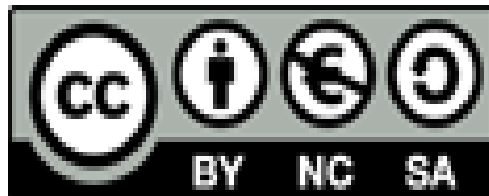


# XML



**[paco@portadaalta.es](mailto:paco@portadaalta.es)**

# Índice

- ✓ ¿Qué es XML?
- ✓ Propiedades de XML
- ✓ Ejemplo de XML
- ✓ Componentes de XML
- ✓ Procesando XML en Android
- ✓ Métodos de XmlPullParser
- ✓ Ejemplo de XmlPullParser
- ✓ Ejercicios

# ¿Qué es XML?

XML significa Extensible Markup Language y fue definido en 1998 por el World Wide Web Consortium (W3C).

XML es un formato muy popular en Internet y es muy usado para compartir datos en Internet. Veremos cómo analizar un documento XML y extraer de él la información necesaria.

Un documento XML consta de elementos, cada uno de ellos tiene una etiqueta de inicio, contenido y una etiqueta de cierre.

# Propiedades de XML

Un documento XML debe estar “bien formado”. Para ello debe cumplir las siguientes condiciones:

- Siempre debe comenzar por un prólogo.
- Cada etiqueta abierta tiene su correspondiente etiqueta de cierre.
- Todas las etiquetas están correctamente anidadas.

Un documento XML se considera válido si está “bien formado” y además contiene un enlace a un esquema XML y es válido de acuerdo a ese esquema.

# Propiedades de XML

Usar XML tiene las siguientes ventajas respecto a usar un formato binario o no estructurado:

- ✓ XML es texto plano
- ✓ XML representa datos sin definir cómo éstos deberían ser mostrados.
- ✓ XML puede ser transformado en otros formatos vía XSL
- ✓ XML puede ser procesado por analizadores estándar
- ✓ Los documentos XML son jerárquicos

# Ejemplo de XML

El siguiente documento XML está “bien formado”:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This is a comment -->
<current>
  <city id="2643743" name="London">
    <coord lon="-0.12574" lat="51.50853"/>
    <country>GB</country>
    <sun rise="2013-10-08 T 06:13:56" set="2013-10-08 T 17:21:45"/>
  </city>
  <temperature value="289.54" min="289.15" max="290.15" unit="kelvin"/>
  <humidity value="77" unit="%"/>
  <pressure value="1025" unit="hPa"/>
</current>
```

# Componentes de XML

Un documento XML tiene varios componentes:

Componente	Descripción
Prolog	Un documento XML comienza con un prólogo, el cual da información sobre el fichero
Events	Los eventos pueden ser: Document starts , Document ends, Tag start , Tag end y Text.
Text	Además de etiquetas y eventos, un documento XML también contiene texto.
Attributes	Son propiedades adicionales de las etiquetas

# Procesando XML en Android

El lenguaje Java facilita varias bibliotecas estándar para procesar ficheros XML: SAX y DOM, las cuales también están disponibles en Android.

Android tiene la interface [XmlPullParser](#) para analizar documentos XML. Este analizador está alojado en [www.xmlpull.org](http://www.xmlpull.org)

En Android es recomendable usar XmlPullParser. Tiene un API relativamente simple comparado con SAX y DOM y es más rápido y requiere menos memoria que el API de DOM.

Usaremos XmlPullParser por su eficiencia y facilidad de uso.



# Métodos de XmlPullParser

Algunos de los métodos de la interface [XmlPullParser](#) son:

Método	Descripción
getEventType()	Devuelve el tipo de evento que ocurre: Document start, ,tag start, text, etc.
next()	Pasa al siguiente evento
getName()	Devuelve el nombre de la etiqueta
getText()	Devuelve el texto contenido en la etiqueta
getAttributeValue(int index)	Devuelve el valor del atributo
getAttributeCount()	Devuelve el número de atributos de la etiqueta actual
getAttributeName(int index)	Devuelve el nombre del atributo especificado por el valor index

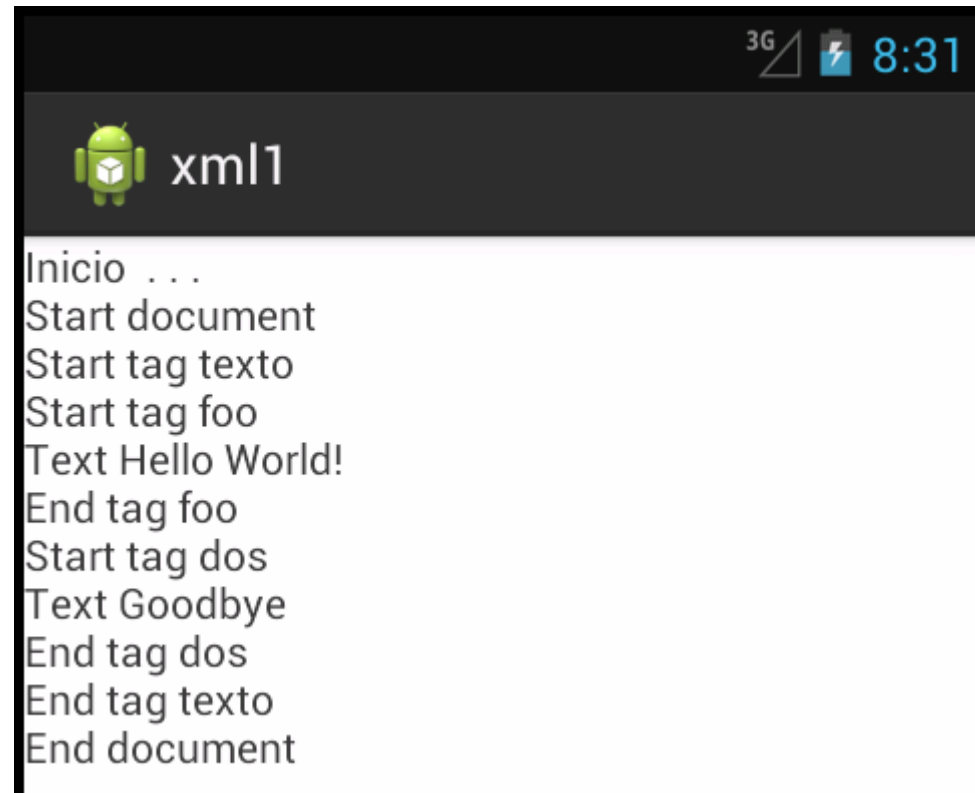
# Ejemplo de XmlPullParser

