textToBeSent);
        String text = et.getText().toString();
        btConn.send(text);
    }
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button

```

```

        android:id="@+id/send"
        android:text="Send!"
        android:textSize="12pt"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:onClick="send" />
    <EditText
        android:id="@+id/textToBeSent"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:textSize="12pt"
        android:hint="Text to be sent" />
</LinearLayout>

```

```

public class BluetoothReceiver extends Activity {
    static EditText outputView;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        outputView = (EditText) findViewById(R.id.receivedText);
        String servname = "Receive Service";
        String uuid = "36AE13EE-7CC3-4ABC-A060-B5E4D4317904";
        BluetoothServerConcurrentCV btServ =
            new BluetoothServerConcurrentCV(servname,uuid);
        btServ.receive(new CallbackHandlerTextOutput(),4096);
    }
    public static class CallbackHandlerTextOutput extends
        BluetoothServerConcurrentCV.BlueetoothCallbackHandlerCV {
        public void handleMessage(Message msg) {
            byte[] receivedData = msg.getData().getByteArray("Data");
            int noBytes = msg.getData().getInt("NumberOfBytes");
            String recString =
                (new String(receivedData)).substring(0,noBytes);
            outputView.setText(recString);
        }
    }
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <EditText
        android:id="@+id/receivedText"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:textSize="12pt"
        android:hint="Received text will be shown here" />
</LinearLayout>

```



Preguntas de repaso: *Bluetooth en Android.*

4.5. La comunicación con los programas en Java SE



Ejercicio paso a paso: *Un mensaje de texto desde Java SE a Android a través de Bluetooth.*

```
import java.io.*;
import javax.microedition.io.*;
import javax.bluetooth.*;
public class BluetoothSender_JavaSE {
    static Object lock1 = new Object(), lock2 = new Object();
    // For synchronization purposes
    static RemoteDevice partnerDevice = null;
    // The device to send data to
    static String connectionURL = null;
    // URL of the connection to this device
    public static void main(String args[]) {
        // The code of main() will be added below
    }
}
```

```
static class MyDisclistener implements DiscoveryListener {
    // See text below
    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();
        // Agent to discover Bluetooth services and devices
        MyDiscListener lis = new MyDiscListener();
        // Discovery listener (see step 7)
        agent.startInquiry(DiscoveryAgent.GIAC,lis);
        // Start search for devices
        synchronized(lock1) { try { lock1.wait(); }
                               catch (InterruptedException iexc) {} }
        // Wait until a device has been found
        // and the listener method has been executed
        javax.bluetooth.UUID[] uuidSet = new javax.bluetooth.UUID[1];
        uuidSet[0] = new javax.bluetooth.UUID(
            "36AE13EE7CC34ABCA060B5E4D4317904",false);
        // UUID of the service on the Android device

        agent.searchServices(null,uuidSet,partnerDevice,lis);
        // Start search for the service specified above
        synchronized(lock2) { try { lock2.wait(); }
                               catch (InterruptedException iexc) {} }
        // Wait until the service has been found
        // and the listener method has been executed
    } catch (BluetoothStateException exc) {
        System.out.println("Exception: "+exc.getMessage()); }
    System.out.println("Connected to: "+connectionURL);
    try {
        StreamConnection sConn =
            (StreamConnection) Connector.open(connectionURL);
        // Open the Bluetooth connection to the device and service
        OutputStream outStream = sConn.openOutputStream();
        PrintWriter pwr = new PrintWriter(new OutputStreamWriter(outStream));
        pwr.write("Message");
        pwr.flush();
        System.out.println("Data sent");
        try {
            Thread.currentThread().sleep(2000);
            // To give the receiver time to read the data
            // (no internal buffering!)
        } 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("36AE13EE7CC34ABCA060B5E4D4317904",false);
String hostname = "examplehost";
// To be replaced with the actual name of the notebook
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");
}
```



Práctica: *Un dispositivo móvil como un mando a distancia.*



Solución: *Un dispositivo móvil como un mando a distancia.*

```
public class ColorSelection extends Activity {
    private BluetoothConnectionToServerConcurrentCV btConn;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.colorselection);
        String devname = "examplehost";
        // Replace this with the name of your notebook
        String uuid = "36AE13EE-7CC3-4ABC-A060-B5E4D4317904";
        btConn = new BluetoothConnectionToServerConcurrentCV(devname,uuid);
    }
    public void onDestroy() {
        super.onDestroy();
        try {
            btConn.send("End");
            btConn.close();
        } catch (Exception e) { }
    }
    // Listener method for the buttons:
    // Sends the button text to the Java SE program.
    // Note: Do not click on a button
    // before the connection has been established!
    public void sendData(View v) {
        try {
            btConn.send(((Button) v).getText().toString());
        } catch (Exception e) { }
    }
}
```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="12pt"
        android:text="Select Color:" />
    <Button
        android:id="@+id/buttonRed"
        android:text="Red"
        android:textSize="12pt"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:onClick="sendData" />
    <Button
        android:id="@+id/buttonGreen"
        android:text="Green"
        ... as above ... />
    <Button
        android:id="@+id/buttonBlue"
        android:text="Blue"
        ... as above ... />
</LinearLayout>

```

```

import java.io.*;
import java.awt.*;
import javax.swing.*;
import javax.microedition.io.*;
import javax.bluetooth.*;
public class BluetoothReceiver_ColorExample {
    public static void main(String[] args) throws IOException {
        UUID uuid = new UUID("36AE13EE7CC34ABCA060B5E4D4317904", false);
        String hostname = "examplehost";
        // Replace this with the name of your notebook
        String url = "btspp://localhost:" + uuid + ";name=" + hostname +
            ";authenticate=false;encrypt=false;";
        ColorFrame colorframe; // The class ColorFrame is defined below
        colorframe = new ColorFrame();
        StreamConnectionNotifier host =
            (StreamConnectionNotifier) Connector.open(url);
        StreamConnection conn = host.acceptAndOpen();
        DataInputStream din = new DataInputStream(conn.openInputStream());
        while (true) {
            byte[] incomingData = new byte[10];
            din.read(incomingData);
            String s = new String(incomingData);
            if (s.startsWith("Red")) colorframe.setColor(0x00FF0000);
            if (s.startsWith("Green")) colorframe.setColor(0x0000FF00);
            if (s.startsWith("Blue")) colorframe.setColor(0x000000FF);
        }
    }
}

```

```
    }  
}  
// JFrame containing a colored rectangle  
class ColorFrame extends JFrame {  
    ColorPanel colorpanel; // The class ColorPanel is defined below  
    ColorFrame() {  
        setSize(950,450);  
        setLocation(0,0);  
        colorpanel = new ColorPanel();  
        getContentPane().add(colorpanel);  
        setVisible(true);  
    }  
    void setColor(int color) {  
        colorpanel.setColor(color);  
        repaint();  
    }  
}  
class ColorPanel extends JPanel {  
    private int color = 0x00000000;  
    void setColor(int color) {  
        this.color = color;  
    }  
    public void paint(Graphics g) {  
        g.setColor(new Color(color));  
        g.fillRect(0, 0, getWidth(), getHeight());  
    }  
}
```



Preguntas de repaso: *Bluetooth en Java SE.*

CAPÍTULO 5.

Servicios en la nube

Por VICENTE CARBONELL

5.1. Introducción a los servicios en la nube

5.2. Notificaciones *push*

5.2.1. Servicio Google Cloud Messaging

5.2.2. Activar Google Cloud Messaging en Google API Console

5.2.3. Aplicación cliente Google Cloud Messaging



Ejercicio paso a paso: *Crear la aplicación Android para recibir notificaciones push.*

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

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

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

```
<receiver android:name="com.google.android.gcm.GCMBroadcastReceiver"
android:permission="com.google.android.c2dm.permission.SEND" >
    <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />
        <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
        <category android:name="org.example.appgcm" />
    </intent-filter>
</receiver>
```

```
<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>
```

```
public final class UtilidadesGCM {
    static final String SERVER_URL =
        "http://cursoandroid.hol.es/notificaciones/";
    //Identificador del proyecto en Google Console que usa el servicio GCM.
    // [TIENES QUE SUSTITUIRLO POR EL TUYO]
    static final String SENDER_ID = "1092916126939";
    static final String DISPLAY_MESSAGE_ACTION =
        "org.example.appgcm.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();
        }
    });
}
```

```
import static org.example.appgcm.UtilidadesGCM.*;
public class ActividadPrincipal extends Activity {
    @Override public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        GCMRegistrar.checkDevice(this);
        GCMRegistrar.checkManifest(this);
    }
};
```

```
public void registrarUsuarioGCM(View v) {
    mostrarMensaje(ActividadPrincipal.this, "Iniciando el registro...");
    final String regId =GCMRegistrar.getRegistrationId(ActividadPrincipal);
    if (regId.equals("")) {
        GCMRegistrar.register(ActividadPrincipal.this, SENDER_ID);
        mostrarMensaje(ActividadPrincipal, "Registrado en GCM...");
    } else {
        mostrarMensaje(ActividadPrincipal, "Ya estás registrado");
    }
}

public void desregistrarUsuarioGCM(View v) {
    final String regId =
        GCMRegistrar.getRegistrationId(ActividadPrincipal);
    if (!regId.equals("")) {
        GCMRegistrar.unregister(ActividadPrincipal);
    } else {
        mostrarMensaje(ActividadPrincipal, "No estás registrado");
    }
}
```

segundo, el identificador del proyecto que obtuvimos en Google Console.

```
import static org.example.appgcm.UtilidadesGCM.*;
public class GCMIntentService extends GCMBaseIntentService {
}
```

```
@Override protected void onError(Context context, String msgError) {
    mostrarMensaje(context, "Error:" + msgError);
}
```

```

@Override protected void onMessage(Context context, Intent intent) {
    String mensaje = intent.getExtras().getString("mensaje");
    mostrarMensaje(context, "Mensaje del servidor web:");
    mostrarMensaje(context, mensaje);
}

