

Francisco Miguel Pardo Vesés



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA

---

UNIVERSIDAD POLITÉCNICA DE VALENCIA  
ESCUELA TÉCNICA SUPERIOR DE  
INFORMATICA APLICADA

*Aplicación que permita emitir contenidos de la Web-TV de la UPV para dispositivos móviles tipo iPhone, windows o Android*

**AUTOR**

Francisco Miguel Pardo Vesés

**DIRECTOR**

Ramón Mollá Vayá

Valencia, a 11 de julio de 2011

## **Resumen**

El proyecto desarrolla la creación de un aplicación para smartphones o tablets bajo sistema operativo Android que permite la búsqueda de contenidos de radio y de televisión alojados en la web-TV de la Universitat Politècnica de València.

Basándose en un estudio de la web-TV de la UPV, se realizó un interfaz simplificado para dispositivos móviles. El proyecto comienza con una especificación de requisitos adaptados a las capacidades gráficas y de comunicación de los dispositivos móviles. A continuación se realiza un análisis de los casos de uso más habituales. Seguidamente se realiza un diseño en tres capas: Presentación, Negocio y Datos y se analiza la forma en la que se comunicará la aplicación con el gestor de bases de datos de la web-TV. También se realiza una descripción de la estructura del proyecto y de las tecnologías empleadas. El proyecto se acompaña de algunos anexos explicativos complementarios.

## **Palabras claves**

Android, UPV, RTV, streaming, Radio, Televisión, smartphone, web-TV, tablet.

## Índice de contenido

1.Introducción.....	5
1.1.Motivación y justificación del proyecto.....	5
1.3.Visibilidad de Android Market.....	6
2. Análisis del problema.....	7
Inicio.....	8
Tv a la carta.....	9
Programación.....	9
Programas.....	10
TV en directo.....	13
Radio.....	14
Actualidad.....	14
Descargas.....	15
3.Especificación de requisitos.....	17
3.1.Introducción.....	17
3.1.1.Propósito.....	17
3.1.2.Ámbito.....	17
3.1.3.Definiciones, acrónimos y abreviaturas.....	17
3.1.4 Referencias.....	18
3.1.5.Visión global.....	18
3.2.Descripción general.....	18
3.2.1.Perspectiva del producto.....	18
3.2.2.Funciones del producto.....	19
3.2.3.Características del usuario.....	19
3.2.4.Restricciones generales.....	19
3.2.4.Supuestos y dependencias.....	19
3.3.Requerimientos específicos.....	20
3.3.1.Requisitos de Interfaces Externas.....	20
3.3.2.Requerimientos de eficiencia.....	20
3.3.3.Atributo.....	20
3.3.4.Requisitos Funcionales.....	21
4.0.Análisis.....	22
4.1.Diagrama de Casos de Uso.....	22
5.0.Diseño.....	24
5.1.Modelo en tres capas.....	24
5.1.1.Capa de presentación.....	24
5.1.2.Capa de Negocio.....	31
5.1.3.Capa de Datos.....	34
6.Sistema de información.....	34
6.1.Comunicación en la aplicación e infraestructura.....	34
7.Estructura del proyecto.....	38
7.1.Pantallas.....	38
7.2.Objetos.....	41
Programa.....	41
Capítulo.....	41
Programa_Parrilla.....	41
7.3.Constantes.....	42

ConstantesAplicación.....	42
MensajesError.....	42
7.4.Parseadores.....	43
Parseador_XML_Parrilla.....	43
Parseador_XML_Programas.....	44
Parseador_XML_Programas_Capitulos.....	44
7.5.Utilidades.....	45
8.Tecnologías usadas.....	46
8.1.SDK de Android.....	46
8.2.JAVA.....	50
8.3.JRE.....	51
8.4.XML.....	51
Ventajas del XML.....	52
8.5.SAX.....	52
8.6.PL-SQL.....	53
9.Herramientas utilizadas.....	55
9.1.Samsung Galaxy Tab.....	55
9.2.Entorno de desarrollo Eclipse Helios.....	56
10.Conclusiones y agradecimientos.....	57
Apéndice A:.....	58
Manual de instalación del entorno de desarrollo.....	58
Instalación de las Android Development Tools (ADT).....	59
Apéndice B:.....	62
Manual de usuario de la aplicación.....	62
Apéndice C:.....	65
Subida aplicación a Android Market.....	65
Assets.....	65
Listing details.....	66
Publishing options.....	66

# 1.Introducción

## 1.1.Motivación y justificación del proyecto

La motivación de este proyecto mas importante es la de utilizar las plataforma móvil como son los Smartphones para difundir el contenido de la RTTV de la Universidad Politécnica de Valencia y hacer que llegue a cualquier personas en cualquier parte del mundo sin tener que disponer de un ordenador para poder visualizar su contenido. Esto es posible gracias a la gran aceptación que ha tenido en la sociedad actual desde hace unos años el uso de estos dispositivos móviles y sobretodo a que las compañías que se encargan de los S.O de los distintos modelos de Smartphone (en nuestro caso Google y su Android) ponen a disposición de todo el que quiera su SDK(Software Development Kit), con el que poder realizar aplicaciones propias y aprovechar todo el potencial que se esconde detrás de estos dispositivos y de su S.O.

En cuanto a motivaciones mas personales, cabe destacar la fascinación que me produjo en su momento cuando empecé a trabajar con Android y su SDK, al ver la gran cantidad de posibilidades y la cuantiosa información que se dispone para poder aprender y desarrollar tus propias aplicaciones. Además siempre me ha interesado mucho el tema de aplicaciones móviles y su integración en el mundo laboral, ya no solo en el apartado de entretenimiento sino en su uso laboral día a día.

Este proyecto nace de la necesidad por parte de la RTVV de la UPV de disponer de una aplicación móvil en la que sus usuarios puedan ir al contenido directamente y ver lo que necesitan en ese momento de una forma rápida y muy intuitiva. Para ello primero se indago y se estudio que plataformas dominan el mercado actual de Smartphones. El mercado se rige por cuatro compañías y sus correspondientes S.O.

- Apple(iOS)
- BlackBerry(RIM BlackBerry OS)
- Google(Android)
- Microsoft(Windows Mobile)

Y entre estos cuatro se decidió optar por Android, ya que actualmente muchas compañías apuestan por el y tiene muy buena cuota de mercado, por lo que la difusión sera muy buena.

Una vez se decidió la plataforma sobre la cual funcionaria nuestra aplicación, se necesita saber si la solución seria viable, por lo cual investigamos y probamos la funcionalidad de aplicaciones similares y se llego a la conclusión que seria una solución viable y que le abriría las puertas a muchos usuarios que posiblemente nunca hubiera entrado en la pagina web de la RTVV pero que gracias a esta aplicación podrían convertirse en usuarios asiduos de los contenidos y poco a poco ir ganando adeptos a los programas tanto de radio como de televisión.

También se eligió por su sistema de distribución (Android Market), en un principio todas las compañías tienen un portal donde poder descargar las aplicaciones tanto propias como de terceros, pero nos pareció mas interesante Android Market.

### **1.3.Visibilidad de Android Market**

En cuanto a la posibilidad de difusión de nuestra aplicación, cabe destacar que Android dispone de un portal virtual denominada Android Market. Este portal virtual dispone al servicio del usuario una gran cantidad de aplicaciones de diversas índoles y debido a que Android dispone como se ha comentado anteriormente de un SDK libre para todo aquel que quiera utilizarlo, las aplicaciones crecen en numero día a día.

Android Market tiene un acceso fácil y rápido a sus aplicaciones. Las aplicaciones son creadas por programadores de todo el mundo y posteriormente puntuadas por los usuarios de Android. El menú tiene las siguientes opciones:

- Mostrado: avanza por los iconos de la parte superior para ver las aplicaciones mostradas.
- Aplicaciones: examina todas las aplicaciones o busca aplicaciones por categorías.
- Juegos: examina todos los juegos o busca juegos por categorías.
- Búsqueda: busca aplicaciones en Android Market.
- Mis descargas: visualiza las aplicaciones que están instaladas en el dispositivo.

Las categorías de juegos y aplicaciones, del menú principal proporcionan submenús para que la búsqueda sea más sencilla. Los usuarios tienen la posibilidad de valorar las aplicaciones mediante un sistema similar a YouTube, con una escala del 1 al 5, también ofrece la posibilidad de poner comentarios sobre la aplicación. La novedad de la nueva versión de Android Market es también la posibilidad de añadir capturas de pantalla de su aplicación.