```
public static void main (String args[])
    throws XmlPullParserException, IOException
{
    XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
    factory.setNamespaceAware(true);
    XmlPullParser xpp = factory.newPullParser();

    xpp.setInput( new StringReader ( "<foo>Hello World!</foo>" ) );
    int eventType = xpp.getEventType();
    while (eventType != XmlPullParser.END_DOCUMENT) {
        if(eventType == XmlPullParser.START_DOCUMENT) {
            System.out.println("Start document");
        } else if(eventType == XmlPullParser.START_TAG) {
            System.out.println("Start tag "+xpp.getName());
        } else if(eventType == XmlPullParser.END_TAG) {
            System.out.println("End tag "+xpp.getName());
        } else if(eventType == XmlPullParser.TEXT) {
            System.out.println("Text "+xpp.getText());
        }
        eventType = xpp.next();
    }
    System.out.println("End document");
}
```

# Ejercicio 1: Partes de un documento XML

A partir de la documentación de [XmlPullParser](#), crear una aplicación que muestre en pantalla las diferentes partes de un documento XML:



# Ejercicio 1: Partes de un documento XML

```
public static final String TEXTO = "<texto><uno>Hello World!  
</uno><dos>Goodbye</dos></texto>";
```

```
public static String analizar(String texto) throws XmlPullParserException, IOException  
{  
    StringBuilder cadena = new StringBuilder();  
    XmlPullParser xpp = Xml.newPullParser();  
  
    xpp.setInput( new StringReader(texto));  
    int eventType = xpp.getEventType();  
    cadena.append("Inicio . . . \n");  
    while (eventType != XmlPullParser. END_DOCUMENT ) {  
  
  
  
  
  
  
  
  
  
    }  
    //System.out.println("End document");  
    cadena.append("End document" + "\n" + "Fin");  
  
    return cadena.toString();  
}
```

## Ejercicio 2: Nombres y notas

Crear una aplicación que lea un fichero en formato xml con los datos y notas de alumnos.

El fichero *alumnos.xml* estará localizado en la carpeta */res/xml*.

Se mostrarán en pantalla los nombres, las asignaturas, las notas, las fechas y la nota media de todos los alumnos.

# Ejercicio 2: Nombres y notas

Fichero *alumnos.xml*:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<alumnos>
  <alumno>
    <nombre>Carmelo Cotón</nombre>
    <nota asignatura="matemáticas" fecha="15/11/2012">7</nota>
    <observaciones>bien</observaciones>
  </alumno>
  <alumno>
    <nombre>Aitor Tilla</nombre>
    <nota asignatura="lengua" fecha="20/11/2012">3.4</nota>
    <observaciones>regular</observaciones>
  </alumno>
  <alumno>
    <nombre>Mercedes Tartalada</nombre>
    <nota asignatura="inglés" fecha="24/11/2012">9</nota>
    <observaciones>muy bien</observaciones>
  </alumno>
</alumnos>
```