```

```

@Override protected void onRegistered(Context context, String regId) {
    registrarDispositivoEnServidorWeb(context, regId);
    mostrarMensaje(context, "Registrado con éxito!!");
}

```

```

@Override protected void onUnregistered(Context context, String regId) {
    desregistrarDispositivoEnServidorWeb(context, regId);
    mostrarMensaje(context, "Te has desregistrado con éxito!!");
}

```

```

private void registrarDispositivoEnServidorWeb(Context context,
                                                final String regId) {
    String serverUrl = SERVER_URL + "registrar.php";
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("iddevice", regId));
    params.add(new BasicNameValuePair("idapp", SENDER_ID));
    HttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost(serverUrl);
    try {
        httpPost.setEntity(new UrlEncodedFormEntity(params));
        HttpResponse response = httpClient.execute(httpPost);
        GCMRegistrar.setRegisteredOnServer(context, true);
        mostrarMensaje(context, "Registrado en servidor web");
    } catch (IOException e) {
        mostrarMensaje(context, "Error en el registro.");
    }
}

```

```

private void desregistrarDispositivoEnServidorWeb(Context context,
                                                final String regId) {
    String serverUrl = SERVER_URL + "desregistrar.php";
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("iddevice", regId));
    params.add(new BasicNameValuePair("idapp", SENDER_ID));
    HttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost(serverUrl);
    try {
        httpPost.setEntity(new UrlEncodedFormEntity(params));
        HttpResponse response = httpClient.execute(httpPost);
        GCMRegistrar.setRegisteredOnServer(context, false);
        mostrarMensaje(context, "Desregistrado en servidor web");
    }
}

```

```

    } catch (IOException e) {
        mostrarMensaje(context, "Error en el desregistro.");
    }
}

```

```

<service android:name=".GCMIntentService" />

```



Ejercicio paso a paso: *Añadir notificaciones en la barra de estado.*

```

mostrarAvisoBarraEstado(context, mensaje);

```

```

private static void mostrarAvisoBarraEstado(Context context,
                                           String message) {
    int icon = R.drawable.ic_launcher;
    long when = System.currentTimeMillis();
    NotificationManager notificationManager = (NotificationManager)
        context.getSystemService(Context.NOTIFICATION_SERVICE);
    Notification notification = new Notification(icon, message, when);
    String title = context.getString(R.string.app_name);
    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);
    notification.setLatestEventInfo(context, title, message, intent);
    notification.flags |= Notification.FLAG_AUTO_CANCEL;
    notificationManager.notify(0, notification);
}

```

```

if (getIntent().hasExtra("mensaje")) {
    Bundle extras = getIntent().getExtras();
    String mensaje = extras.getString("mensaje");
    mostrarMensaje(ActividadPrincipal.this, "Mensaje del servidor web:");
    mostrarMensaje(ActividadPrincipal.this, mensaje);
}

```



Vídeo[Tutorial]: *Las notificaciones de la barra de estado.*



Ejercicio paso a paso: *Sustituir los mensajes toast por texto en un TextView.*

```
<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);
}
```

2.

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

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

5.2.4. Aplicación servidor Google Cloud Messaging



Ejercicio paso a paso: *Crear la aplicación del servidor con PHP y MySQL [OPCIONAL].*

```
<?php
//Conexión a la base de datos
mysql_connect("nombre_servidor", "nombre_usuario", "password") or die
(mysql_error());

mysql_select_db("base_datos");

//Insertamos id de registro devuelto por el GCM.
mysql_query("INSERT INTO dispositivos (iddevice) VALUES
('".$_POST["iddevice"]."')") or die(mysql_error());

mysql_close();

?>
```

```
<?php
//Conexión a la base de datos
mysql_connect("nombre_servidor", "nombre_usuario", "password") or die
(mysql_error());

mysql_select_db("base_datos");

//Eliminamos el dispositivo basándonos en el id de registro del GCM.
$sql = "DELETE FROM dispositivos WHERE iddevice=
".$_POST["iddevice"]."';

mysql_query($sql) or die(mysql_error());
mysql_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
$conection = mysql_connect ($host, $user, $pass) or die ('Error al
conectar con el servidor'.mysql_error());
mysql_select_db($database) or die ('->Error seleccionando la base
de datos'.mysql_error());

if ( $_POST['mensaje'] != "" ) {
    $message = $_POST['mensaje'];
    //Cambiar por API key de acceso para server del Google Console
    $apiKey = "API_KEY_ACCESO_SERVER";
    $result=mysql_query("SELECT * FROM dispositivos");
    while($row = mysql_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' => 'ck_'. 'col_key',
            'data.mensaje' => $message,
            'data.title' => 'Enviar notificación');
        //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);
    };
}
?>

```




Preguntas de repaso: *Notificaciones push.*



Práctica: *Enviar notificaciones desde la aplicación.*

```
String serverUrl = SERVER_URL + "desregistrar.php";

Map<String, String> params = new HashMap<String, String>();
params.put("iddevice", regId);
params.put("idapp", SENDER_ID);
```

```
HttpClient httpClient = new DefaultHttpClient();
HttpPost httpPost = new HttpPost(serverUrl);
try {
    httpPost.setEntity(new UrlEncodedFormEntity(params));
    HttpResponse response = httpClient.execute(httpPost);
} catch (IOException e) {
    // Error
}
```

5.3. Almacenamiento en la nube

5.3.1. Almacenamiento en Google Drive

5.3.2. Google Drive API v2

5.3.2.1. 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
```

5.3.2.2. Habilitar el servicio Google Drive API

5.3.2.3. Autorizar el acceso a Google Drive



Ejercicio paso a paso: *Preparar la aplicación para usar la API de Google Drive.*

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



Ejercicio paso a paso: *Obtener acceso a Google Drive.*

```
public final class UtilidadesDrive {
    static Drive servicio=null;
    static GoogleAccountCredential credencial=null;
    static String nombreCuenta = null;
}
```

```

import static org.example.googledriveapp.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.main);
        credencial = GoogleAccountCredential.usingOAuth2(
            ActividadPrincipal.this, 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);
        }
    }

    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();
}

public void showToast(final String toast) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(getApplicationContext(), toast,
                Toast.LENGTH_SHORT).show();
        }
    });
}
}
}

```

5.3.2.4. Subir ficheros a Google Drive

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



Ejercicio paso a paso: *Subir un fichero a Google Drive.*

```

public void hacerFoto(View v) {
    if (nombreCuenta == null) {
        showToast("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) {
        showToast("Debes seleccionar una cuenta de Google Drive");
    } else {
        Intent seleccionFotografiaIntent = new Intent();
        seleccionFotografiaIntent.setType("image/*");
        seleccionFotografiaIntent.setAction(Intent.ACTION_GET_CONTENT);
        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 {
        java.io.File ficheroJava = new java.io.File
            (obtenerLocalizacionUltimaImagen());
        FileContent contenido = new FileContent("image/jpeg",
            ficheroJava);

        File ficheroDrive = new File();
        ficheroDrive.setTitle(ficheroJava.getName());
        ficheroDrive.setMimeType("image/jpeg");
        File ficheroSubido = servicio.files().insert(ficheroDrive,
            contenido).execute();

        if (ficheroSubido != null) {
            runOnUiThread(new Runnable() {
                public void run() {
                    Toast.makeText(getApplicationContext(), "¡Foto subida!",
                        Toast.LENGTH_LONG).show();
                }
            });
        }
    } catch (UserRecoverableAuthIOException e) {
        startActivityForResult(e.getIntent(), SOLICITUD_AUTORIZACION);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
});
t.start();
}

```

3.