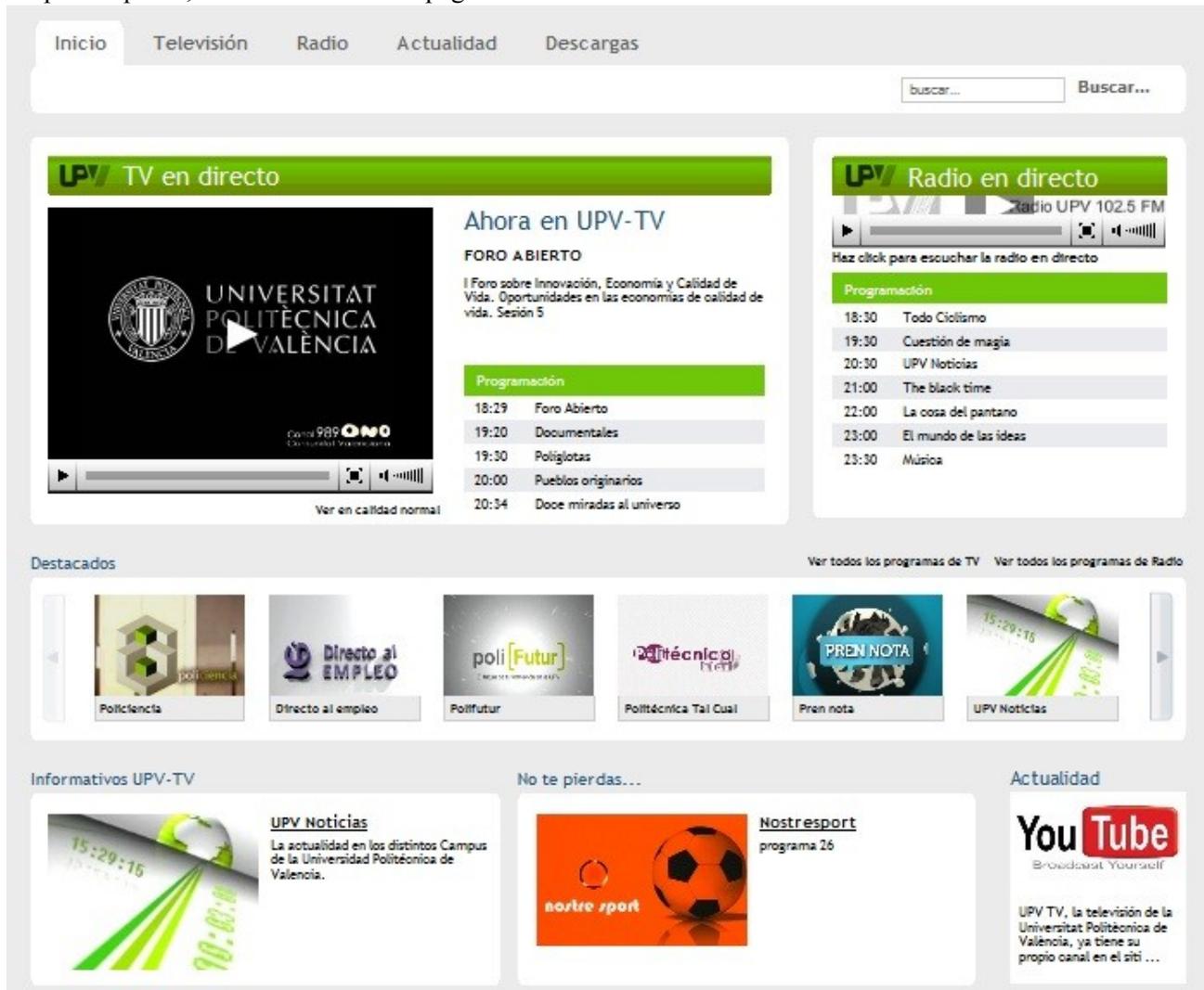
Las aplicaciones de las que dispone son tanto de pago como gratuitas, eso lo decide la persona o empresa que ha desarrollado dicha aplicación, aunque las aplicaciones gratuitas son un gran numero de ellas.

Este sistema de difusión juega un buen papel en la difusión y socialización de nuestra aplicación, lo que conllevara a que gracias a este portal que en poco tiempo los usuarios de esta aplicación crezcan significativamente y en conseguir una de las metas propuestas.

## 2. Análisis del problema

En este apartado, trataremos 3 puntos fundamentales para poder realizar y comprender el porque de este proyecto y su punto de partida.

En primer punto, hablaremos sobre la pagina web de la TV de la UPV



Como se puede observar la estructura de la pagina web es clara y se divide en 5 partes o pestañas, que son :

- Inicio
- Televisión
- Radio
- Actualidad
- Descargas

Ahora detallaremos cada una de las partes y que contienen, ya que la estructura de la aplicación que se ha desarrollado sigue un patrón muy similar, pero con ciertas diferencias.

## Inicio

En este apartado de la pagina web, disponemos de la posibilidad de ver lo que se esta emitiendo en directo en la televisión de la Politécnica, a la derecha la programación que podremos ver y en que horario sera retransmitida.

A la misma altura, también tenemos la posibilidad de escuchar la radio de la Politécnica en directo y también con su horario debajo en donde podremos ver que programa se esta emitiendo.

Y debajo de estos dos apartados de la pestaña Inicio, tenemos una selección de los programas destacado de la pagina que podemos visitar.

## Televisión

The screenshot shows the 'Televisión' section of a website. At the top, there is a navigation menu with 'Inicio', 'Televisión', 'Radio', 'Actualidad', and 'Descargas'. Below this is a search bar with the text 'buscar...' and a 'Buscar...' button. The main content area is titled 'Televisión a la carta' and features a grid of program cards. Each card includes a thumbnail image, a title, a brief description, and a 'Más información' link. The programs displayed are: 'Campus Link' (Espacio de entrevistas que 'linka' la Politécnica con la realidad exterior), 'Politecnos' (Politecnos es una coproducción de la Unidad de Comunicación Científica e Innovación de la Universidad Po...), 'Politécnica Tal Cual' (Programa de la actualidad en los campus de la UPV), 'Caminos en el Aire' (Programa de divulgación en materia aeronáutica), 'Notre Sport' (Magazine de actualidad del deporte valenciano y de la UPV), 'Poliglotos' (Poliglotos es el espacio destinado a compartir con el espectador todo lo que tenga que ver con la Actividad...), 'Directo al Empleo' (Directo al empleo es la cita del Servicio Integrado de Empleo en UPV Televisión), 'Pren Nota' (Pren Nota repasa semanalmente las actividades culturales, que se realizan a lo largo de la semana), and 'Polifutur' (Polifutur, el programa de Formación Permanente de la Universidad Politécnica de Valencia).

En la pestaña de Televisión, podemos encontrar a su vez 4 submenús que son los siguientes:

- TV a la carta
- Programación
- Programas
- TV en directo

## Tv a la carta

Televisión a la carta

2/3

 <p><b>Acto Institucional UPV</b> Actos institucionales de la UPV <a href="#">Más información</a></p>	 <p><b>Foro Abierto</b> Programa que recopila y difunde las conferencias, jornadas y seminarios que han tenido lugar en los días ... <a href="#">Más información</a></p>	 <p><b>Mundo de Ideas</b> Mundo de Ideas del Instituto IDEAS UPV es un programa semanal sobre investigación, innovación, desarrollo ... <a href="#">Más información</a></p>
 <p><b>Telegrafies</b> Telegrafies <a href="#">Más información</a></p>	 <p><b>Más allá de la noticia</b> "Más allá de la noticia" es un programa producido por la Asociación de Antiguos Alumnos de la Universidad ... <a href="#">Más información</a></p>	 <p><b>Tránsito</b> Tránsito es un programa de divulgación científica de la televisión de la Universidad Politécnica de Vale ... <a href="#">Más información</a></p>
 <p><b>Antena Paranoica</b> Antena Paranoica es un programa gamberro con toques de humor, que da pie a películas bizarras ... <a href="#">Más información</a></p>	 <p><b>Entre Jardines</b> Entre Jardines <a href="#">Más información</a></p>	 <p><b>Sa i Bo</b> Sa i Bo <a href="#">Más información</a></p>

En ella, podemos elegir entre un grupo de programas y ver cualquier video de dicho programa, es decir es como una recomendación de los programas mas interesante que pueden serlo para el usuario o no, pero tiene esa posibilidad y así no tener que buscarlo entre todo el listado de programas.

## Programación

Programación de Televisión  
Lunes 13 de Junio

seleccione un día Lunes 13 de Junio

3ª VUELTA	2ª VUELTA	HORA INICIO	CONTENIDO
01:30	17:30	09:30	EURONEWS CONEXIÓN EN DIRECTO CON EURONEWS
02:00	18:00	10:00	DOCUMENTALES HUMEDALES ESPAÑOLES
02:23	18:23	10:23	CORTOMETRAJE REBORIN
02:29	18:29	10:29	FORO ABIERTO FORO SOBRE INNOVACIÓN, ECONOMÍA Y CALIDAD DE VIDA. OPORTUNIDADES EN LAS ...
03:20	19:20	11:20	DOCUMENTALES LA VIDA EN EL ESTANQUE
03:30	19:30	11:30	POLÍGLotas [72]
04:00	20:00	12:00	PUEBLOS ORIGINARIOS TARJETES, DEL MONTE A LA CIUDAD
04:34	20:34	12:34	DOCE MIRADAS AL UNIVERSO PLANETOLOGÍA, DE MERCURIO A LOS EXOPLANETAS
04:59	20:59	12:59	EURONEWS CONEXIÓN EN DIRECTO CON EURONEWS
06:30	21:30	13:30	SAI BO [185] 20 AÑOS DE SANDÍAS SIN PERITAS BOUQUET
06:19	22:19	14:19	UNIVERSO EXTRAORDINARIO EL SISTEMA SOLAR
06:29	22:29	14:29	EL PÚBLICO LEE ANTONIO RIVERO (2)
07:29	23:29	15:29	UPV NOTICIAS UPV NOTICIAS
07:45	23:45	15:45	UNIVERSO EXTRAORDINARIO ARTE PRIMITIVO Y DE MESOPOTAMIA
07:58	23:58	15:58	DOCUMENTALES LAHA

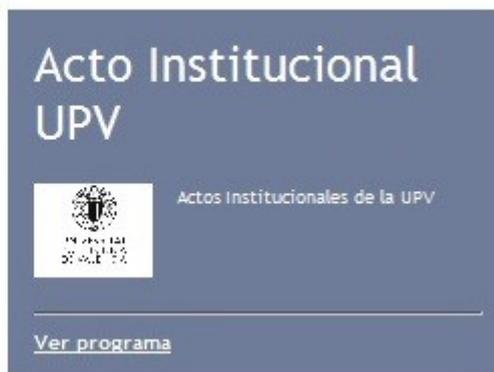
En este submenu, podemos echar un vistazo a toda la programación del día de hoy o del día que queramos dentro del rango de 7 días.