## Ejercicio 2: Nombres y notas



## Ejercicio 2: Nombres y notas

```
public static String analizarNombres(Context c) throws XmlPullParserException,
IOException {
    boolean esNombre = false;
    boolean esNota = false;
    StringBuilder cadena = new StringBuilder();
    Double suma = 0.0;
    int contador = 0;

    XmlResourceParser xrp = c.getResources().getXml(R.xml.alumnos);
    int eventType = xrp.getEventType();
    while (eventType != XmlPullParser.END_DOCUMENT) {
        switch (eventType) {
            case XmlPullParser.START_TAG :

                break;
            case XmlPullParser.TEXT :

                break;
            case XmlPullParser.END_TAG :

                break;
        }
        eventType = xrp.next();
    }

    return cadena.toString();
}
```



# Ejercicio 3: RSS

Crear una aplicación que muestre los títulos de las últimas entradas en el [blog de Alejandro Suárez](#).

La URL del Feed es <http://www.alejandro Suarez.es/feed/>

Ayuda:

[RSS](#)

[¿Qué es RSS?](#)



# Ejercicio 3: RSS

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0"
  ... >
<channel>
  <title>Alejandro Suárez Sánchez-Ocaña</title>
  <atom:link href="http://www.alejandro-suarez.es/feed/" rel="self" type="application/rss+xml" />
  <link>http://www.alejandro-suarez.es</link>
  ...
  <item>
    <title>Emprender a los 50</title>
    <link>http://www.alejandro-suarez.es/2014/11/emprender-a-los-50-2/</link>
    ...
  </item>
  <item>
    <title>Morir con las botas puestas</title>
    <link>http://www.alejandro-suarez.es/2014/11/morir-con-las-botas-puestas/</link>
    ...
  </item>
  <item>
    ...
  </item>
</channel>
</rss>
```

# Ejercicio 3: RSS

Visor xml:



JSONFormatter.org | News | My Ip | Recent Links | Sample | Save | More ▾ | Sign in | (?)

## XML VIEWER

Save & Share

XML Input

sample ✕ ↺