```

case SOLICITUD_AUTORIZACION:
    if (resultCode == Activity.RESULT_OK) {
        guardarFicheroEnDrive();
    } else {
        Toast.makeText(getApplicationContext(),
            "El usuario no autoriza usar Google Drive", Toast.LENGTH_LONG).show();
    }
    break;

```



Preguntas de repaso: *Google Drive.*

5.4. Servicio de Backup de Google

5.4.1. Fundamentos

5.4.2. Declaración del agente de copia de seguridad en Manifest

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

5.4.3. 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>
```

5.4.4. BackupAgent

5.4.5. BackupAgentHelper

5.4.5.1. Copia de seguridad de SharedPreferences

```
public class miAgenteBackup extends BackupAgentHelper {
    // El nombre del archivo SharedPreferences
    static final String PREFS = "Preferencias";
    // Una clave para identificar univocamente la copia de seguridad
    static final String PREFS_BACKUP_KEY = "GoogleDrive";
    // Asigna un ayudante al agente de copia de seguridad
    @Override public void onCreate() {
        SharedPreferencesBackupHelper helper =
            new SharedPreferencesBackupHelper(this, PREFS);
        addHelper(PREFS_BACKUP_KEY, helper);
    }
}
```

5.4.5.2. Copia de seguridad de archivos de almacenamiento interno

```
public class miAgenteBackup extends BackupAgentHelper {
    // El nombre de los archivos
    static final String FILE_FOTOGRAFIAS = "fotografias.txt";
    static final String FILE_LUGARES = "lugares.xml";
    // Una clave para identificar unívocamente la copia de seguridad
    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);
    }
}
```

5.4.6. Comprobación de la versión al restaurar los datos

5.4.7. Solicitud de copia de seguridad y restauración

5.4.8. Un ejemplo paso a paso



Ejercicio paso a paso: Copia de seguridad de *SharedPreferences* para *GoogleDriveApp* con el Servicio de Backup de Google.

```
<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="AEdPqrEAAAAIJuQYD05cd9ikBIUDC4oTTIcc0Hj0Ai9jBTEHCA" />
```

```
public class MisPreferenciasBackupAgent extends BackupAgentHelper {
    static final String PREFS = "Preferencias";
    static final String PREFS_BACKUP_KEY = "GoogleDriveApp";

    @Override
```



```
public void onCreate(){  
    SharedPreferencesBackupHelper helper =  
        new SharedPreferencesBackupHelper(this, Prefs);  
    addHelper(Prefs_Backup_Key, helper);  
}  
}
```

```
private BackupManager backupManager;
```

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



Ejercicio paso a paso: *Prueba del agente de copia de seguridad.*

```
adb shell bmgr enable true
```

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

```
adb shell bmgr run
```

```
adb uninstall org.example.googleDriveApp
```

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



Preguntas de repaso: *Android Backup Service.*

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



Ejercicio paso a paso: *Crear la aplicación web 3 en Raya.*

```
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <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>
  <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>
      </table>
    </div>
  </div>
</body>
</html>
```

```

        </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>
<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>

```

```

<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>
    <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>
</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;
    }
    .td0{
        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 = "0";
    casilla.style.fontSize = (ancho * 0.8) + "px";
    if (buscaGanador('0')) {
        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



Ejercicio paso a paso: *Definición del manifest para hacer una aplicación web fuera de línea.*

CACHE MANIFEST

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

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



Preguntas de repaso: *Introducción a las aplicaciones web.*

6.2. Uso de WebView

6.2.1. Mostrar contenido web usando una intención



Ejercicio paso a paso: *Abrir contenido web mediante un intent.*

```
package org.example.aplicacionweb;

public class ActividadPrincipal extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.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. mostrar contenido web



Ejercicio paso a paso: *Abrir contenido web en línea en un WebView.*

```
<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.main);
    navegador = (WebView) findViewById(R.id.webkit);
    navegador.loadUrl("http://cursoandroid.hol.es/appweb/index.html");
}
```

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



Ejercicio paso a paso: *Abrir contenido web fuera de línea en un WebView.*

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

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

6.2.3. Aspectos básicos de un WebView

6.2.3.1. Evitar el reinicio de la actividad



Ejercicio paso a paso: *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">
```

6.2.3.2. Abrir los enlaces en el WebView



Ejercicio paso a paso: *Abrir los enlaces en el WebView.*

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

6.2.3.3. Opciones de inicio



Ejercicio paso a paso: *Habilitar JavaScript y deshabilitar el zoom.*

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

6.2.3.4. Barra de progreso



Ejercicio paso a paso: *Añadir 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);
        }
    }
});
```

Hemos utilizado anteriormente `WebViewClient` y ahora `WebChromeClient`. Los



Ejercicio paso a paso: *Añadir diálogo de carga.*

```
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();
    }
});
```

6.2.3.5. Navegación



Ejercicio paso a paso: *Añadir funcionalidad página siguiente, anterior y detener carga.*

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    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) {
        dialogo = new ProgressDialog(ActividadPrincipal.this);
        dialogo.setMessage("Cargando...");
        dialogo.setCancelable(true);
        dialogo.show();
        btnDetener.setEnabled(true);
    }

    @Override
    public void onPageFinished(WebView view, String url) {
        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);
        }
    }
});

```

6.2.3.6. Controlar el botón «Volver»



Ejercicio paso a paso: *Controlar el botón «Volver» del dispositivo.*

```

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

```

```
}  
}
```

6.2.3.7. Habilitar alertas JavaScript



Ejercicio paso a paso: *Habilitar alertas JavaScript.*

```
@Override public boolean onJsAlert(WebView view, String url,  
                                   String message, JsResult result) {  
    return super.onJsAlert(view, url, message, result);  
}
```

6.2.3.8. Gestión de errores



Ejercicio paso a paso: *Gestionar 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();  
}
```

6.2.3.9. Descargas



Ejercicio paso a paso: *Descarga de un fichero en la tarjeta SD.*

```
<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 = "";  
        HttpClient httpClient = new DefaultHttpClient();  
        HttpGet httpGet = new HttpGet(urlDescarga);  
        InputStream inputStream = null;
```

```

try {
    HttpResponse httpResponse = httpClient.execute(httpGet);
    BufferedHttpEntity bufferedHttpEntity =
        new BufferedHttpEntity(httpResponse.getEntity());
    inputStream = bufferedHttpEntity.getContent();
    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("?")));
    FileOutputStream fileOutputStream = new FileOutputStream(file);
    ByteArrayOutputStream byteArray = new ByteArrayOutputStream();
    byte[] buffer = new byte[1024];
    int len = 0;
    while (inputStream.available() > 0 &&
        (len = inputStream.read(buffer)) != -1) {
        byteArray.write(buffer, 0, len);
    }
    fileOutputStream.write(byteArray.toByteArray());
    fileOutputStream.flush();
    mensaje = "Guardado en: " + file.getAbsolutePath();
} 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();
            }
        }
    }).setNegativeButton("Cancelar",
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            dialog.cancel();
        }
    });
builder.create().show();
}
});

```

6.2.3.10. Conectividad



Ejercicio paso a paso: *Comprobación básica de 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.*

Comentario [LM4]: No funciona el enlace

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. Escalado

6.3.3. Densidad de pantalla del dispositivo

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



Ejercicio paso a paso: *Establecer la propiedades de Viewport.*

```
<!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" />
```

6.4. Aplicaciones híbridas



Ejercicio paso a paso: *Comunicación webview-Android y viceversa.*

```
public class InterfazComunicacion {
    Context mContext;

    InterfazComunicacion(Context c) {
        mContext = c;
    }

    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;
}

```



Ejercicio paso a paso: Personalizar el nombre del jugador 1.

```

<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(getBaseContext());
    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;
    }
}

```



Preguntas de repaso: *Aplicaciones híbridas.*

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

6.5.1. Phonegap



Ejercicio paso a paso: Usar PhoneGap en Eclipse que utilice la cámara.

```
import org.apache.cordova.*;
```

```
public class ActividadPrincipal extends DroidGap {
```

```
@Override public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    super.loadUrl("file:///android_asset/www/index.html");
}
```

```
<supports-screens
    android:anyDensity="true"
    android:largeScreens="true"
    android:normalScreens="true"
    android:smallScreens="true"
    android:resizeable="true"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission
    android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.BROADCAST_STICKY"/>
<uses-permission android:name="android.permission.CAMERA"/>
```

```
android:configChanges="orientation|keyboardHidden"
```

```
<?xml version="1.0" encoding="utf-8"?>
<plugins>
  <plugin name="App" value="org.apache.cordova.App"/>
  <plugin name="GeoLocation" value="org.apache.cordova.GeoBroker"/>
  <plugin name="Device" value="org.apache.cordova.Device"/>
  <plugin name="Accelerometer" value="org.apache.cordova.AccelListener"/>
  <plugin name="Compass" value="org.apache.cordova.CompassListener"/>
  <plugin name="Media" value="org.apache.cordova.AudioHandler"/>
  <plugin name="Camera" value="org.apache.cordova.CameraLauncher"/>
  <plugin name="Contacts" value="org.apache.cordova.ContactManager"/>
  <plugin name="File" value="org.apache.cordova.FileUtils"/>
  <plugin name="NetworkStatus" value="org.apache.cordova.NetworkManager"/>
  <plugin name="Notification" value="org.apache.cordova.Notification"/>
  <plugin name="Storage" value="org.apache.cordova.Storage"/>
  <plugin name="Temperature" value="org.apache.cordova.TempListener"/>
  <plugin name="FileTransfer" value="org.apache.cordova.FileTransfer"/>
  <plugin name="Capture" value="org.apache.cordova.Capture"/>
  <plugin name="Battery" value="org.apache.cordova.BatteryListener"/>
  <plugin name="SplashScreen" value="org.apache.cordova.SplashScreen"/>
</plugins>
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Capturar foto</title>
    <script type="text/javascript" charset="utf-8" src="cordova-2.4.0.js">
    </script>
    <script type="text/javascript" charset="utf-8">
      var tipoDestino; // Tipo de datos (camara)

      document.addEventListener("deviceready",dispositivoListo,false);

      function dispositivoListo() {
        tipoDestino=navigator.camera.DestinationType;
      }

      function capturarFoto() {
        navigator.camera.getPicture(exitoFoto, errorFoto, { quality: 50,
          destinationType: tipoDestino.DATA_URL});
      }

      function exitoFoto(datosImagen) {
        var imagen = document.getElementById('imagen');
        imagen.src = "data:image/jpeg;base64," + datosImagen;
      }

      function errorFoto(mensaje) {
        alert('Error por: ' + mensaje);
      }
    </script>
  </head>
  <body>
    <div>
      <img alt="Capturar foto" data-bbox="100 800 200 850" id="imagen" />
    </div>
  </body>
</html>
```

```

</script>
</head>
<body>
  <button onclick="capturarFoto();">Capturar foto</button> <br>
  <img style="width:200px; border: 1px solid #333;" id="imagen"
      src="" />
</body>
</html>