## Programas

Si no hemos encontrado lo que necesitábamos en el apartado de TV a la carta, podemos ir a este apartado y buscar el programa que nos interesa y ver sus capítulos.

### Programas de UPV TV

- [Acto Institucional UPV](#)
- [Ágora Valencia](#)
- [Antena Paranoica](#)
- [Caminos en el Aire](#)
- [Campus Link](#)
- [Cartelera de Cine](#)
- [Ciudadano Freak](#)
- [Conciertos](#)
- [Conferencia](#)
- [Crónicas de la crisis](#)
- [Directo al empleo](#)
- [Documentales](#)
- [FnCartel](#)



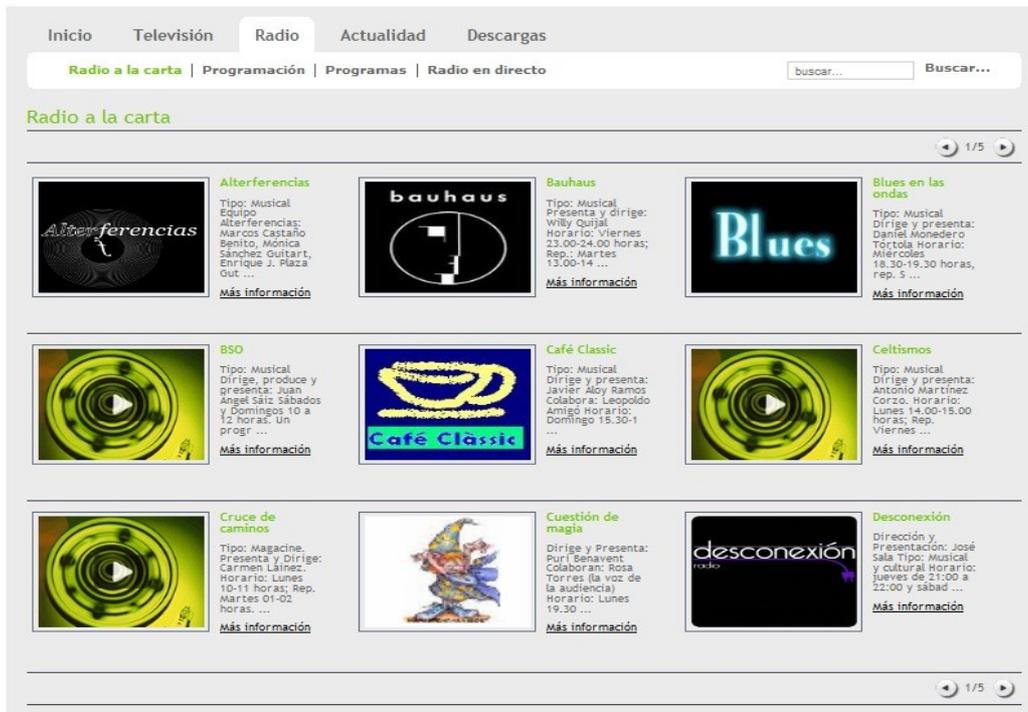
## TV en directo

Francisco Miguel Pardo Veses

En este ultimo submenu, encontramos como bien dice su nombre, un reproductor de video embebido en la pagina web en donde podemos ver la televisión en directo.



## Radio



En esta pestaña, tenemos los mismos submenús, que en Televisión y los mismos contenidos, pero para los programas de radio.

## Actualidad



En esta sección de la página web, podemos consultar las últimas noticias sobre la televisión y la radio de la UPV.

## Descargas



Y en la última pestaña podemos descargar la programación tanto de la televisión como de la radio de la UPV en formato PDF.

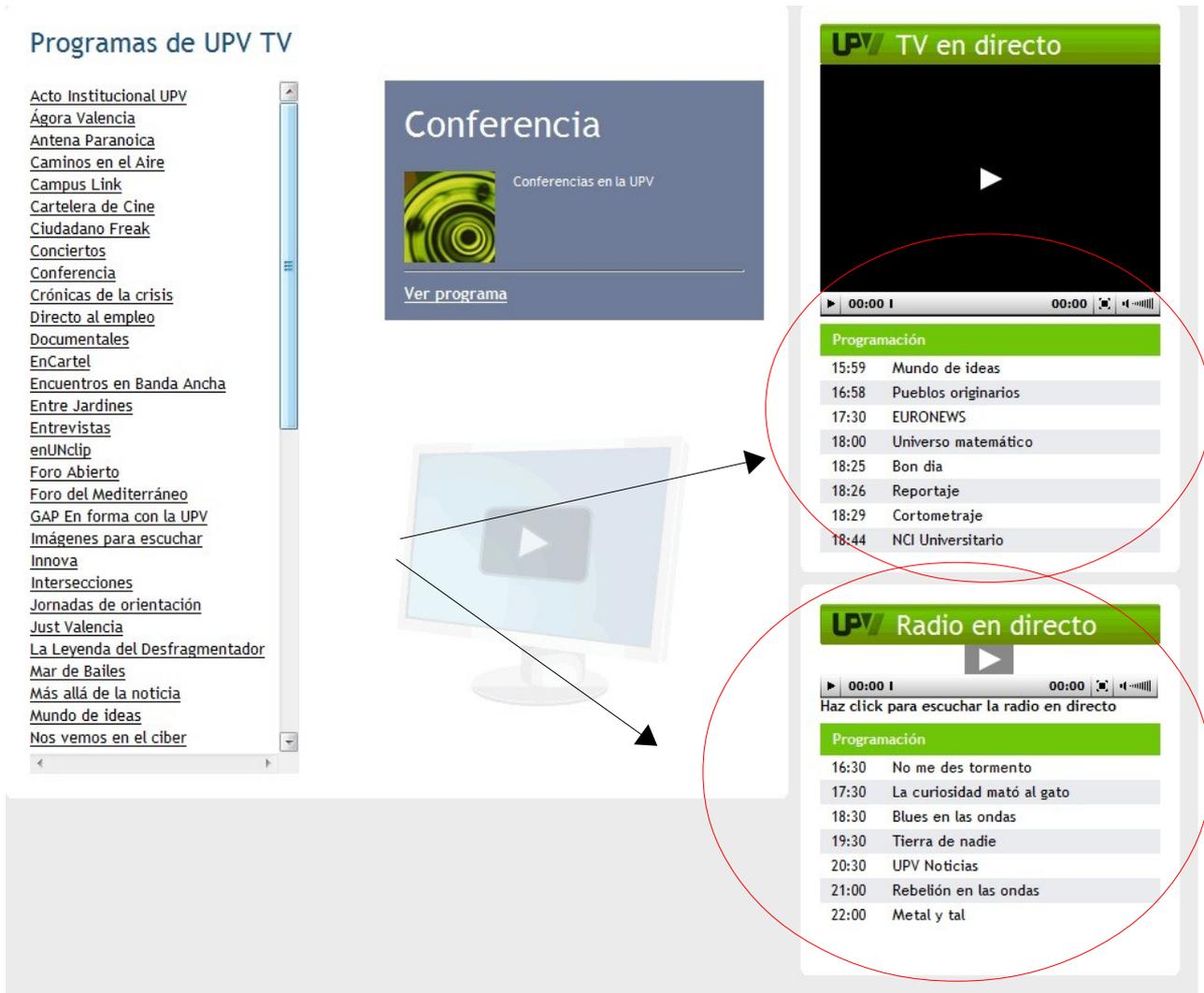
Una vez vista cada parte de la web, procederemos a analizar la web y hablar sobre ella y que nos servirá de inspiración para la aplicación móvil.

Para comenzar, cabe destacar la buena distribución de las secciones ya que el usuario las tiene muy visibles y claras para poder acceder a ellas, esto será una de las cosas que mantendremos en la aplicación móvil. Pero por el contrario disponemos de mucha información y puede resultar algo tedioso si necesitamos encontrar algo, aunque disponemos de un buscador en la web, en la aplicación móvil conseguimos minimizar al máximo el tiempo de búsqueda ofreciendo al usuario lo que necesita directamente sin hacerle navegar más de lo necesario.

Una baza importante que tiene la web, son los accesos directos que dispone en la página de inicio en donde tenemos un apartado de “Destacados” y tres más debajo de él, en los que podemos acceder a los programas destacados y otra información de la página sin tener que navegar hasta ellos, simplemente clickeando en ellos.