```
1 <?xml version="1.0" encoding="UTF-8"?><rss version="2.0"
2   xmlns:content="http://purl.org/rss/1.0/modules/content/"
3   xmlns:wfw="http://wellformedweb.org/CommentAPI/"
4   xmlns:dc="http://purl.org/dc/elements/1.1/"
5   xmlns:atom="http://www.w3.org/2005/Atom"
6   xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
7   xmlns:slash="http://purl.org/rss/1.0/modules/slash/"
8   >
9
10 <channel>
11   <title>Alejandro Suárez Sánchez-Ocaña</title>
12   <atom:link href="http://www.alejandrosuarez.es/feed/" rel="self" type
13   ="application/rss+xml" />
14   <link>http://www.alejandrosuarez.es</link>
15   <description></description>
16   <lastBuildDate>Mon, 23 Nov 2015 03:52:08 +0000</lastBuildDate>
17   <language>es-ES</language>
18   <sy:updatePeriod>hourly</sy:updatePeriod>
19   <sy:updateFrequency>1</sy:updateFrequency>
20   <generator>http://wordpress.org/?v=4.3.1</generator>
21   <item>
22     <title>SER Valencia</title>
23     <link>http://www.alejandrosuarez.es/2015/11/ser-valencia/</link>
24     <comments>http://www.alejandrosuarez.es/2015/11/ser-valencia
25     /#comments</comments>
26     <pubDate>Thu, 19 Nov 2015 18:33:42 +0000</pubDate>
27     <dc:creator><![CDATA[diego]]></dc:creator>
28     <category><![CDATA[General]]></category>
29     <guid isPermaLink="false">http://www.alejandrosuarez.es/?p=3792
30     <description><![CDATA[It 72-10 11s for sale will 72-10 11s help
31     promote nail 72-10 11s and will 72-10 11s motivate some nail 72-10 11s
32     expansion. Make pre order 72-10 jordans use of a 72-10 11s for sale top
33     jacket http://integraleon.com/pre-order-aqua-8s.html whenever aqua 8s you
34     shine aqua 8s your nails auqa 8s for sale to avoid chrome 8s [&#8230;]]>
35     </description>
```

Load Url

Browse

Tree View

Beautify / Format

Minify

XML To JSON

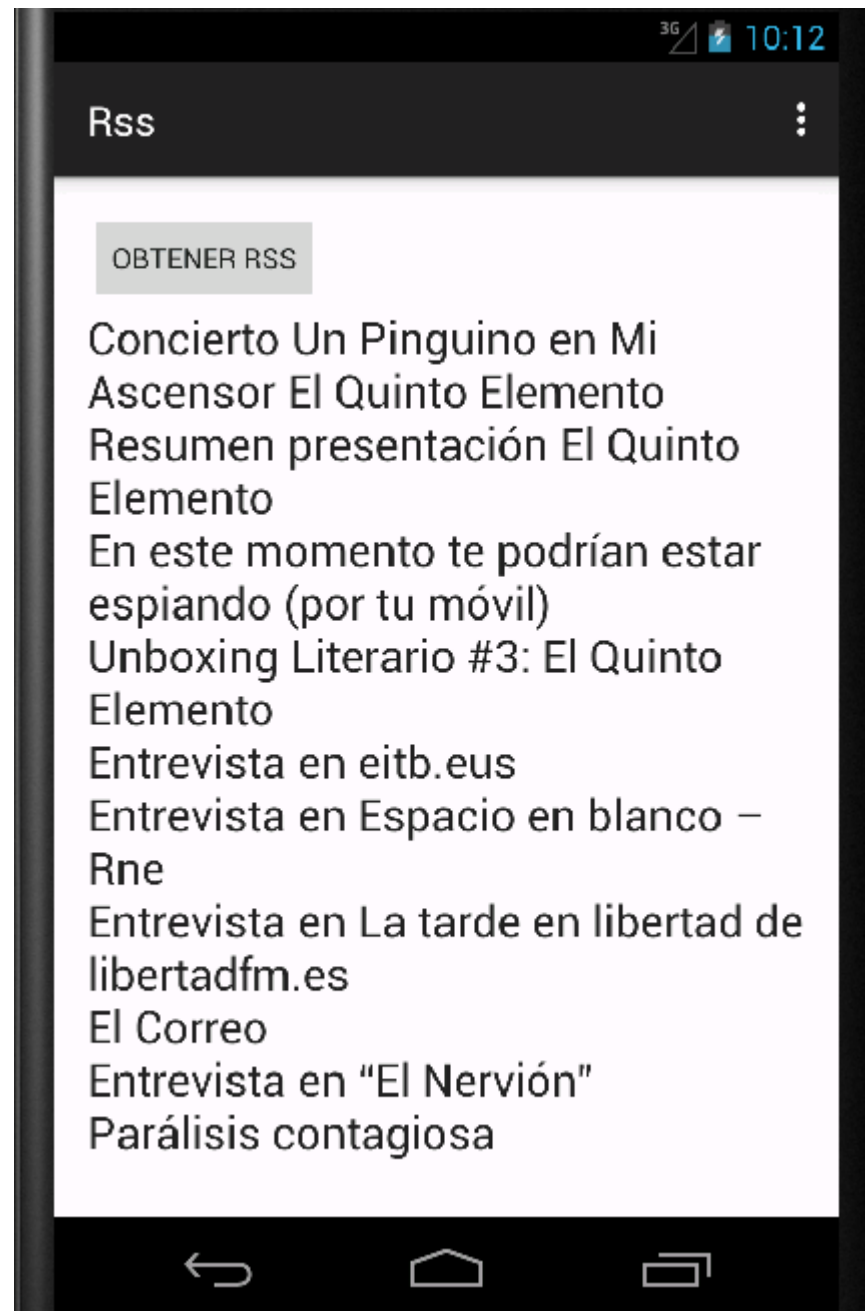
Export To CSV

Download

Result : XML Tree View

```
rss ..
  @xmlns:content: http://purl.org/rss/1.0/modules/content/
  @xmlns:wfw: http://wellformedweb.org/CommentAPI/
  @xmlns:dc: http://purl.org/dc/elements/1.1/
  @xmlns:atom: http://www.w3.org/2005/Atom
  @xmlns:sy: http://purl.org/rss/1.0/modules/syndication/
  @xmlns:slash: http://purl.org/rss/1.0/modules/slash/
  @version: 2.0
  channel ..
    title Alejandro Suárez Sánchez-Ocaña
    atom:link
      @href: http://www.alejandrosuarez.es/feed/
      @rel: self
      @type: application/rss+xml
    link http://www.alejandrosuarez.es
    description
    lastbuilddate Mon, 23 Nov 2015 03:52:08 +0000
```

# Ejercicio 3: RSS



# Ejercicio 3: RSS

## Dependencias en Gradle:

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    compile 'com.android.support:appcompat-v7:23.0.1'  
    compile 'cz.msebera.android:httpclient:4.4.1.1'  
    compile 'com.loopj.android:android-async-http:1.4.9'  
}
```