```

```

<script type="text/javascript" charset="utf-8" src="cordova-2.4.0.js">
</script>

```

6.5.2. JQuery Mobile

```

<link rel="stylesheet"
  href="http://code.jquery.com/mobile/1.3.0/jquery.mobile-1.3.0.min.css"/>
<script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
<script src="http://code.jquery.com/mobile/1.3.0/jquery.mobile-
1.3.0.min.js"></script>

```

6.5.2.1. 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/1.3.0/jquery.mobile-1.3.0.min.css"/>
  <script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
  <script src=
    "http://code.jquery.com/mobile/1.3.0/jquery.mobile-1.3.0.min.js">
  </script>
</head>
<body>
<div data-role="page">
  <div data-role="header">
    <h1>Mi página</h1>
  </div>
  <div data-role="content">
    <p> Hola mundo </p>
  </div>
</div>
</body>
</html>

```



Ejercicio paso a paso: *Crear una página web con JQueryMobile.*

```
<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.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" />
```

6.5.2.2. Añadir contenido

6.5.2.3. 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>
```

6.5.2.4. 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>
```

6.5.2.5. Crear un botón

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

6.5.2.6. Temas



Ejercicio paso a paso: *Temas con JQueryMobile.*

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

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



Ejercicio paso a paso: *Juego 3 en Raya con jQuery Mobile.*

```
<!DOCTYPE HTML5>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.0/
    jquery.mobile-1.3.0.min.css" />
  <script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
  <script src="http://code.jquery.com/mobile/1.3.0/
    jquery.mobile-1.3.0.min.js"></script>
  <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 data-role="page" data-theme="e" id="inicio">
    <div data-role="header">
      <h1>3 en Raya</h1>
    </div>
    <div data-role="content">
      <div style="width:50%; margin-left:auto; margin-right:auto;">
        <table align="center">
          <tr>
            <td>
```

```

        
    </td>
</tr>
<tr>
    <td align="center">
        <a href="http://www.androidcurso.com">Android Curso</a>
    </td>
</tr>
</table>
<a href="#partida" data-role="button" data-transition="pop"
    onClick="iniciarJuego();">Jugar</a><br>
<a href="#ayuda" data-role="button" data-transition=
    "pop">Ayuda</a>
</div>
</div>
</div>
<!-- /Pág. Inicio -->

<!-- Pág. Partida -->
<div data-role="page" data-theme="e" id="partida">
    <div data-role="header">
        <h1 id="turno">Turno Jugador 1</h1>
    </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>
                        <td name="2_2" id="2_2" value=" "
                            onClick="click_celda('2_2');"></td>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
</div>

```

```

        </table>
    </div>
</div>

<div data-role="footer">
    <a href="#inicio" data-role="button" data-transition="pop"
        data-icon="home">Inicio</a>
    <a href="#" data-role="button" data-transition="pop"
        onClick="iniciarJuego();" data-icon="refresh">Reiniciar</a>
</div>
</div>
<!-- /Pág. Partida -->

<!-- Pág. Ayuda -->
<div data-role="page" data-theme="e" id="ayuda">
    <div data-role="header">
        <h1>Ayuda</h1>
    </div>
    <div data-role="content">
        <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 data-role="footer">
        <a href="#inicio" data-role="button" data-transition="pop"
            data-icon="home">Inicio</a>
    </div>
</div>
<!-- /Pág. Ayuda -->
</body>
</html>

```

```

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

```



Preguntas de repaso: *Alternativas a la programación independiente de la plataforma para móviles*

CAPÍTULO 7.

Programación en código nativo

Por MIGUEL GARCÍA PINEDA

7.1. Android NDK



Preguntas de repaso: *Android NDK.*

Comentario [LM5]: no hay enlace

7.2. Instalación de Android NDK

7.2.1. Instalación Android NDK en Windows

7.2.2. Instalación Android NDK en Linux



Ejercicio paso a paso: *Instalación de Android NDK en Linux (Debian 7).*

```
ANDROID_SDK="/home/crs03/Android/AndroidSDK/sdk"
ANDROID_NDK="/home/crs03/Android/AndroidNDK"
JAVA_HOME="/usr/lib/jvm/java-1.7.0-openjdk-i386"
```

```
PATH=$PATH:$JAVA_HOME/bin:$ANDROID_SDK/tools:$ANDROID_SDK/platform-
tools:$ANDROID_NDK
```

```
export PATH JAVA_HOME ANDROID_SDK ANDROID_NDK
```

7.3. Funcionamiento y estructura de Android NDK

```
static {  
    System.loadLibrary ("Fichero");  
}
```

7.3.1. Desarrollo práctico de Android NDK

7.3.2. Situación del código fuente nativo



Ejercicio paso a paso: *Comprobación y exploración del directorio jni del ejemplo san-angeles.*

7.3.2.1. Fichero *Android.mk*



Ejercicio paso a paso: *Comprobación y exploración del archivo Android.mk del ejemplo san-angeles.*

```
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 \  
    app-android.c \
```

```
LOCAL_LDLIBS := -lGLv1_CM -ldl -llog
```

```
include $(BUILD_SHARED_LIBRARY)
```

7.3.2.2. Fichero *Application.mk* (opcional)



Ejercicio paso a paso: *Evaluación de una archivo Application.mk.*

```
# The ARMv7 is significantly faster due to the use of the hardware FPU  
APP_ABI := armeabi armeabi-v7a  
APP_PLATFORM := android-8
```

7.3.2.3. La herramienta *ndk-build*

```
cd $PROYECTO  
ndk-build
```



Ejercicio paso a paso: *Prueba de las opciones de la herramienta de construcción ndk-build.*

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



Preguntas de repaso: *Funcionamiento y estructura de Android NDK.*

7.4. Interfaz entre JAVA y C/C++ (JNI)

7.4.1. Librerías de enlace estático y dinámico

7.4.2. 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);
```

7.4.3. Desarrollo paso a paso de un programa mediante JNI (I)



Ejercicio paso a paso: *Desarrollo de la aplicación nativa HolaMundoNDK 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>
```

7.4.3.1. Declaración del método nativo y creación del archivo *Android.mk*

```
static {
    System.loadLibrary("libreria");
}
public native String nombreMetodo();
```



Ejercicio paso a paso: *Desarrollo de la aplicación nativa HolaMundoNDK mediante JNI (I) - continuación.*

```
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)

```

7.4.3.2. Creación del fichero de cabecera nativo

7.4.3.3. Implementación del método nativo



Ejercicio paso a paso: *Desarrollo de la aplicación nativa HolaMundoNDK mediante JNI (I) – continuación.*

```

#include "com_holamundondk_HolaMundoNDK.h"
JNIEXPORT jstring Java_com_holamundondk_HolaMundoNDK_dameDatos (JNIEnv *
env, jobject this) {
    return (*env)->NewStringUTF(env,"App nativa");
}

```

7.4.4. Acceso a métodos Java desde código nativo (JNI *call-back*)

7.4.4.1. 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,
...);
```

7.4.4.2. 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, ...);
```

7.4.4.3. Invocar constructores



Ejercicio paso a paso: *Desarrollo de la aplicación nativa HolaMundoNDK mediante JNI (II).*

```
...
<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)
```

```
/*
 * 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 <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 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 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;
}

```



Preguntas de repaso: *Interfaz entre Java y C/C++ (JNI).*

7.5. Rendimiento de aplicaciones con código nativo



Ejercicio paso a paso: *Desarrollo de una aplicación para comprobar el rendimiento del 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);
}

```



```
}  
}
```

```
include $(CLEAR_VARS)  
  
LOCAL_MODULE := fibonacci  
LOCAL_SRC_FILES := fibonacci.c  
  
include $(BUILD_SHARED_LIBRARY)
```

```
...  
/*  
 * 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_fibonacciindk_FibonacciNDK_fibonacciNativoR
    (JNIEnv *env, jclass obj, jint n) {
    return fibonacciNativoR(n);
}

JNIEXPORT jlong JNICALL Java_com_fibonacciindk_FibonacciNDK_fibonacciNativoI
    (JNIEnv *env, jclass obj, jint n) {
    return fibonacciNativoI(n);
}

```

7.6. Procesado de imagen con código nativo



Ejercicio paso a paso: *Desarrollo de una aplicación de procesamiento 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="Grisés"
        android:onClick="onConvertirGrisés" />
    </LinearLayout>
</LinearLayout>

```

```

public class ImgProcesadoNDK extends Activity {
    private String tag = "ImgProcesadoNDK";
    private Bitmap bitmapOriginal = null;
    private Bitmap bitmapGrisés = null;
    private ImageView ivDisplay = null;

    static {
        System.loadLibrary("imgprocesadondk");
    }
    public native void convertirGrisés(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, "Resetear Imagen");
        ivDisplay.setImageBitmap(bitmapOriginal);
    }

    public void onConvertirGrisés(View v) {
        Log.i(tag, "Conversion a escala de grises");
        bitmapGrisés = Bitmap.createBitmap(bitmapOriginal.getWidth(),
                                            bitmapOriginal.getHeight(), Config.ALPHA_8888);
        convertirGrisés(bitmapOriginal, bitmapGrisés);
        ivDisplay.setImageBitmap(bitmapGrisés);
    }
}

```

```

LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)
LOCAL_MODULE := imgprocesadondk
LOCAL_SRC_FILES := imgprocesadondk.c
LOCAL_LDLIBS := -llog -ljnigraphics

include $(BUILD_SHARED_LIBRARY)

```

```

#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;
} argb;

/*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, &pixelsgris))<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 *) pixelsgris;
        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);
}

```



Práctica: *Mi InstagramNDK*.