También como se puede observar en la web, disponemos de múltiples opciones dentro de la misma pestaña (podemos acceder a la televisión online, radio online, la programación...), lo cual está muy bien ya que el usuario dispone de la máxima información en cualquier pestaña sin tener que volver hacia atrás.



Como último punto de este apartado de Análisis del Problema, comentaremos el punto de partida del cual parte nuestro proyecto y como se ha enfocado.

Este proyecto parte de la idea de expandir y darse a conocer a gente de cualquier parte del mundo mediante una aplicación en S.O Android, que dispone de un sistema de distribución muy concurrido. Una de las premisas fundamentales del proyecto es que fuera directo y claro, para que el usuario encontrara rápidamente lo que quiere y pueda consultarlo en el acto.

## 3.Especificación de requisitos

### 3.1.Introducción

#### 3.1.1.Propósito

El propósito de esta ERS es definir cómo y cuales son los parámetros necesarios para realizar una aplicación software de calidad, estableciendo unas directrices por las que nos regiremos a lo largo de todo el proyecto.

Para ello estableceremos una jerarquía por importancia entre los diferentes elementos, estableceremos las bases para un correcto funcionamiento y mostraremos la verificabilidad del proyecto, así como su estabilidad y posibles modificaciones que se puedan realizar en un futuro.

#### 3.1.2.Ámbito

Los profesionales que trabajan en la RTV UPV, desean dar a conocer sus contenidos a través de la tecnología móvil actual para expandir su numero de seguidores.

Por ello, desarrollaremos una aplicación para dispositivos Android (UPV TV).

En ella podremos visualizar y escuchar todos los contenidos de su pagina web en cualquier lugar sin tener que conectarnos a ella.

#### 3.1.3.Definiciones, acrónimos y abreviaturas

- **Android**: es un sistema operativo basado en Linux diseñado originalmente para dispositivos móviles
- **Oracle** : Oracle es un sistema de gestión de base de datos objeto-relacional, desarrollado por Oracle Corporation.
- **3G**: Es la abreviación de tercera generación de transmisión de voz y datos a través de telefonía móvil.

### 3.1.4 Referencias

Para la realización de este proyecto, se han tenido en cuenta:

- Guía del IEEE 839 para la Especificación de Requerimientos Software.
- Pagina Web de la wikipedia <http://es.wikipedia.org>.
- Apuntes de Ingeniería de Software de Sistemas.
- Varios proyectos anteriores para distintas consultas.

### 3.1.5. Visión global

A través de este documento, vamos a mostrar las diferentes fases por las que nuestro proyecto va a evolucionar, hasta llegar a completar las tareas para los que ha sido diseñado.

Primero nos centraremos en una visión más general de nuestro proyecto, viendo los aspectos generales de la aplicación, para posteriormente hacer mayor hincapié en cada uno de los componentes por separados, de las características principales de cada uno.

Analizaremos los distintos casos de uso y los diagramas de clases para tener una visión general y una primera posible implementación.

Después, explicaremos el diseño de nuestra aplicación, con las distintas posibilidades que posee, y terminaremos hablando sobre la implementación de la aplicación, las funciones principales, sus posibilidades y futuras mejoras u opciones de las que podemos dotar a la aplicación.

## 3.2. Descripción general

### 3.2.1. Perspectiva del producto

La aplicación pretende dar a conocer los contenidos tanto de la televisión como de la radio de la UPV a través de un sector de rápida expansión como son los dispositivos móviles y para ser más exactos, los Smartphones con S.O Android.

Mediante esta aplicación cualquier persona que se la descargue puede ver los contenidos de la web sin estar conectado a ella desde cualquier lugar con WIFI o a través de su conexión 3G.

### **3.2.2.Funciones del producto**

Aplicación:

Usuario:

- Reproducir capítulos de programas de TV
- Reproducir capítulos de programas de radio
- Consultar la cartelera tanto de radio
- Consultar listado de programas tanto de radio como de televisión
- Consultar la pagina web de la UPV TV
- Consultar la pagina del Youtube de la UPV TV

### **3.2.3.Características del usuario**

Esta aplicación esta diseñada para cualquier tipo de usuario, con grandes , pocos o nulos conocimientos en informática. Esto se debe a que el planteamiento de la aplicación es muy intuitivo y sencillo, y en un breve espacio de tiempo puedes estar viendo cualquier contenido.

Al no existir registro ni un login, todos los usuarios, pueden realizar todas las acciones, no existen restricciones de ningún tipo.

### **3.2.4.Restricciones generales**

No existen restricciones generales, ya que cualquier usuario que disponga de un dispositivo Android, podrá realizar todas las funciones anteriormente mencionadas.

### **3.2.4.Supuestos y dependencias**

La única dependencia es que para poder utilizar la aplicación debemos tener un Smartphone con S.O Android.

### **3.3.Requerimientos específicos**

#### **3.3.1.Requisitos de Interfaces Externas**

La aplicación se intenta hacer lo mas agradable, sencilla y cómoda para el usuario para que su navegación a través de ella sea simple y intuitiva. Para conseguir esto, la aplicación dispone de la siguiente estructura:

- Un menú principal muy sencillo, directo y agradable a la vista del usuario.
- En el menú principal podemos observar arriba del todo el logotipo de la UPV TV.
- Este menú principal tenemos 3 botones correspondientes a la televisión, la radio y las noticias.
  
- En las secciones de radio y televisión, cada una dispone de un menú compuesto de tres pestañas donde podemos consultar:
  - Cartelera
  - Programación a la carta
  - Programas en general.
  
- En la parte inferior de la aplicación si apretamos el botón “menú” del dispositivo móvil, nos aparecerán 3 opciones que son :
  - Ir a web
  - Atrás
  - Acerca de

#### **3.3.2.Requerimientos de eficiencia**

Esta aplicación no necesita de una base de datos interna en la aplicación, por ello todos puedes hacerlo todo. Además todo funciona a través de XML. En este caso cabe destacar la rapidez y fluidez de este sistema.

#### **3.3.3.Atributo**

- Seguridad

Ya que nuestra aplicación no se comunica con ninguna BBDD debido a que todo es a través de unos XML formados mediante unos procedimientos PL/SQL, ni es necesario identificarse para poder utilizarla, el nivel de seguridad que se alcanza con esta estructura es elevado.

### 3.3.4.Requisitos Funcionales

En esta sección trataremos de explicar las diferentes funciones que llevará a cabo nuestra aplicación.

#### Usuario Anónimo

- Consulta del contenido de la radio
  - Mediante un TabHost puedes acceder al contenido de la radio, divididos en 3 pestañas
    - Consulta de la cartelera de la radio
      - Salida: Un ListView muestra al usuario toda la cartelera del día
    - Consulta de la carta de radio
      - Salida: Un ListView muestra la carta de programas de la radio
    - Consulta de los programas de radio
      - Salida: Un ListView muestra al usuario la totalidad de los programas.
- Consulta del contenido de la televisión
  - Mediante un TabHost puedes acceder al contenido de la televisión, divididos en 3 pestañas
    - Consulta de la cartelera de la televisión
      - Salida: Un ListView muestra al usuario toda la cartelera del día
    - Consulta de la carta de televisión
      - Salida: Un ListView muestra la carta de programas de la televisión
    - Consulta de los programas de televisión
      - Salida: Un ListView muestra al usuario la totalidad de los programas.
- Consulta y navegación de la pagina de Youtube de la UPV TV
  - Salida:A través de un WebView puedes consultar la pagina web.

## 4.0.Análisis

En el siguiente apartado vamos a enumerar las fases del análisis del proyecto, en el cual, se describirá su estructura y funcionalidad mediante distintos diagramas, que permiten comprender un poco más a fondo cómo funcionará la misma.

Los siguientes diagramas describen de manera gráfica, cada una de las funciones que llevara a cabo la aplicación.

Vamos a utilizar el lenguaje UML de propósito general para visualizar de una manera más cómoda e intuitiva.

### 4.1.Diagrama de Casos de Uso

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

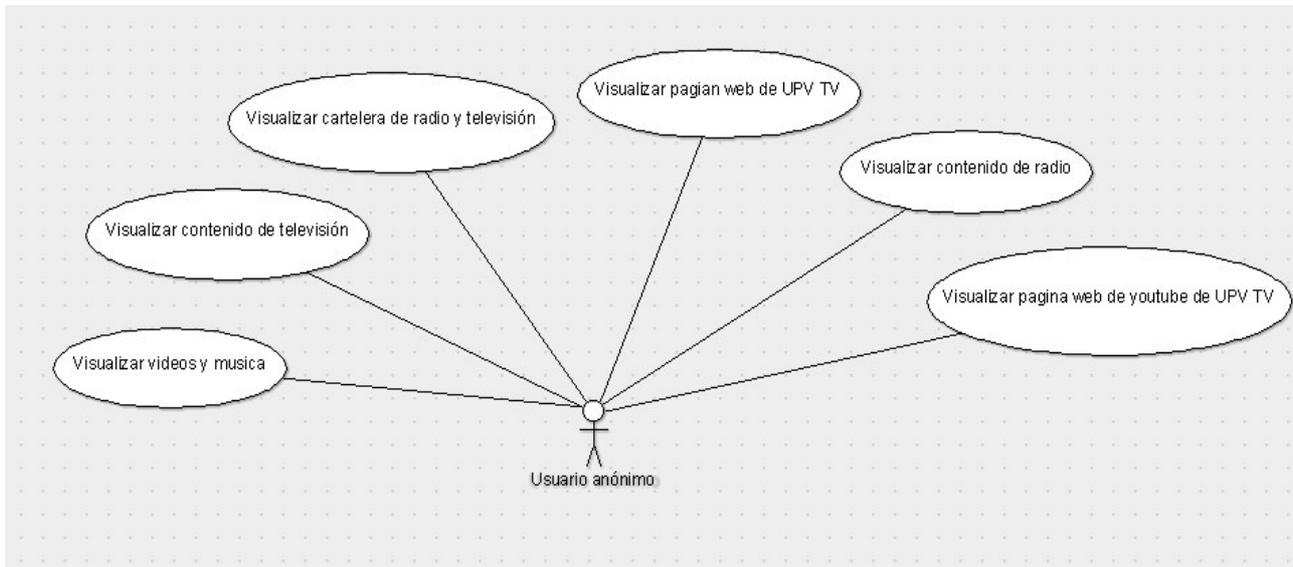
Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente.