## Permisos en el manifiesto:

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

# Ejercicio 3: RSS

```
public static String analizarRSS(File file) throws NullPointerException, XmlPullParserException,
IOException {
    boolean dentroItem = false;
    boolean dentroTitle = false;

    StringBuilder builder = new StringBuilder();
    XmlPullParser xpp = Xml.newPullParser();

    xpp.setInput(new FileReader(file));
    int eventType = xpp.getEventType();
    while (eventType != XmlPullParser.END_DOCUMENT) {
        switch (eventType) {
            case XmlPullParser.START_TAG:

                break;
            case XmlPullParser.TEXT:

                break;
            case XmlPullParser.END_TAG:

                break;
        }
        eventType = xpp.next();
    }
    return builder.toString();
}
```

# Ejercicio 3: RSS

```
public class TitularesRss extends AppCompatActivity implements OnClickListener {  
    public static final String RSS = "http://www.alejandro Suarez.es/feed/";  
    //public static final String RSS = "http://10.0.2.2/feed/alejandro.xml";  
    public static final String TEMPORAL = "alejandro.xml";  
    Button boton;  
    TextView informacion;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_rss);  
  
        boton = (Button) findViewById(R.id.button);  
        boton.setOnClickListener(this);  
        informacion = (TextView) findViewById(R.id.informacion);  
    }  
  
    @Override  
    public void onClick(View v) {  
        if (v == boton)  
            descarga(RSS, TEMPORAL);  
    }  
}
```

# Ejercicio 3: RSS

```
private void descarga(String rss, String temporal) {  
    final ProgressDialog progreso = new ProgressDialog(this);  
  
    File miFichero = new File(Environment.getExternalStorageDirectory().getAbsolutePath(), temporal);  
    RestClient.get(rss, new FileAsyncHttpResponseHandler(miFichero) {  
        @Override  
        public void onStart() {  
            super.onStart();  
            progreso.setProgressStyle(ProgressDialog.STYLE_SPINNER);  
            progreso.setMessage("Conectando . . .");  
            progreso.setCancelable(false);  
            progreso.show();  
        }  
    });  
}
```



# Ejercicio 4: Lista de noticias

Crear una aplicación que muestre los titulares de las noticias para Andalucía en [Europa press](#).

## Contenidos RSS

Europa Press ofrece la posibilidad de acceder a los contenidos de su página web en diversos formatos (como por ejemplo **RSS** o inclusión en tu Web) para una mayor facilidad y flexibilidad de acceso.

### RSS

---

El formato **RSS** es un estándar de intercambio de artículos, que permite la integración y actualización de contenidos de forma automática en diversas plataformas, como por ejemplo programas lectores en su ordenador, páginas web, etc.

Se analizará el feed rss y se desplegará en pantalla una lista, la cual mostrará los titulares del feed rss.

Al pulsar en un titular de la lista, se abrirá el navegador para ver el artículo completo correspondiente a ese titular.

## Ejercicio 4: Lista de noticias



xml5

Sierra Nevada afronta el puente con 45 kilómetros esquiables



Errejón no se ajustó al contrato y no pidió compatibilidad para actividad privada

El Betis arregla el desastre y el Getafe zarandea al Eibar

Alertan de que "han desaparecido siete bebés" tras el rescate de la patera

Desmantelan en Níjar un laboratorio de cocaína similar a los de la selva colombiana