```

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. Preliminares

8.1.1.1. Darse de alta en Facebook como desarrollador

8.1.1.2. SDK de Facebook para Android

8.1.1.3. Configurando nuestra aplicación

```
keytool -exportcert -alias androiddebugkey -keystore %HOMEPATH%\android\
debug.keystore | openssl sha1 -binary | openssl base64
Escriba la contraseña del almacén de claves: android
```

8.1.2. Nuestro proyecto Android

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.cuandrav.pruebafacebooksdk"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="8" android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <application
        android:allowBackup="true" android:icon="@drawable/ic_launcher"
```

```

android:label="@string/app_name"
android:theme="@style/AppTheme" >
<activity
android:name="org.cuandrav.pruebafacebooksdk.MainActivity"
android:label="@string/app_name" >
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<meta-data android:name="com.facebook.sdk.ApplicationId" an-
droid:value="@string/app_id"/>
<activity android:name="com.facebook.LoginActivity"></activity>
</application>
</manifest>

```

8.1.3. Aplicación de ejemplo

```

public class GestorTareasPendientes {
    public interface TareaPendiente {
        boolean hacerla(); // true: si está terminada y se puede eliminar
    }
    private ArrayList<TareaPendiente> lasTareasPendientes;

    public GestorTareasPendientes() {
        lasTareasPendientes = new ArrayList<TareaPendiente>();
    }

    public void vaciar() {
        lasTareasPendientes.clear();
    }

    public void anyadirTarea (TareaPendiente t) {
        lasTareasPendientes.add(t);
    }

    public void ejecutarTareas() {
        int cuantas = lasTareasPendientes.size();
        for (int i=cuantas-1; i>=0; i--) {
            boolean hecha = lasTareasPendientes.get(i).hacerla();
            if (hecha) {
                lasTareasPendientes.remove(i);
            }
        }
    }
}

```

```

lasTareasPendientes.anyadirTarea(
    new GestorTareasPendientes.TareaPendiente() {
        @Override public boolean hacerla() {
            if (!tengoPermisoParaPublicar()) {

```



```
        return false;
    }
    Request petition = Request.newStatusUpdateRequest(
        Session.getActiveSession(), mensaje, null);
    petition.executeAsync();
    return true;
}
});
```

```
final List<String> permisoParaPublicar = Arrays.asList("publish_actions");
laSesion.requestNewPublishPermissions(new Session.NewPermissionsRequest(
    this, permisoParaPublicar));
```

```
public class MainActivity extends Activity
    implements Session.StatusCallback {
    private GestorTareasPendientes lasTareasPendientes =
        new GestorTareasPendientes();
    private TextView elTextoDeBienvenida;
    private Button botonHacerLogin;
    private Button botonLogOut;
    private TextView textoConElMensaje;
    private Button botonCompartir;
    // el "ayudador" para saber los cambios de estado en la sesión de facebook
    private UiLifecycleHelper elFacebookLifeCycleHelper;

    // al implementar Session.StatusCallback, elFacebookLifeCycleHelper
    // llamará a este método
    @Override public void call(Session session, SessionState state,
        Exception exception) {
        Log.d("cuandrav.call", "CAMBIO DE ESTADO a=" + state.name()
            + ", sesion abierta=" + state.isOpened());
        actualizarVentanita();
        lasTareasPendientes.ejecutarTareas();
    }

    private void cerrarSesion() {
        Session laSesion = Session.getActiveSession();
        if (laSesion == null) {
            return;
        }
        laSesion.close();
        lasTareasPendientes.vaciar();
    }

    private boolean hayRed() {
        ConnectivityManager connectivityManager = (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo activeNetworkInfo =
            connectivityManager.getActiveNetworkInfo();
        return activeNetworkInfo != null && activeNetworkInfo.isConnected();
    }
}
```

```

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

@Override public void onPause() {
    super.onPause();
    elFacebookLifecycleHelper.onPause(); //
}

@Override public void onDestroy() {
    super.onDestroy();
    elFacebookLifecycleHelper.onDestroy(); // avisarle
}

@Override public void onStop() {
    super.onStop();
    lasTareasPendientes.ejecutarTareas();
}

@Override protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d("cuandrav.onCreate", "onCreate() llamado");
    // creo el UI LifecycleHelper dándole el callback (yo, 2o param)
    // y llamo a evento onCreate() para que lo sepa
    elFacebookLifecycleHelper = new UiLifecycleHelper(this, this);
    elFacebookLifecycleHelper.onCreate(savedInstanceState);
    // obtengo referencias a los widgets en el layout
    elTextoDeBienvenida = (TextView) findViewById(R.id.elTextoDeBienvenida);
    botonHacerLogin = (Button) findViewById(R.id.boton_hacerLogin);
    botonLogout = (Button) findViewById(R.id.boton_Logout);
    textoConElMensaje = (TextView) findViewById(R.id.txt_mensajeFB);
    botonCompartir = (Button) findViewById(R.id.boton_EnviarAFB);
    cerrarSesion();
    Log.d("cuandrav.onCreate", "final onCreate() ");
}

private Session getLaSesion() {
    // devuelve null si no existe o está cerrada, o el ptr a ella si está
    // abierta obtengo la sesión activa
    Session laSesion = Session.getActiveSession();
    if (laSesion == null || !laSesion.isOpened() ) {
        return null;
    }
    return laSesion;
}

private void actualizarVentanita() {
    Log.d("cuandrav.actualizarVentanita", "empiezo");
    // obtengo la sesión activa
    Session laSesion = getLaSesion();
    if (laSesion == null) {
        Log.d("cuandrav.actualizarVentanita", "no hay sesion, deshabilito");
        // sesion con facebook cerrada
    }
}

```

```

        botonHacerLogin.setEnabled(true);
        botonLogout.setEnabled(false);
        textoConElMensaje.setEnabled(false);
        botonCompartir.setEnabled(false);
        elTextoDeBienvenida.setText("haz login");
    } else {
        Log.d("cuandrav.actualizarVentanita",
            "hay sesion habilito y hago request a /me");
        // sesion abierta
        botonHacerLogin.setEnabled(false);
        botonLogout.setEnabled(true);
        textoConElMensaje.setEnabled(true);
        botonCompartir.setEnabled(true);
        // hago un request a /me (pregunta sobre datos básicos del usuario
        // autenticado)
        Request.executeMeRequestAsync(laSesion,
            new Request.GraphUserCallback() {
                // callback para cuando llegue la respuesta:
                // escribo el nombre del usuario autenticado en un campo
                @Override
                public void onCompleted(GraphUser usuario, Response respuesta) {
                    Log.d("cuandrav.actualizarVentana()",
                        "request a /me terminado");
                    if (usuario != null) {
                        elTextoDeBienvenida.setText("bienvenido: "
                            + usuario.getName());
                    } // if
                    else {
                        Log.d("cuandrav.actualizarVentana",
                            "request a /me terminado pero usuario es null");
                        cerrarSesion();
                    }
                }
            });
    }
}

private boolean tengoPermisoParaPublicar() {
    // obtengo la sesión activa
    Session laSesion = getLaSesion();
    // devuelve null si no hay sesion o hay pero no esta abierta
    // si no hay sesión activa
    if (laSesion == null) {
        return false;
    }
    // si hay sesión, pero NO tengo permiso para publicar
    if (!laSesion.getPermissions().contains("publish_actions")) {
        return false;
    }
    return true;
}

private void pedirPermisoParaPublicar() {

```

```

if (tengoPermisoParaPublicar()) {
    return; // si ya tengo permiso, termino
}
// no tengo permiso para publicar, los voy a pedir
Log.d("cuandrav.permisos", "no tenía permisos, los pido");
// obtengo la sesión activa
Session laSesion = getLaSesion();
// no hay sesión activa
if (laSesion == null) {
    Log.d("cuandrav.permisos",
        "quiero pedir permiso para publicar, pero session es null/cerrada");
    return;
}
// pido permisos para publicar
final List<String> permisoParaPublicar = Arrays
    .asList("publish_actions");
try {
    laSesion.requestNewPublishPermissions(new
        Session.NewPermissionsRequest(this, permisoParaPublicar));
} catch (Exception ex) {
    Log.d("cuandrav.permisos", "EXCEPCION al pedir permisos !!! : "
        + ex.getMessage());
}
Log.d("cuandrav.permisos", "no tenía permisos, acabo de pedirlos");
}

public void boton_enviarTextoAFB_pulsado(View quien) {
    // cojo el mensaje que ha escrito el usuario
    final String mensaje = "msg:" + textoConElMensaje.getText() +
        ":" + System.currentTimeMillis();
    textoConElMensaje.setText(""); // borro lo escrito
    // cierro el soft-teclado
    InputMethodManager imm = (InputMethodManager)
        getSystemService (Context.INPUT_METHOD_SERVICE);
    imm.hideSoftInputFromWindow(textoConElMensaje.getWindowToken(), 0);
    if (!hayRed()) {
        Toast.makeText(this, "¿no hay red? No puedo publicar",
            Toast.LENGTH_LONG).show();
    }
    pedirPermisoParaPublicar();
    // acumulo una nueva tarea pendiente (ha de esperar hasta que hayan
    // permisos para publicar)
    lasTareasPendientes.anyadirTarea(new
        GestorTareasPendientes.TareaPendiente() {
        @Override public boolean hacerla() {
            if (!tengoPermisoParaPublicar()) {
                return false;
            }
            Request petition = Request.newStatusUpdateRequest(
                Session.getActiveSession(), mensaje, null );
            petition.executeAsync();
            return true;
        }
    })
}

```