Un caso de uso debe especificar un comportamiento deseado, pero no imponer cómo se llevará a cabo ese comportamiento, es decir, debe decir QUÉ pero no CÓMO. Esto se realiza utilizando escenarios.

A continuación vamos a mostrar el modelo de nuestro sistema con el actor implicado en el



*Actor del Diagrama de casos de uso*



*Diagrama de casos de uso*

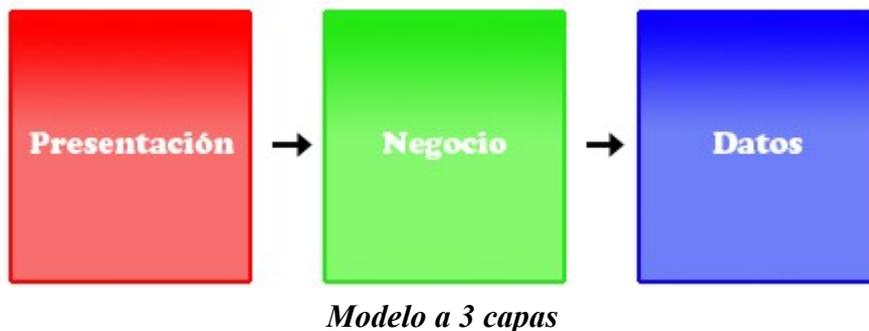
En la ilustración podemos observar el diagrama de casos de uso, solamente disponemos de 1 debido a que como solo tenemos un usuario(Usuario anónimo), este usuario realiza todas las acciones disponibles en la aplicación

## 5.0.Diseño

Para la aplicación móvil, se estudio que arquitectura era la mas adecuada y después de sopesar todas las opciones, nos hemos decantado por el modelo de arquitectura en tres capas. Este estilo de programación permite una abstracción de las partes bien diferenciadas en la aplicación y esto conlleva que una modificación en una capa superior, no afecta a una que esta por debajo.

Además, cada una de ellas está conectada a la anterior, con tal de dotar a la aplicación de toda su funcionalidad y una movilidad entre las distintas capas.

### 5.1.Modelo en tres capas



#### 5.1.1.Capa de presentación

Esta capa resuelve la presentación de datos al usuario. Esta capa se encarga entonces de "dibujar" las pantallas de la aplicación al usuario, y tomar los eventos que el cliente genere. Existen numerosas tecnologías para esta capa (JSP, Struts, JSF, Flex, Velocity, Wicket, GWT, etc, etc), cada una con sus ventajas y desventajas.

Esta capa es la encargada de interactuar con el usuario de una representación visual. En esta capa también se ofrecen servicios al usuario.

Por ello tanto su contenido como su usabilidad, deben de ser lo más fácil posible, ya que será esta capa, quien refleje el contenido final de la aplicación.

A continuación mostrare algunas pantallas de la aplicación y comentare su contenido.







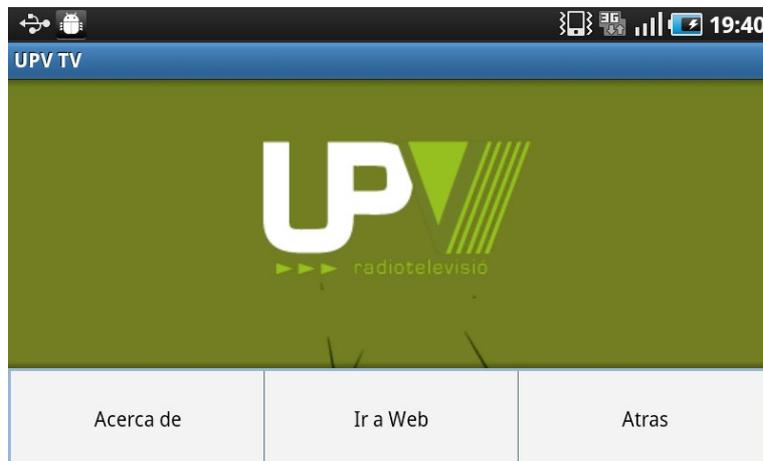


Esta es la pantalla principal de la aplicación y desde donde se accede al resto del contenido de la aplicación. Como se puede observar consta de una cabecera en donde aparece el logotipo de la **UPV radiotelevisión**. Debajo de esta cabecera vemos 4 imágenes centrales que corresponden a los 4 apartados principales de la aplicación



Francisco Miguel Pardo Veses

También disponemos de un menú emergente que aparece cuando apretamos la tecla “menú” del dispositivo móvil.



Este menú emergente, se ha puesto para no sobrecargar el menú principal y dejarlo lo mas sencillo y agradable posible.

Las secciones de “Radio” y “Televisión” se dividen a su vez en 3 pestañas su contenido, que son :

- Programación
- Carta
- Programas



En este apartado el usuario puede consultar la programación del día actual, en cada apartado de la programación veremos la hora de inicio, el nombre del programa y si obtiene información sobre el capítulo que se emitirá, aparecerá también en la aplicación.



En el apartado de “Carta” observamos una lista de los programas que pertenecen a la carta de radio o de televisión respectivamente.



Y el ultimo apartado “Programas” tenemos una lista con todos los programas de radio o televisión respectivamente.

### 5.1.2.Capa de Negocio

En esta capa es donde residen todas las peticiones del usuario, es decir, el acceso a la información de cartelera, el listado de todos los programas tanto de radio como de televisión.

Francisco Miguel Pardo Veses

También sirve de enlace entre las otras dos capas, ya que la capa de presentación no accede a la base de datos directamente, sino que ésta se comunica con la capa de aplicación para demandarle el servicio deseado y la capa de aplicación se comunica con la capa de persistencia para recuperar los datos necesarios. A continuación mostrare unos de los procedimientos utilizados en la aplicación, las variables y los nombres de las tablas han sido modificados por motivos de seguridad.

**procedure** ListaProgRadioLis\_movil **is**

l\_prog **varchar2**(250);

l\_rss **varchar2**(250);

l\_num **integer**;

l\_cad **varchar2**(250);

l\_desc **varchar2**(250);

l\_foto **varchar2**(500);

**begin**

**for** r **in** (**select** datos **from** tablaPrograma **where** list\_prog=1 **and** tipo='R' **and** activo='S' **order**  
**by** 2)

**loop**

l\_prog:=r.tit;

**end loop**;

**if** l\_prog **is null** **then return**; **end if**;

owa\_util.mime\_header('text/xml', **false**, 'ISO-8859-15');

owa\_util.http\_header\_close;

-- http.p('<?xml version="1.0" encoding="ISO-8859-15"?>');

http.p('<?xml version="1.0" encoding="ISO-8859-15"?>

<rss version="2.0">

<channel>

<title><![CDATA[||l\_cad||]]></title>

<link>http://www.upv.es/pls/oreg/rtv\_web.Ppal?p\_idioma=||l\_id||</link>

<description><![CDATA[||l\_Desc||]]></description>

<language>es</language>

<copyright><![CDATA[Copyright 2009, RTV Universitat Polit&egrave;cnica de  
Val&egrave;ncia]]></copyright>

```
<pubDate>||to_char(systimestamp,'Dy, dd Mon yyyy hh24:mi:ss
TZHTZM','nls_date_language=english')||</pubDate>
<lastBuildDate>||to_char(systimestamp,'Dy, dd Mon yyyy hh24:mi:ss
TZHTZM','nls_date_language=english')||</lastBuildDate>
<ttl>60</ttl>
<image>
  <title>UPV</title>
  <url>http://www.upv.es/rtv/nueva/img/logo.gif</url>
  <link>http://www.upv.es/pls/oreg/rtv_web.Ppal?p_idioma='||l_id||'</link>
  <width>120</width>
  <height>43</height>
  <description>Logo RTV-UPV</description>
</image>');
l_num:=1;
for r in (select datos
          from tablaPrograma where en_lista_prog=1 and tipo='R' and activo='S' order by 2)
loop
-- Si hay capítulo, usar su foto si tiene y usar su título como tooltip
  if r.foto is not null then
    l_foto:=http://www.upv.es/pls/drtv/rtv_utl.GetFoto?p_idioma=c&p_foto='||r.foto;
  else
    l_foto:=http://www.upv.es/rtv/nueva/img/alacarta/1.gif;
  end if;
  l_num:=l_num+1;
  if l_num>g_max_rss then
    htp.p('</channel></rss>');
    return;
  end if;

  l_cad:=http://www.upv.es/pls/oreg/rtv_web.ProgRadioFicha?p_id='||
r.id||'&p_idioma=c&p_cap_id=';
  htp.p('<item>
    <title><![CDATA[||r.tit||]]></title>
    <description><![CDATA[||r.tit||]]></description>
    <pubDate>||to_char(systimestamp,'Dy, dd Mon yyyy hh24:mi:ss
```