3G 8:52



m.europapress.es/a



europapress



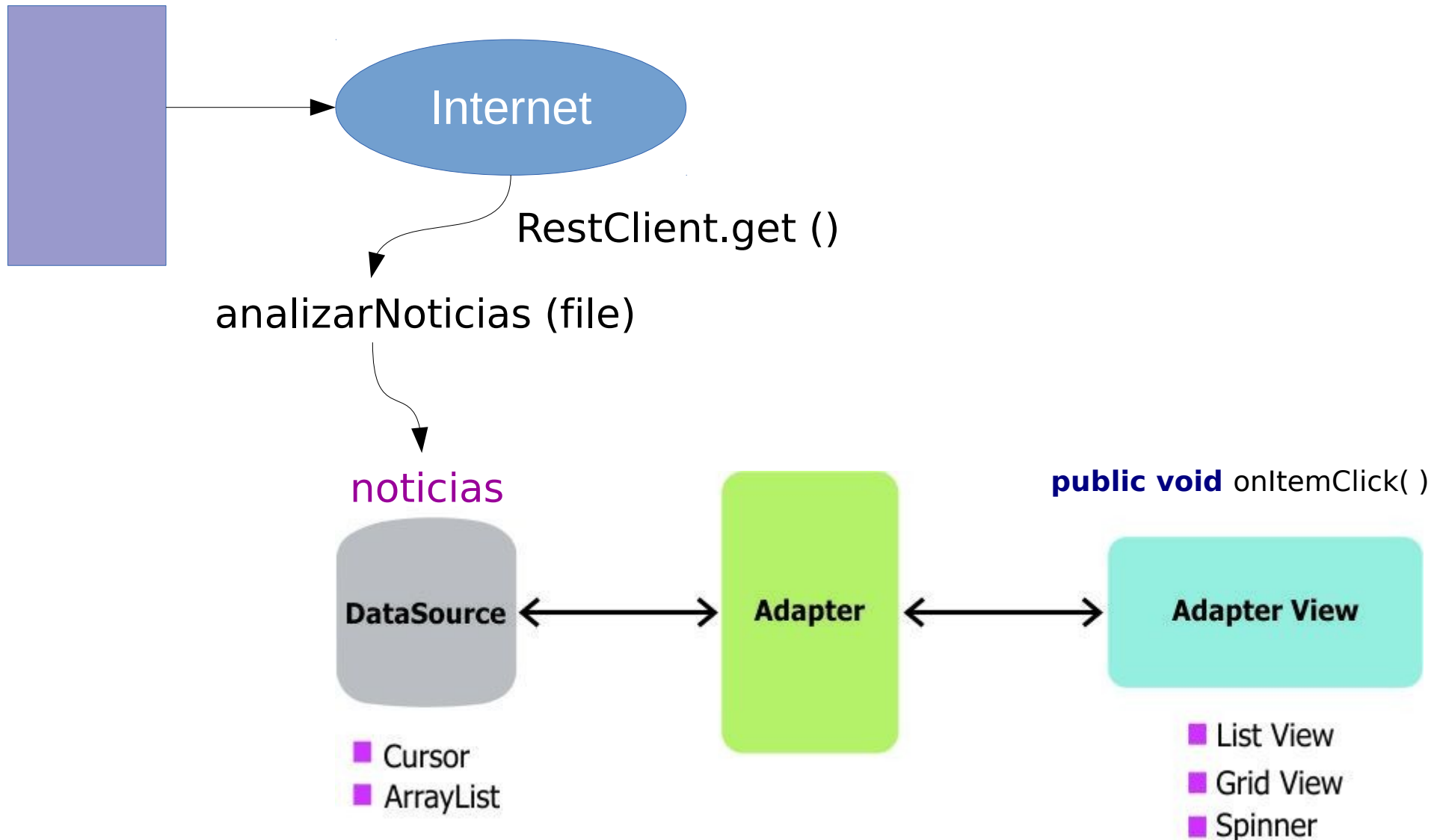
14

compartidos



# Ejercicio 4: Lista de noticias

[www.EuropaPress.es/rss](http://www.EuropaPress.es/rss)



# Ejercicio 4: Lista de noticias

Botón flotante:

Floating Action Buttons

Material design icons site

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    android:id="@+id/main_content"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"></ListView>

    <android.support.design.widget.FloatingActionButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|right"
        android:layout_margin="26dp"
        android:src="@drawable/autorenew"
        app:layout_anchor="@id/listView"
        app:layout_anchorGravity="bottom|right|end"
        android:id="@+id/floatingActionButton"
        app:backgroundTint="@android:color/holo_green_dark" />

</android.support.design.widget.CoordinatorLayout>
```

# Ejercicio 4: Lista de noticias

## Clase Noticia

```
public class Noticia {  
    private String title;  
    private String link;  
    private String description;  
    private String pubDate;  
  
    public String getTitle() {return title;}  
    public void setTitle(String title) {  
        this.title = title;  
    }  
    public String getLink() {return link;}  
    public void setLink(String link) {  
        this.link = link;  
    }  
    public String getDescription() {return description;}  
    public void setDescription(String description) {  
        this.description = description;  
    }  
    public String getPubDate() {return pubDate;}  
    public void setPubDate(String pubDate) {  
        this.pubDate = pubDate;  
    }  
  
    @Override  
    public String toString() {  
        return title;  
    }  
}
```

# Ejercicio 4: Lista de noticias

## Analizar Noticias:

```
public static ArrayList<Noticia> analizarNoticias(File file) throws XmlPullParserException, IOException {  
    int eventType;  
    ArrayList<Noticia> noticias = null;  
    Noticia actual = null;  
    boolean dentroItem = false;  
  