```

    });
    // llamo a ejecutar tareas pendientes, puede que ya tenga el permiso
    lasTareasPendientes.ejecutarTareas();
}

public void boton_LogOut_pulsado(View quien) {
    // compruebo la red
    if (!hayRed()) {
        Toast.makeText(this, "¿no hay red? No puedo cerrar sesión",
            Toast.LENGTH_LONG).show();
    }
    cerrarSesion();
}

public void boton_hacerLogin_pulsado(View quien) {
    hacerLogin();
}

private void hacerLogin() {
    if (!hayRed()) {
        Toast.makeText(this, "¿no hay red? No puedo hacer login",
            Toast.LENGTH_LONG).show();
    }
    Session laSesion = getLaSesion();
    // devuelve null si no hay sesión o si hay pero está cerrada
    if (laSesion != null) { // Ya hay sesión, termino
        return;
    }
    // no había sesión: hago login
    Log.d("cuandrav.hacerLogin()",
        "hacerLogin(): voy a hacer login en Facebook");
    Session.openActiveSession(this, true, null); // abro sesión
    Log.d("cuandrav.hacerLogin()",
        "hacerLogin(): ya he pedido el login (openActiveSession())");
}

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

@Override public void onActivityResult(int requestCode, int resultCode,
    Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    // avisar a la sesión activa que otra actividad ha terminado
    Session.getActiveSession().onActivityResult(this, requestCode,
        resultCode, data);
}
}

```



Preguntas de repaso: Facebook.

Comentario [LM6]: No hay enlace

8.2. Android y Twitter

8.2.1. Preliminares

8.2.2. Configurando nuestra aplicación

```
// copiar aquí los valores CONSUMER_KEY y CONSUMER_SECRET
public static final String CONSUMER_KEY = "xxxxxx";
public static final String CONSUMER_SECRET = "xxxxxx";

// urls para pedir los distintos tokens (autorizaciones) necesarias
public static final String REQUEST_URL =
"http://api.twitter.com/oauth/request_token";
public static final String ACCESS_URL =
"http://api.twitter.com/oauth/access_token";
public static final String AUTHORIZE_URL =
"http://api.twitter.com/oauth/authorize";

final public static String OAUTH_CALLBACK_URL = "x-latify-oauth-
twitter://callback";
```

```
<intent-filter>
<action android:name="android.intent.action.VIEW" />
<category android:name="android.intent.category.DEFAULT" />
<category android:name="android.intent.category.BROWSABLE" />
<data android:host="callback" android:scheme="x-latify-oauth-twitter" />
</intent-filter>
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="org.cuandrav.pruebatwitter"
android:versionCode="1"
android:versionName="1.0" >
<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="8" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<application
android:allowBackup="true"
android:icon="@drawable/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme" >
<activity
android:name="org.cuandrav.pruebatwitter.MainActivity"
android:label="@string/app_name"
android:launchMode="singleTask">
<!-- para que se conserven las variables al volver a ser llamados -->
<intent-filter>
<action android:name="android.intent.action.MAIN" />
```

```

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
    <!-- para que nos avise el navegador cuando termina de autenticar -->
    <!-- simulamos ser un navegador que puede responder a host=callback
        scheme=x-latify-oauth-twitter -->
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:host="callback" android:scheme=
                                "x-latify-oauth-twitter"/>
    </intent-filter>
</activity>
</application>
</manifest>

```

8.2.3. Aplicación de ejemplo

```

public class MainActivity extends Activity {
    // copiar aquí los valores CONSUMER_KEY y CONSUMER_SECRET sacados
    // de la pestaña "Details" de nuestra aplicación en Twitter
    public static final String CONSUMER_KEY = "6mybg2dcdrwKEP1fWd9hXw";
    public static final String CONSUMER_SECRET =
        "gAAE60xMDMFcmYqX3FIFkHk4AwDzaymUR50oisTs";
    // urls para pedir los distintos tokens (autorizaciones) necesarias
    public static final String REQUEST_URL =
        "http://api.twitter.com/oauth/request_token";
    public static final String ACCESS_URL =
        "http://api.twitter.com/oauth/access_token";
    public static final String AUTHORIZE_URL =
        "http://api.twitter.com/oauth/authorize";
    // url donde pediremos al navegador que nos redirija cuando un usuario
    // se autentifica. Esa redirección con este url hace que en Android
    // sea capturada por esta misma aplicación (ver AndroidManifest.xml y
    // onNewIntent())
    final public static String OAUTH_CALLBACK_URL =
        "x-latify-oauth-twitter://callback";
    // objetos para negociar la autenticación (con OAuth)
    private CommonsHttpOAuthConsumer elConsumidorOAuth;
    private CommonsHttpOAuthProvider elProveedorOAuth;
    // punteros a los elementos gráficos
    private TextView elTextoDeBienvenida;
    private Button botonHacerLogin;
    private Button botonLogout;
    private TextView textoConElMensaje;
    private Button botonCompartir;

    @Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("cuandrav.onCreate", "empieza");
    }
}

```

Comentario [LM7]: no se lee con claridad

```

// obtengo referencias a los widgets en el layout
elTextoDeBienvenida = (TextView) findViewById(R.id.elTextoDeBienvenida);
botonHacerLogin = (Button) findViewById(R.id.boton_hacerLogin);
botonLogout = (Button) findViewById(R.id.boton_hacerLogout);
textoConElMensaje = (TextView) findViewById(R.id.txt_textoTweet);
botonCompartir = (Button) findViewById(R.id.boton_enviarTweet);
actualizarVentanita();
Log.d("cuandrav.onCreate", "acaba");
}

public void boton_hacerLogin_pulsado(View quien) {
    Log.d("cuandrav.boton_hacerLogin_pulsado", "empieza");
    // He de pedir el permiso para acceder a twitter
    // (de haberlo tenido el boton estaria deshabilitado, ver
    // actualizarVentanita() y no se hubiera producido este evento)

    try {
        Log.d("cuandrav.boton_hacerLogin_pulsado ",
            " creo consumidor y proveedor");
        // creo los objetos consumidor y proveedor para la negociación OAuth
        elConsumidorOAuth = new CommonsHttpOAuthConsumer(CONSUMER_KEY,
            CONSUMER_SECRET);
        elProveedorOAuth = new CommonsHttpOAuthProvider(REQUEST_URL,
            ACCESS_URL, AUTHORIZE_URL);

        // negociación OAuth:
        // 1. pido el token de petición (es una url)
        // Además doy el url de call back, para que la página de twitter
        // sepa a dónde redirigir tras la autenticación
        Log.d("cuandrav.boton_hacerLogin_pulsado ", " pido request token");
        // final String[] urlArray = {"vacío"};
        // como la petición retrieveRequestToken accede a la red
        // he de utilizar un AsyncTask
        AsyncTask<Void, Void, Void> at = new AsyncTask<Void, Void, Void>() {
            @Override protected Void doInBackground(Void... voids) {
                try {
                    String url = elProveedorOAuth.retrieveRequestToken(
                        elConsumidorOAuth, OAUTH_CALLBACK_URL);

                    // negociación OAuth
                    // 2. abro un navegador (ACTION_VIEW) y le digo que vaya al url
                    // (página de twitter) donde nos autenticamos (usuario y contraseña)
                    Intent intent = new Intent(Intent.ACTION_VIEW,
                        Uri.parse(url))
                        .setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP
                            | Intent.FLAG_ACTIVITY_NO_HISTORY
                            | Intent.FLAG_FROM_BACKGROUND);
                    Log.d("cuandrav.boton_hacerLogin_pulsado ",
                        " arranco navegador para autenticar");

                    startActivity(intent);
                } catch (Exception e) {
                    Log.d("cuandrav.boton_hacerLogin_pulsado, asyncTask",
                        "EXCEPCION: " + e.getMessage());
                }
            }
        };
        return null;
    }
}

```

Comentario [LM8]: un (masc.)