Francisco Miguel Pardo Veses

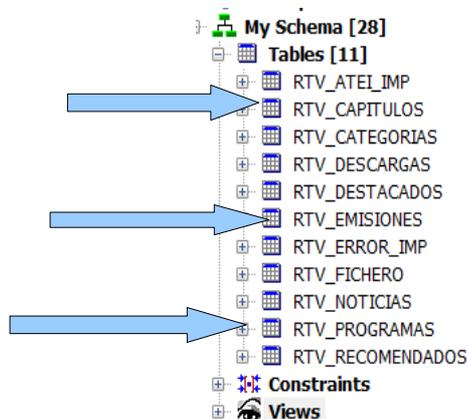
```
TZHTZM',nls_date_language=english')||'</pubDate>
    <RSS>http://www.upv.es/pls/drtv/rtv_web.rss_radio_movil?p_id='||
r.id||'&amp;p_idioma=c</RSS>
    <foto>'||l_foto||'</foto>
</item>');
end loop;
htp.p('</channel></rss>');
exception
    when others then
        sic_upve.error('Error: '||sqlerrm);
end ListaProgRadioLis_movil;
```

### 5.1.3.Capa de Datos

La capa de persistencia es la que dota al sistema de poder mantener los datos de una forma persistente como su propio nombre indica, es decir, mantener información de una sesión a otra aunque el sistema no esté en uso.

Es el único nivel capaz de acceder a la información contenida en las tablas de la base de datos a través de consultas a los datos.

En nuestra aplicación se utiliza a una base de datos Oracle que se encuentra en un servidor externo a la aplicación y la comunicación con ella es mediante procedimientos almacenados. Estas son las tablas:



## 6.Sistema de información

### 6.1.Comunicación en la aplicación e infraestructura

En este apartado comentaremos como es la comunicación de la aplicación con la base de datos y como se realiza.

Francisco Miguel Pardo Veses

Toda la aplicación realiza su comunicación con la base de datos mediante 6 procedimientos almacenados en la base de datos. Estos procedimientos crean un XML que nuestra aplicación parsea en tiempo real y recoge los datos necesarios para mostrarlos posteriormente en la aplicación.

A continuación mostrare un ejemplo de un procedimiento que utiliza la aplicación, veremos que muestra al poner la dirección en un navegador cualquiera y después que tiene por debajo que es lo realmente utiliza nuestra aplicación para su buen funcionamiento.

 [http://www.upv.es/pls/drtv/rtv\\_web.TVCartaLis\\_movil](http://www.upv.es/pls/drtv/rtv_web.TVCartaLis_movil)

**Policiencia**

sábado, 25 de junio de 2011 10:48

Policiencia

**Trasfondo**

sábado, 25 de junio de 2011 10:48

Trasfondo

**5 minuts de ciència**

sábado, 25 de junio de 2011 10:48

5 minuts de ciència

**Más allá de la noticia**

sábado, 25 de junio de 2011 10:48

Más allá de la noticia

**Caminos en el Aire**

sábado, 25 de junio de 2011 10:48

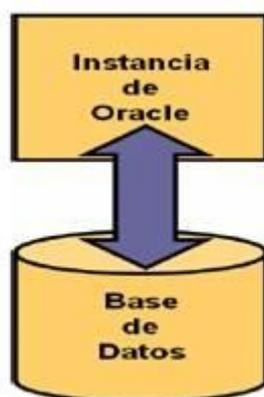
*Visualización del contenido en un navegador*

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<rss version="2.0">
<channel>
  <title><![CDATA[Pedazo de Ciudad]]></title>
  <link>http://www.upv.es/pls/oreg/rtv_web.Fpal?p_idioma=c</link>
  <description><![CDATA[Pedazo de Ciudad]]></description>
  <language>es</language>
  <copyright><![CDATA[Copyright 2009, RTV Universitat Polit&egrave;cnica de Val&egrave;ncia]]></copyright>
  <pubDate>Sat, 25 Jun 2011 10:48:49 +0200</pubDate>
  <lastBuildDate>Sat, 25 Jun 2011 10:48:49 +0200</lastBuildDate>
  <ttl>60</ttl>
<image>
  <title>UPV</title>
  <url>http://www.upv.es/rtv/nueva/img/logo.gif</url>
  <link>http://www.upv.es/pls/oreg/rtv_web.Fpal?p_idioma=c</link>
  <width>120</width>
  <height>43</height>
  <description>Logo RTV-UPV</description>
</image>
  <item>
    <title><![CDATA[Policiencia]]></title>
    <description><![CDATA[Policiencia]]></description>
    <pubDate>Sat, 25 Jun 2011 10:48:49 +0200</pubDate>
    <RSS>http://www.upv.es/pls/drtv/rtv_web.rss_tv_movil?p_id=1603&p_idioma=c</RSS>
    <foto>http://www.upv.es/pls/drtv/rtv_utl.GetFoto?p_idioma=c&p_foto=F11894/Policiencia.JPG</foto>
  </item>
  <item>
    <title><![CDATA[Trasfondo]]></title>
    <description><![CDATA[Trasfondo]]></description>
    <pubDate>Sat, 25 Jun 2011 10:48:49 +0200</pubDate>
    <RSS>http://www.upv.es/pls/drtv/rtv_web.rss_tv_movil?p_id=932&p_idioma=c</RSS>
    <foto>http://www.upv.es/pls/drtv/rtv_utl.GetFoto?p_idioma=c&p_foto=F19034/Trasfondo.jpg</foto>
  </item>
</channel>
</rss>
```

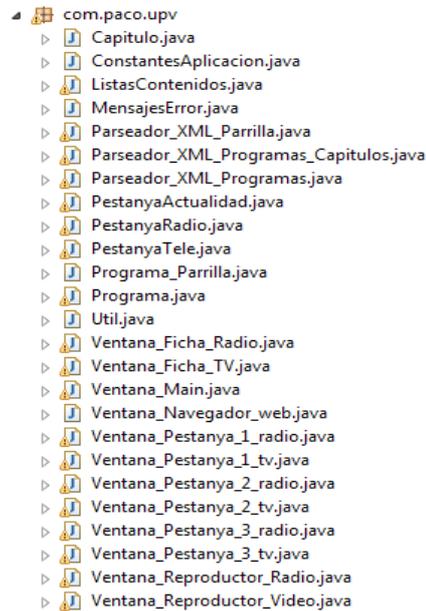
*Estructura XML de la respuesta del procedimiento*

Esta estructura es la que nuestro programa reconoce y puede interpretar, ya que es fácilmente parseable ya que es un lenguaje estructurado. También se puede parsear una pagina HTML , pero a la hora de la verdad almacenarías mucha basura y elementos innecesarios.

Por motivos de seguridad, no podemos mostrar el código que tiene el procedimiento almacenado, ya que al utilizar una base de datos de la UPV, no es recomendable mostrar nombres de tablas y otros contenidos, que podrían ser utilizados para intentar atacarla. De este tema, tan solo podemos comentar que son 6 procedimientos almacenados creados para esta aplicación y que cuando accedemos a una dirección del procedimiento, es este el que realiza las consultas necesarias y lo estructura en formato XML para poder ser utilizado.



## 7. Estructura del proyecto



En este apartado hablaremos sobre como esta estructurado el proyecto y como se relacionan las clases unas con otras. Primero tenemos que diferenciar el proyecto en varios grupos, que son los siguientes:

- Pantallas
- Objetos
- Constantes
- Parseadores
- Utilidades

### 7.1. Pantallas

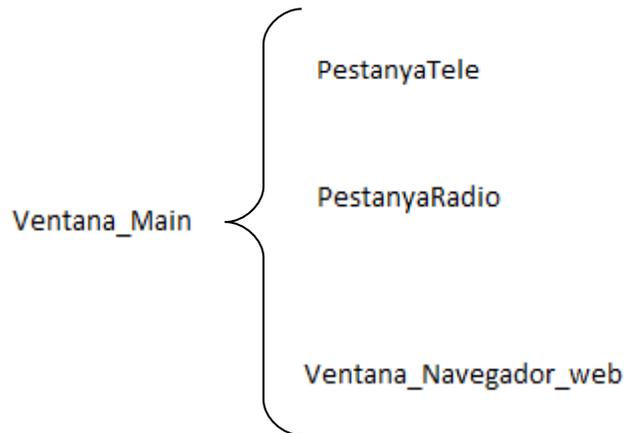
En este grupo, encontramos las siguientes clases:

- Ventana\_Ficha\_Radio.java
- Ventana\_Ficha\_TV.java
- Ventana\_Main.java
- Ventana\_Navegador\_web.java
- Ventana\_Pestanya\_1\_radio.java
- Ventana\_Pestanya\_2\_radio.java
- Ventana\_Pestanya\_3\_radio.java
- Ventana\_Pestanya\_1\_tv.java
- Ventana\_Pestanya\_2\_tv.java
- Ventana\_Pestanya\_3\_tv.java

Francisco Miguel Pardo Veses

- Ventana\_Reproductor\_Radio.java
- Ventana\_Reproductor\_Video.java
- PestanyaRadio.java
- PestanyaTelevision.java

En este grupo de pantallas, el orden que sigue es el siguiente. Primero tenemos la **Ventana\_Main**, que es la que contiene los accesos a las distintas pantallas de la aplicación,



Aunque tenemos 4 opciones en el menú principal(**Ventana\_Main**), las opciones “Actualidad” y “Youtube” ambas se dirigen al navegador web cada una con una dirección web.

Las dos opciones **PestanyaTele** y **PestanyaRadio**, son contenedores donde declaramos las pestañas que va incluir, es el esqueleto de las tablas.



Francisco Miguel Pardo Veses

En ambos contenedores se declaran las pestañas que va a contener, su título y su icono. Aquí muestro un ejemplo del método, existe un método por pestaña.