    XmlPullParser xpp = Xml.newPullParser();  
    xpp.setInput(new FileReader(file));  
    eventType=xpp.getEventType();  
    while (eventType!=XmlPullParser.END_DOCUMENT){  
        switch (eventType) {  
            case XmlPullParser.START_DOCUMENT:  
  
                break;  
            case XmlPullParser.START_TAG:  
  
                break;  
            case XmlPullParser.END_TAG:  
  
                break;  
        }  
        eventType = xpp.next();  
    }  
    //devolver el array de noticias  
    return noticias;  
}
```

# Ejercicio 4: Lista de noticias

## Actividad Noticias:

```
public class Noticias extends Activity implements OnClickListener, OnItemClickListener {  
    public static final String CANAL = "http://www.europapress.es/rss/rss.aspx?ch=279";  
    //public static final String CANAL = "http://192.168.1.200/feed/europapress.xml";  
    public static final String TEMPORAL = "europapress.xml";  
    ListView lista;  
    ArrayList<Noticia> noticias;  
    ArrayAdapter<Noticia> adapter;  
    FloatingActionButton fab;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_noticias);  
  
        lista = (ListView) findViewById(R.id.listView);  
        lista.setOnItemClickListener(this);  
        fab = (FloatingActionButton) findViewById(R.id.floatingActionButton);  
        fab.setOnClickListener(this);  
    }  
  
    @Override  
    public void onClick(View v) {  
        if (v == fab)  
            descarga(CANAL, TEMPORAL);  
    }  
}
```

# Ejercicio 4: Lista de noticias

```
private void descarga(String canal, String temporal) {
```

```
}
```

```
private void mostrar() {
```

```
    if (noticias != null)
```

```
        if (adapter == null) {
```

```
            else {
```

```
            }
```

```
        else
```

```
            Toast.makeText(getApplicationContext(), "Error al crear la lista", Toast.LENGTH_SHORT).show();
```

```
    }
```

```
@Override
```

```
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
```

```
    Uri uri = Uri.parse((String) noticias.get(position).getLink());
```

```
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
```

```
    if (intent.resolveActivity(getPackageManager()) != null)
```

```
        startActivity(intent);
```

```
    else
```

```
        Toast.makeText(getApplicationContext(), "No hay un navegador", Toast.LENGTH_SHORT).show();
```