```

        // de momento nada más. Si todo va bien (nos autenticamos)
        // el proceso sigue por onNewIntent()
    }
};
at.execute();
Log.d("cuandrav.boton_hacerLogin_pulsado ",
    " un thread se está encargando de hacer login (pasos 1 y 2 de OAuth");
} catch (Exception e) {
    Log.d("cuandrav.boton_hacerLogin_pulsado", "EXCEPCION: " + e.getMessage());
}
// actualizarVentanita();
Log.d("cuandrav.boton_hacerLogin_pulsado", "termina");
}

// onNewIntent es llamado tras la autenticación en la página de twitter.
// El navegador redirige a alguien que pueda responder al intent
// "android.intent.action.VIEW"
// "android.intent.category.DEFAULT"
// "android.intent.category.BROWSABLE"
// data android:host="callback" android:scheme="x-latify-oauth-twitter"
// Eso lo hemos puesto en nuestro Manifest y es el truco
// para que esta aplicación se vuelva a activar tras la autenticación
// en el navegador. Además esta actividad en el manifest tiene
// android:launchMode="singleTask"
// para que no se rearranque y se conserven las variables de antes.
// Por tanto, ahora no va a ocurrir ni onCreate() ni onStart() sino
// onNewIntent() que es donde terminamos el proceso de autenticación.
@Override public void onNewIntent(Intent intento) {
    Log.d("cuandrav.onNewIntent", "empieza");
    super.onNewIntent(intento); // esto siempre
    // el objeto elTwitter, lo voy a obtener ahora
    elTwitter = null;
    // me aseguro que nos está re-arrancando el navegador
    Uri uri = intento.getData();
    if (uri != null && uri.getScheme().contains("x-latify-oauth-twitter")) {
        // Así es, ahora finalizo la autenticación
        Log.d("cuandrav.onNewIntent", "finalizo la autenticacion");
        Log.d("cuandrav.onNewIntent", " el uri que recibo:" + uri.toString());
        try {
            final Editor lasPreferencias = PreferenceManager
                .getDefaultSharedPreferences(this).edit();
            // para obtener el accessToken+secreto (autorización final)
            // necesito el oauth_verifier que me dan en el uri
            // al volver del navegador donde me he autenticado
            final String oauth_verifier =
                uri.getQueryParameter(OAuth.OAUTH_VERIFIER);
            // negociación OAuth
            // 3. obtengo el acces token, como accede a red
            // (retrieveAccessToken()) he de hacerlo en un thread separado
            AsyncTask<Void, Void, Void> at = new AsyncTask<Void, Void, Void>() {
                @Override protected Void doInBackground(Void... voids) {
                    try {
                        elProveedorOAuth.retrieveAccessToken(

```

```

        elConsumidorOAuth, oauth_verifier);
Log.d("cuandrav.onNewIntent",
    " hecho retrieveAccessToken:");
String token = elConsumidorOAuth.getToken();
String secret = elConsumidorOAuth.getTokenSecret();
Log.d("cuandrav.onNewIntent", " token=" + token
    + " secret=" + secret);
// guardo el token+secreto en preferencias
Log.d("cuandrav.onNewIntent",
    " guardo en preferencias token y secret ");
lasPreferencias.putString(OAuth.OAUTH_TOKEN, token);
lasPreferencias.putString(OAuth.OAUTH_TOKEN_SECRET, secret);
lasPreferencias.commit();
Log.d("cuandrav.onNewIntent",
    " guardo en preferencias token y secret: HECHO ");
// los guardo tambien en el consumidor
elConsumidorOAuth.setTokenWithSecret(token, secret);
Log.d("cuandrav.onNewIntent", " ya está ");
// llamo a autenticar() para que ponga la variable elTwitter.
// Es otro método porque si ya obtuve el token+secreto estaran
// guardos en las preferencias y pese a no haber pasado por aquí,
// podré regenerar "elTwitter" para poder pedir cosas a Twitter
regeneraVariableTwitter();
} catch (Exception ex) {
    Log.d("cuandrav.onNewIntent() asynctask ",
        " EXCEPCION:" + ex.getClass().toString()
        + " " + ex.getMessage());
}
return null;
}

protected void onPostExecute(Void ingnorar) {
    // tras obtener el accessToken
    // en thread ordinario (UI) actualizo la ventanita
    actualizarVentanita();
}
};
at.execute();
Log.d("cuandrav.onNewIntent",
    "un thread se está encargando de conseguir el access token");
} catch (Exception ex) {
    Log.d("cuandrav.onNewIntent", " EXCEPCION:"
        + ex.getClass().toString() + " " + ex.getMessage());
}
}
}

// variable con la que ya pido cosas conretas a Twitter
// se crea una vez tenemos el accessToken+secret
private Twitter elTwitter = null;

private boolean estoyAutenticado() {
    return elTwitter != null;
}

```

```

}

// regenerar la variable elTwitter que sirve para pedir cosas a Twitter
private Twitter regeneraVariableTwitter() {
    if (elTwitter != null) {
        return elTwitter; // si ya lo tengo, lo devuelvo
    }
    // no lo tengo, he de regenerarlo
    Log.d("cuandrav.regeneraVariableTwitter()", "empiezo");
    // obtengo las preferencias
    SharedPreferences prefs = PreferenceManager
        .getDefaultSharedPreferences(this);
    // y de ellas intonco sacar el accessToken+secreto
    String token = prefs.getString(OAuth.OAUTH_TOKEN, "");
    String secret = prefs.getString(OAuth.OAUTH_TOKEN_SECRET, "");
    Log.d("cuandrav.regeneraVariableTwitter()", "token y secret obtenidos: "
        + token);

    // regenero la variable elTwitter
    AccessToken a = new AccessToken(token, secret);
    Twitter twitter = new TwitterFactory().getInstance();
    twitter.setOAuthConsumer(CONSUMER_KEY, CONSUMER_SECRET);
    twitter.setOAuthAccessToken(a);
    Log.d("cuandrav.regeneraVariableTwitter()", "accessToken puesto");
    // ahora realmente compruebo simplemente con lo de getAccountSettings()
    // que las peticiones a Twitter funcionan
    // si no fuera así saltaría una excepción
    try {
        twitter.getAccountSettings();
        Log.d("cuandrav.regeneraVariableTwitter()", "SI autenticado: ");
        elTwitter = twitter;
        return elTwitter;
    } catch (TwitterException e) {
        Log.d("cuandrav.regeneraVariableTwitter()", "EXCEPCION: " +
            e.getMessage());

        // no he conseguido regenerar elTwitter
        elTwitter = null;
        return null;
    }
}

public void enviaTweet(String mensaje) throws Exception {
    // obtengo el twitter para poder enviar (me lo devuelve si ya lo tenía)
    Twitter twitter = regeneraVariableTwitter();
    if (twitter == null) {
        // si no tengo acceso a Twitter termino,
        Toast.makeText(this, "no estoy atentificado para twitear",
            Toast.LENGTH_LONG).show();
        Log.d("cuandrav.enviaTweet", "no estoy autenticado?");
        return;
    }
    // envío por fin el tweet
    Log.d("cuandrav.enviaTweet", " envio " + mensaje);
    twitter.updateStatus(mensaje);
}

```

```

    Log.d("cuandrav.enviaTweet", " habre enviado? " + mensaje);
}

private void actualizarVentanita() {
    Log.d("cuandrav.actualizarVentanita", "empiezo");
    if (!estoyAutenticado()) {
        Log.d("cuandrav.actualizarVentanita", "no autenticado, deshabilito");
        botonHacerLogin.setEnabled(true);
        botonLogout.setEnabled(false);
        textoConElMensaje.setEnabled(false);
        botonCompartir.setEnabled(false);
        elTextoDeBienvenida.setText("haz login");
    } else {
        Log.d("cuandrav.actualizarVentanita", "autenticado: habilito");
        botonHacerLogin.setEnabled(false);
        botonLogout.setEnabled(true);
        textoConElMensaje.setEnabled(true);
        botonCompartir.setEnabled(true);
        final Twitter t = regeneraVariableTwitter();
        // voy a averiguar el nombre del twitero autenticado,
        // como accede a red hay que hacerlo en un AsyncTask
        AsyncTask<Void, Void, String> at = new AsyncTask<Void, Void, String>(){
            @Override protected String doInBackground(Void... voids) {
                try {
                    return t.showUser(t.getId()).getName();
                } catch (Exception ex) {
                    Log.d("cuandrav.actualizarVentanita()",
                        "no puedo obtener el nombre? EXCEPCION: " + ex.getMessage());
                    return " no he encontrado tu nombre ";
                }
            }
        };

        protected void onPostExecute(String nombre) {
            // pero el cambiar el nombre de la vista, lo
            // ha de hacer el thread UI, cuando el otro thread termine
            elTextoDeBienvenida.setText("Bienvenido: " + nombre);
        }
    };
    at.execute();
    Log.d("cuandrav.actualizarVentanita()",
        "he enviado a un thread a por el nombre del twitero");
}

public void boton_enviarTweet_pulsado(View q) {
    // cojo el mensaje que ha escrito el usuario
    final String mensaje = "msg:" + textoConElMensaje.getText() + " :"+
        System.currentTimeMillis();

    // borro lo escrito
    textoConElMensaje.setText("");
    // cierro el soft-teclado
    InputMethodManager imm = (InputMethodManager)
        getSystemService(Context.INPUT_METHOD_SERVICE);

```

```

imm.hideSoftInputFromWindow(textoConElMensaje.getWindowToken(), 0);
// como el método para publicar (enviaTweet) acaba accediendo
// a la red, he de utilizar otro thread distinto del UI: AsyncTask
AsyncTask<String, Void, Void> at = new AsyncTask<String, Void, Void>() {
    @Override protected Void doInBackground(String... params) {
        try {
            Log.d("cuandrav.boton_enviarTweet_pulsado",
                "enviando desde asyncTask");

            enviaTweet(params[0]);
        } catch (Exception e) {
            Log.d("cuandrav.boton_enviarTweet_pulsado", "EXCEPCION: "
                + e.getMessage());
        }
        return null;
    }
};
at.execute(mensaje);
Log.d("cuandrav.boton_enviarTweet_pulsado",
    "un thread se está encargando de twitear");
}

public void boton_Logout_pulsado(View quien) {
    try {
        // quito los valores de las preferencias
        SharedPreferences prefs = PreferenceManager
            .getDefaultSharedPreferences(this);

        final Editor edit = prefs.edit();
        edit.remove(OAuth.OAUTH_TOKEN);
        edit.remove(OAuth.OAUTH_TOKEN_SECRET);
        edit.commit();
    } catch (Exception ex) {
        Log.d("cuandrav.boton_Logout_pulsado", "EXCEPCION: "+ex.getMessage());
    } finally {
        elTwitter = null; // borro el acceso a Twitter
        actualizarVentanita();
    }
}

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

```



Preguntas de repaso: *Twitter*.

Comentario [LM9]:

CAPÍTULO 9.

Ingeniería inversa en Android

Por JESÚS TOMÁS

9.1. El formato APK



Vídeo[Tutorial]: *AirDroid: qué es, cómo descargarlo y usarlo.*



Vídeo[Tutorial]: *La firma digital.*



Vídeo[Tutorial]: *Firmar una aplicación Android.*



Ejercicio paso a paso: *Estudio de la firma digital de una aplicación.*

```
Manifest-Version: 1.0
Created-By: 1.6.0_29 (Sun Microsystems Inc.)