```
private void addTabUno() {  
    Intent intent = new Intent(this, Ventana_Pestanya_1_radio.class);  
  
    TabSpec spec = mTabHost.newTabSpec(TAG_OPTIONS);  
    spec.setIndicator(mResources.getString(R.string.pestanya1), mResources  
        .getDrawable(R.drawable.programacion));  
    spec.setContent(intent);  
    mTabHost.addTab(spec);  
}
```

Una vez dentro de las pestañas, cuando seleccionamos alguna opción de la **Ventana\_Pestanya\_2** o **Ventana\_Pestanya\_3** tanto de radio como de televisión, nos lanza a otra pantalla que son la de **Ventana\_Ficha\_Radio** o **Ventana\_Ficha\_TV** respectivamente. Ambas ventanas corresponden a la pestañas llamadas “Carta” y “Programas” respectivamente.



Estas dos ventanas, llama a la clase de ficha de radio o televisión haciendo un click en la opción correspondiente y dándole como atributo la URL del programa seleccionado, mas adelante comentare que hace con esta URL. Una vez nos encontramos en las pantallas de ficha, obtenemos una lista de los capítulos de dicho programa, ya sea de radio o televisión y pulsando en el capítulo correspondiente, iremos a otra la ultima pantalla de la aplicación que son o bien el reproductor de video o bien el reproductor de audio en donde le daremos como atributos las direcciones del archivo a reproducir, ya sea audio o video.



En esta ultima pantalla, conectamos con la dirección que hemos recibido de la pantalla anterior y mostramos en streaming el contenido del archivo.

## 7.2.Objetos

En este grupo tenemos estas clases:

- Programa.java
- Capitulo.java
- Programa\_Parrilla.java

En cada uno se almacenada el contenido de los XML que parseamos con las clases parseadoras. La estructura de cada uno es la siguiente:

### Programa

Contiene todo los atributos del programa en cuestión:

- Titulo(titulo)
- Descripción(descripcion)
- URL del procedimiento almacenado del programa(url)
- Fecha(fecha)
- Dirección de la foto del programa(dirFoto)

### Capitulo

Contiene todos los atributos del capitulo, mediante la dirección url del programa, obtenemos la lista de capítulos del programa.

- Titulo del capitulo(titulo)
- URL del archivo para reproducir(url\_capitulo)

### Programa\_Parrilla

Contiene los atributos de cada programa que aparece en la parrilla, contiene la fecha del inicio, el titulo del capitulo y el titulo del programa

- Titulo del programa(titulo)
- Titulo del capitulo del programa(capitulo)
- Hora de inicio del programa(hora\_ini)

### 7.3.Constantes

En este grupo tenemos todas aquellas constantes de la aplicación, esto nos proporciona facilidad a la hora de modificar los atributos que son invariables durante toda la ejecución de la aplicación, estas clases son:

- ConstantesAplicación.java
- MensajesError.java

Dentro de estas clases, tenemos por una parte:

#### ConstantesAplicación

En esta clase tenemos las direcciones correspondientes a los procedimientos almacenados que se han creado en la base de datos:

- Dirección del procedimiento almacenado de la cartelera de la radio(dirCarterlaRadio)
- Dirección del procedimiento almacenado de la cartelera de la televisión(dirCarteleraTele)
- Dirección del procedimiento almacenado de la televisión a la carta (dirCartaTele)
- Dirección del procedimiento almacenado de todos los programas de televisión(dirProgTele)
- Dirección del procedimiento almacenado de la cartelera de la radio(dirCarteleraRadio)
- Dirección del procedimiento almacenado de la radio a la carta(dirCartaRadio)
- Dirección del procedimiento almacenado de todos los programas de la radio(dirProgRadio)
- Dirección de la pagina web de la UPV radiotelevision (dirWeb)
- Dirección de la pagina de Youtube de la UPV radiotelevision(dirWebYoutube)
- Dirección de la pagina de las noticias de la UPV radiotelevision(dirNoticiasWeb)

#### MensajesError

En esta clase, como bien indica su nombre almacenamos todos los mensajes de error de la aplicación para poder tenerlos todos en un mismo sitio y poder añadir o quitarlos sin tener que buscarlos en cada clase.

## 7.4.Parseadores

En este ultimo grupo, tenemos las clases que son el puente entre los procedimientos de la base de datos y la aplicación . Tenemos 3 clases que corresponden a este grupo:

- Parseador\_XML\_Parrilla.java
- Parseador\_XML\_Programas.java
- Parseador\_XML\_Programas\_Capitulos.java

En cada una de estas clases parseamos un procedimiento almacenado diferente y se obtienen distintos atributos para poder mostrarlos posteriormente en la aplicación.

### Parseador\_XML\_Parrilla

En esta clase parseadora, el funcionamiento es el siguiente. Primero le pasamos la dirección del procedimiento y se comprueba si es accesible y si es correcto. Una vez esta todo correcto, se realiza una lectura del XML y se decide que hacer con que etiqueta. En esta clase, las etiquetas que son identificadoras para obtener sus datos, son :

- <hora\_ini>
  - Corresponde al atributo **hora\_ini** de Programa\_Parrilla.java
- <capitulo>
  - Corresponde al atributo **capitulo** de Programa\_Parrilla.java
- <programa>
  - Corresponde al atributo **programa** de Programa\_Parrilla.java

Para conseguir llegar a la etiqueta correspondiente tienes que recorrer el XML hasta la etiqueta <item> que es la que inicia el proceso de obtención de datos.

```
RootElement root = new RootElement("rss");
Element channel = root.getChild("channel");
Element item = channel.getChild("item");

item.setStartElementListener(new StartElementListener() {
    public void start(Attributes attrs) {
        prog = new Programa_Parrilla();
    }
});
```

Y aquí que se hace cada vez que encuentra la etiqueta que necesitamos, en nuestro caso añadimos cada atributo del XML a su atributo correspondiente en el programa y cuando encuentra el fin de la etiqueta <item>, añadimos a la lista que mostraremos posteriormente.