```
}
```



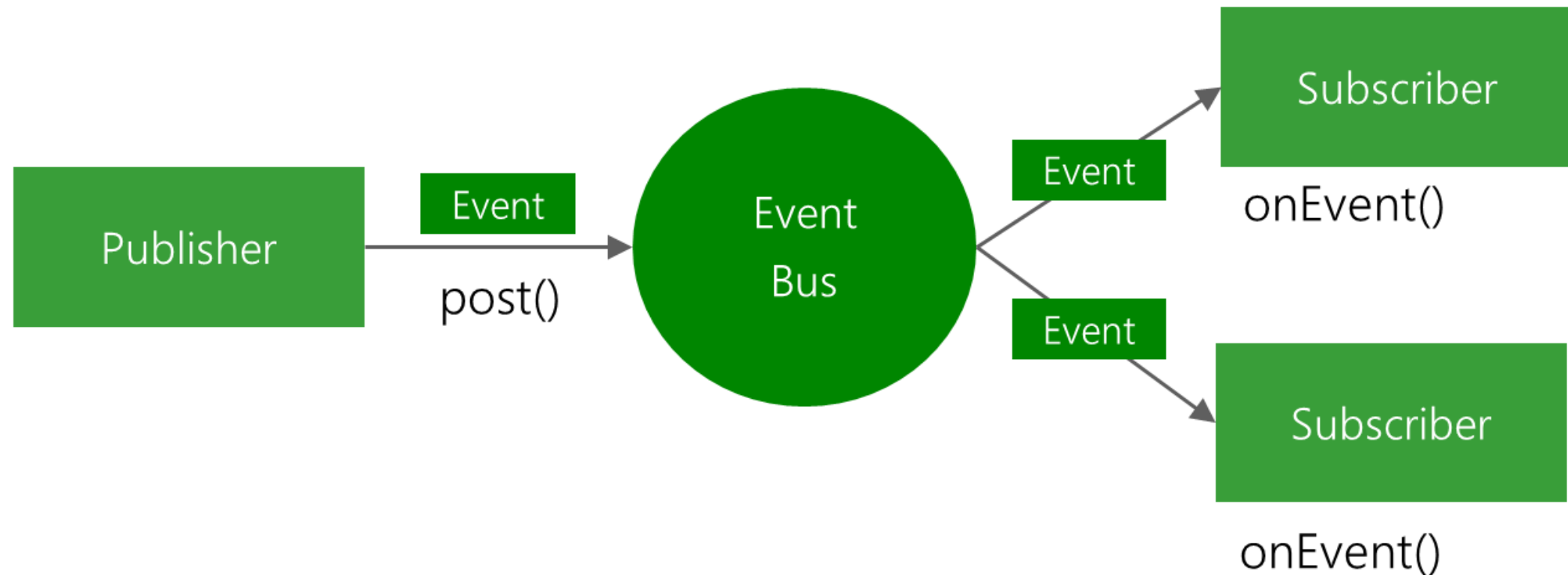
# Ejercicio 4: Lista de noticias

Mejora: Utilizar un **RecyclerView**



# Ejercicio 4: Lista de noticias

Mejora: Utilizar EventBus



**How to get started with EventBus in 3 steps**

# Ejercicio 5: Escribir XML

Crear un documento XML (en un fichero en memoria externa) que contenga los titulares, enlaces, descripciones y fechas de los artículos obtenidos del [RSS](#) del blog de Alejandro Suárez.

## **Alejandro Suárez Sánchez-Ocaña**

### **[Emprender a los 50](#)**

03/11/14 21:30

Emprender a los 50

### **[Morir con las botas puestas](#)**

03/11/14 21:29

Morir con las botas puestas

### **[Internet: el día más infame](#)**

23/09/14 20:23

Internet: el día más infame

### **[Tu gurú, tururú...](#)**

23/09/14 20:20

Tu gurú, tururú ...

# Ejercicio 5: Escribir XML

## Interface `XmlSerializer`:

<code>setOutput(OutputStream os, String encoding)</code>	Set to use binary output stream with given encoding.
<code>startDocument(String encoding, Boolean standalone)</code>	Write <code>&lt;?xml</code> declaration with encoding (if encoding not null) and standalone flag (if standalone not null)
<code>setFeature(String name, boolean state)</code>	Set feature identified by name
<code>startTag(String namespace, String name)</code>	Writes a start tag with the given namespace and name.
<code>attribute(String namespace, String name, String value)</code>	Write an attribute.
<code>text(String text)</code>	Writes text, where special XML chars are escaped automatically
<code>endTag(String namespace, String name)</code>	Write end tag.
<code>endDocument()</code>	Finish writing.

# Ejercicio 5: Escribir XML

Fichero *titulares.xml*:

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<titulares>
  <item>
    <titulo fecha="Mon, 03 Nov 2014 20:30:33 +0000">Emprender a los 50</titulo>
    <enlace>http://www.alejandro-suarez.es/2014/11/emprender-a-los-50-2/</enlace>
    <descripcion>Emprender a los 50</descripcion>
  </item>
  <item>
    <titulo fecha="Mon, 03 Nov 2014 20:29:01 +0000">Morir con las botas puestas</titulo>
    <enlace>http://www.alejandro-suarez.es/2014/11/morir-con-las-botas-puestas/</enlace>
    <descripcion>Morir con las botas puestas</descripcion>
  </item>
  <item>
    <titulo fecha="Tue, 23 Sep 2014 18:23:40 +0000">Internet: el día más infame</titulo>
    <enlace>http://www.alejandro-suarez.es/2014/09/internet-el-dia-mas-infame/</enlace>
    <descripcion>Internet: el día más infame</descripcion>
  </item>
  . . .

</titulares>
```

# Ejercicio 5: Escribir XML

```
public static void crearXML(ArrayList<Noticia> noticias, String fichero) throws IOException {  
    FileOutputStream fout;  
  
    fout = new FileOutputStream(new File(Environment.getExternalStorageDirectory().getAbsolutePath(), fichero));  
    XmlSerializer serializer = Xml.newSerializer();  
    serializer.setOutput(fout, "UTF-8");  
    serializer.startDocument(null, true);  
    serializer.setFeature("http://xmlpull.org/v1/doc/features.html#indent-output", true); //poner tabulación  
  
    serializer.startTag(null, "titulares");  
    for (int i = 0; i < noticias.size(); i++) {  
  
    }  
    serializer.endTag(null, "titulares");  
    serializer.endDocument();  
    serializer.flush();  
    fout.close();  
}
```

# Ejercicio 5: Escribir XML

```
public class CreacionXML extends AppCompatActivity implements OnClickListener {
    public static final String RSS = "http://www.alejandro Suarez.es/feed/";
    //public static final String RSS = "http://10.0.2.2/feed/alejandro.xml";
    public static final String TEMPORAL = "alejandro.xml";
    public static final String FICHERO_XML = "resultado.xml";
    Button boton;
    ArrayList<Noticia> noticias;

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

        boton = (Button) findViewById(R.id.boton);
        boton.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        if (v == boton)
            descarga(RSS, TEMPORAL);
    }

    private void descarga(String rss, String temporal) {

    }
}
```

¿Dudas?

¿Sugerencias?



**paco@portadaalta.es**