Name: res/drawable/facebook_icon.png
SHA1-Digest: 9jTbeG4b8nJZ1Zss16UPfagqlUs=
```

Name: res/drawable/com_facebook_picker_list_longpressed.9.png
SHA1-Digest: /uU9+qqC9MjC0cwR3L+DskCy9xM=

Name: res/drawable/square_d.xml
SHA1-Digest: k1+o4e30jU9gil/1+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: 5UTm9fULZG3sQFPfSsTpoFL7Yro=

Name: res/drawable-ldpi/warply_notifications.png
SHA1-Digest: 4KFSug0Gx0l27lgbgNaqEUvi8Fk=

...

```
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
```



Preguntas de repaso: *El formato APK.*

Comentario [LM10]: o hay enlace No
hay enlace

9.2. Decompilando aplicaciones Android

9.2.1. La máquina virtual Dalvic

9.2.2. Decompilando aplicaciones Android



Ejercicio paso a paso: *Obtención del código Java de una aplicación.*

```
C:\C9>dex2jar\d2j-dex2jar classes.dex
dex2jar classes.dex -> classes-dex2jar.jar
```



Preguntas de repaso: *Decompilando aplicaciones Android.*

9.3. Modificando aplicaciones Android

9.3.1. Modificando recursos binarios de una aplicación



Ejercicio paso a paso: *Modificar recursos de una aplicación Android.*

APK MULTI-TOOL SETUP FILE

```
*****
1. Check for update
   This Will Check if there is a update to the main program
   For this option is not functional please visit
   http://apkmultitool.com for updates
*****
2. Installing Framework-Res
   This Will install one of the Framwork-Res Files for 1.x/2.x/3.x/4.x
   This Feature also will install any of the other Dependencies needed
   In order to use this feature make sure to drop all of the needed files
   into the other Folder or else this script will not find them
*****
3. Setup Directories
   This will setup the appropate directories needed by Apk Multi-Tools
   (This script only needs to be ran for first time users do not use if your
   Just updating from previous Version)
```

Simple Tasks Such As Image Editing	Advanced Tasks Such As Code Editing
0 Adb pull	9 Decompile apk
1 Extract apk	10 Decompile apk (with dependencies) (For proprietary rom apks)
2 Optimize images inside	11 Compile System APK files
3 Zip apk	12 Compile Non-System APK Files
4 Sign apk (Dont do this IF its a system apk)	13 Sign apk
5 Zipalign apk (Do once apk is created/signed)	14 Install apk
Install apk (Dont do this IF system apk, do adb push)	15 Compile apk / Sign apk / Install (Non-System Apps Only)
7 Zip / Sign / Install apk (All in one step)	
8 Adb push (Only for system apk)	
----- tools Stuff -----	
17 Batch Optimize Apk (inside place-apk-here-to-batch-optimize only)	
18 Sign an apk(Batch support)(inside place-apk-here-for-signing folder only)	
19 Batch optimize ogg files (inside place-ogg-here only)	
20 Clean Files/Folders	
21 Select compression level for apk's	
22 Set Max Memory Size (Only use IF getting stuck at decompiling/compiling)	
23 Read Log	
24 Set current project	
25 About / Tips / Debug Section	
26 Switch decompile mode (Allows you to pick to fully decompile the APK's or JAR's or to just decompile Sources or just the Resources or do a raw dump allowing you to just edit the normal images)	
00 Quit	

9.3.2. Modificando recursos XML de una aplicación



Ejercicio paso a paso: *Modificar recursos XML de una aplicación.*

```
...
<activity android:theme="@*android:style/Theme.NoTitleBar.Fullscreen"
    android:label="@string/app_name" android:name="LiteSplashActivity_"
    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>
...
```

9.3.3. Modificando el código de una aplicación



Ejercicio paso a paso: *Modificar el código ensamblador de una aplicación Android.*

```
package com.etermax.gamescommon.login.ui;
...

public abstract class BaseSplashActivity extends FragmentActivity
{
    protected static final int LOGIN_REQUEST;
    protected static int SPLASH_DURATION = 2000;
    ...
}
```

```
.class public abstract
Lcom/etermax/gamescommon/login/ui/BaseSplashActivity;
.super Landroid/support/v4/app/FragmentActivity;
.source "BaseSplashActivity.java"

# static fields
.field protected static final LOGIN_REQUEST:I
.field protected static SPLASH_DURATION:I

# direct methods
.method static constructor <clinit>()V
    .locals 1
    .prologue
    .line 21
    const/16 v0, 0x7d0
    sput v0, Lcom/etermax/gamescommon/login/ui/BaseSplashActivity;
                                                SPLASH_DURATION:I
    return-void
.end method

.method public constructor <init>()V
    .locals 0
    .prologue
    .line 19
    invoke-direct {p0}, Landroid/support/v4/app/FragmentActivity;.<init>()V
    return-void
.end method
```



Preguntas de repaso: *Modificando aplicaciones Android.*

Comentario [LM11]: No hay enlace

9.4. Ofuscación del código

```
...  
.field protected static SPLASH_DURATION:I  
  
# direct methods  
.method static constructor <clinit>()V  
  .locals 1  
  .prologue  
  .line 21  
...  
...
```



Ejercicio paso a paso: *Uso de Proguard para ofuscar una aplicación Android.*

```
C:\C9>dex2jar\d2j-dex2jar classes_sin.dex  
dex2jar classes_sin.dex -> classes_sin-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 *;
}

```

```

.class public Lcom/example/android/apis/ApiDemos;
.super Landroid/app/ListActivity;

# static fields
.field private static final a:Ljava/util/Comparator;

# direct methods
.method static constructor <clinit>()V
    .locals 1
    new-instance v0, Lcom/example/android/apis/a;
    invoke-direct {v0}, Lcom/example/android/apis/a;-><init>()V
    sput-object v0, Lcom/example/android/apis/ApiDemos; a:Ljava/util/Comparator;

    return-void
.end method
...

```

```

.class public Lcom/example/android/apis/ApiDemos;
.super Landroid/app/ListActivity;
source "ApiDemos.java"

# static fields
.field private static final sDisplayNameComparator:Ljava/util/Comparator;
annotation system Ldalvik/annotation/Signature;
value = {
    ...
}
end annotation
end field

# direct methods

```

```

.method static constructor <clinit>()V
    .locals 1
    prologue
    line 113
    new-instance v0, Lcom/example/android/apis/ApiDemos$1;
    invoke-direct {v0}, Lcom/example/android/apis/ApiDemos$1;-><init>()V
    line 112
    sput-object v0, Lcom/example/android/apis/ApiDemos; ->
        sDisplayNameComparator: Ljava/util/Comparator;
    line 119
    return-void
.end method
...

```



Preguntas de repaso: *Ofuscación en Android.*

9.5. Obtención de licencias con Google Play

9.5.1. Cómo funciona el servicio de licencias

9.5.2. Como añadir una licencia a nuestra aplicación



Ejercicio paso a paso: *Uso de la librería LVL.*

```

<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();
    }
}

```



Preguntas de repaso: *El servicio de licencias Google Play.*

9.6. Cómo evitar que se elimine la verificación de licencia en nuestras aplicaciones

9.6.1. Ingeniería inversa en una aplicación con licencia



Ejercicio paso a paso: *Ingeniería inversa en una aplicación con licencia.*

```

.class public Long/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
```

```
@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
    iput-boolean v0, p0, Lorg/tomas/girones/jesus/obtencionlicencia/
        MainActivity; ->permitir:Z
    ...
```

9.6.2. Primera contramedida: ofuscar el código



Ejercicio paso a paso: *Ofuscación de la aplicación con licencia.*

```
.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, Long/.../obtenccionlicencia/MainActivity;->d:[B
    return-void
    ...
```



Práctica: Ingeniería inversa en una aplicación ofuscada.



Solución: Algunas posibilidades podrían ser:

```
.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



Ejercicio paso a paso: Modificando la librería LVL.

```
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/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



Ejercicio paso a paso: *Verificar la firma de nuestra aplicación.*

```
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();
    }
}
```



Preguntas de repaso: *Evitar que pirateen nuestra aplicación.*