```
item.getChild("hora_ini").setEndElementListener(  
    new EndTextElementListener() {  
        public void end(String body)  
        {  
            prog.setHora_ini(body);  
        }  
    });
```

## Parseador\_XML\_Programas

En esta clase parseadora, el funcionamiento es el siguiente. Primero le pasamos la dirección del procedimiento y se comprueba si es accesible y si es correcto. Una vez esta todo correcto, se realiza una lectura del XML y se decide que hacer con que etiqueta. En esta clase, las etiquetas que son identificadoras para obtener sus datos, son :

- <title>
  - Corresponde al atributo **titulo** de Programa.java
- <foto>
  - Corresponde al atributo **dirFoto** de Programa.java
- <RSS>
  - Corresponde al atributo **url** de Programa.java

Para conseguir llegar a la etiqueta correspondiente tienes que recorrer el XML hasta la etiqueta <item> que es la que inicia el proceso de obtención de datos y cuando encuentra el fin de la etiqueta <item>, añadimos a la lista que mostraremos posteriormente.

## Parseador\_XML\_Programas\_Capitulos

En esta clase parseadora, el funcionamiento es el siguiente. Primero le pasamos la dirección del procedimiento y se comprueba si es accesible y si es correcto. Una vez esta todo correcto, se realiza una lectura del XML y se decide que hacer con que etiqueta. En esta clase, las etiquetas que son identificadoras para obtener sus datos, son :

- <title>
  - Corresponde al atributo **titulo** de Capitulo.java
- <url>
  - Corresponde al **url\_capitulo** de Capitulo.java

Para conseguir llegar a la etiqueta correspondiente tienes que recorrer el XML hasta la etiqueta <item> que es la que inicia el proceso de obtención de datos y cuando encuentra el fin de la etiqueta <item>, añadimos a la lista que mostraremos posteriormente.

## 7.5.Utilidades

En este reducido grupo, tenemos una clase llamada Util.java, en donde almacenamos aquellos métodos estáticos que son útiles en la aplicación y que pueden ser utilizados por cualquier otra clase. Por ahora dispone de dos metodos utilizados, que son:

- **public static** boolean checkConex(Context ctx)
- **public static** String validadorDireccion(String dir)

El primer método se encarga de comprobar la conectividad del dispositivo, en este caso el WIFI, si el dispositivo tiene una red WIFI configurada y accesible el método se encarga de conectarlo a la red para poder utilizar la aplicación .

El segundo método se encarga de parsear la dirección del archivo y convertirlo en una dirección valida para poder acceder al archivo que contiene, ya que debemos sustituir por ejemplo los espacio por “%20” y las vocales con acentos también. Entonces este metodo devuelve la dirección correctamente parseada.

## 8.Tecnologías usadas

Las tecnologías utilizadas en este proyecto son las siguiente que después comentaremos:

- SDK de Android
- Java
- XML
- SAX

### 8.1.SDK de Android

Android es un sistema operativo basado en Linux para dispositivos móviles, tales como teléfonos inteligentes o tablets. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005.<sup>7</sup> Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.<sup>8</sup> Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos, en el segundo y tercer trimestres de 2010,<sup>9</sup> 10 11 con una cuota de mercado de 43,6% en el tercer trimestre.<sup>12</sup>

Android tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, existen cerca de 200.000 aplicaciones disponibles para Android. Android Market es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java.<sup>15</sup> Respecto del malware, ya existen varios para este sistema.<sup>16</sup>

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles.<sup>17</sup> 18 Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto.<sup>19</sup> Actualmente Android posee el 32,9% de cuota de mercado a escala mundial de los teléfonos inteligentes, por delante de Symbian OS que tiene 30,6%. En tercer lugar se sitúa Apple con cuota de mercado del 16%.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic.

El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

Francisco Miguel Pardo Veses

En referencia a sus características:

- ***Diseño del dispositivo***

La plataforma es adaptable a pantallas más grandes, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.

- ***Almacenamiento***

SQLite, una base de datos liviana, la cual es usada para propósitos de almacenamiento de datos.

- ***Conectividad***

Android soporta las siguientes tecnologías de conexión: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE y WiMAX.

- ***Mensajería***

SMS y MMS son formas de mensajería, incluyendo mensajería de texto y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de Push Messaging de Android.

- ***Navegador web***

El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador obtiene una puntuación de 93/100 en el test Acid3.

- ***Soporte de Java***

Aunque las aplicaciones son escritas en Java, no hay una Máquina Virtual de Java en la plataforma. El código Java no es ejecutado. El código Java se compila en el ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.

- ***Soporte multimedia***

Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.

- ***Soporte para streaming***

Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.

- ***Soporte para hardware adicional***

Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetro, giroscopios, magnetómetros, sensores de proximidad y de presión, termómetro, aceleración 2d y 3d.

- ***Entorno de desarrollo***

Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse (actualmente 3.4 o 3.5) usando el plugin de Herramientas de Desarrollo de Android.

- ***Market***

El Android Market es un catálogo de aplicaciones que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.

- ***Multi-táctil***

Android tiene soporte nativo para pantallas multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de Kernel (posiblemente para evitar infringir una patente de Apple relacionada con tecnología de pantallas táctiles). Más tarde, Google publicó una actualización que activa el soporte para pantallas multi-táctil de forma nativa

- ***Bluetooth***

El soporte para A2DP y AVRCP fue agregado en la versión 1.5, el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0 y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.

- ***Video llamada***

La versión principal de Android no soporta video llamada. Sin embargo algunos dispositivos podrían tener una versión personalizada del sistema operativo que lo soporta, ya sea por la red del operador (como el Samsung i9000 Galaxy S) o sobre IP.

- ***Tethering***

Android soporta tethering, el cual permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico. Para permitir a un computador portátil usar la conexión 3G del móvil Android se podría requerir la instalación de software adicional.

Francisco Miguel Pardo Veses

De entre todas estas características que hemos visto, las mas importantes para el proyecto son las **multimedia** y el **streaming**, ya que el punto fundamental del proyecto es el de poder reproducir contenido multimedia a través del dispositivo móvil.

## 8.2.JAVA

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típica-mente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrollados por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre diciembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java aún no lo es).

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

La sintaxis de Java se deriva en gran medida de C++. Pero a diferencia de éste, que combina la sintaxis para programación genérica, estructurada y orientada a objetos, Java fue construido desde el principio para ser completamente orientado a objetos. Todo en Java es un objeto (salvo algunas excepciones), y todo en Java reside en alguna clase (recordemos que una clase es un molde a partir del cual pueden crearse varios objetos).

### 8.3.JRE

El JRE (Java Runtime Environment, o Entorno en Tiempo de Ejecución de Java) es el software necesario para ejecutar cualquier aplicación desarrollada para la plataforma Java. El usuario final usa el JRE como parte de paquetes software o plugins (o conectores) en un navegador Web. Sun ofrece también el SDK de Java 2, o JDK (Java Development Kit) en cuyo seno reside el JRE, e incluye herramientas como el compilador de Java, Javadoc para generar documentación o el depurador. Puede también obtenerse como un paquete independiente, y puede considerarse como el entorno necesario para ejecutar una aplicación Java, mientras que un programador debe además contar con otras facilidades que ofrece el JDK.

### 8.4.XML

XML, siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades[cita requerida]. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de trozos de información. Ejemplos son un tema musical, que se compone de compases, que están formados a su vez por notas. Estas partes se llaman elementos, y se las señala mediante etiquetas.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de éste como un elemento. Un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma <nombre>, donde nombre es el nombre del elemento que se está señalando.

A continuación se muestra un ejemplo para entender la estructura de un documento XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">

<item>

  <title></title>
  <description></description>
  <pubDate</pubDate>
  <RSS></RSS>
```

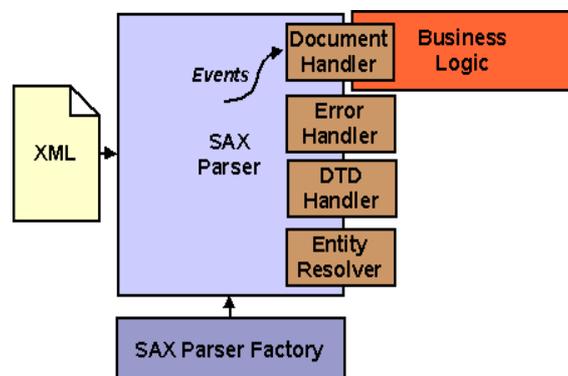
</item>

## Ventajas del XML

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan *bugs* y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos, es decir, podríamos tener una aplicación en Linux con una base de datos Postgres y comunicarla con otra aplicación en Windows y Base de Datos MS-SQL Server.
- Transformamos datos en información, pues se le añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.

## 8.5.SAX

El API SAX (Simple API for XML) usa un modelo basado en eventos y permite el procesamiento de un documento fuente como un stream de eventos. Los eventos son disparados mientras se analiza como un flujo continuo de retro-llamadas e invocaciones a métodos. Los eventos están anidados de la misma forma que los elementos en el documento, por lo tanto, no se crea ningún modelo de documento intermedio. Como el uso de memoria es bajo, el modelo de programación puede ser complejo especialmente si la estructura no corresponde con la estructura de datos de la aplicación. Como genera un flujo temporal de eventos, el API SAX no puede utilizarse cuando un modelo de documento tiene que ser editado o procesado varias veces.



*Esquema de como se utiliza el API SAX*

## 8.6.PL-SQL

PL/SQL (Procedural Language/Structured Query Language) es un lenguaje de programación incrustado en Oracle.

PL/SQL soporta todas las consultas, ya que la manipulación de datos que se usa es la misma que en SQL, incluyendo nuevas características:

- El manejo de variables.
- Estructuras modulares.
- Estructuras de control de flujo y toma de decisiones.
- Control de excepciones.

El lenguaje PL/SQL está incorporado en:

- Servidor de la base de datos.
- Herramientas de Oracle (Forms, Reports, ...).

En un entorno de base de datos los programadores pueden construir bloques PL/SQL para utilizarlos como procedimientos o funciones, o bien pueden escribir estos bloques como parte de scripts SQL\*Plus.

Los programas o paquetes de PL/SQL se pueden almacenar en la base de datos como otro objeto, y todos los usuarios que estén autorizados tienen acceso a estos paquetes. Los programas se ejecutan en el servidor para ahorrar recursos a los clientes.

Bloque PL/SQL

Bloque es la unidad de estructura básica en los programas PL/SQL. Supone una mejora en el rendimiento, pues se envían los bloques completos al servidor para ser procesados en lugar de enviar cada secuencia SQL.

Partes de un bloque:

- Zona de declaraciones: zona opcional. Se declaran los objetos locales (variables, constantes...).
- Zona de instrucciones: zona obligatoria.
- Zona de tratamiento de excepciones: zona opcional. Se tratan excepciones en el programa.

Ejemplo de Bloque PL/SQL:

```
[ DECLARE | IS / AS ]
    <declaraciones>
BEGIN
    <instrucciones>
[ EXCEPTION ]
```

Francisco Miguel Pardo Veses

<tratamiento de excepciones>

END;

/

Tipos de bloques

- Anónimo (sin nombre)

Siempre comienza con **DECLARE** o directamente con **BEGIN**.

- Subprogramas (tienen nombre) A su vez, se divide en :

- Procedimientos en PLSQL

Los procedimientos tienen la utilidad de fomentar la re-utilización de programas que se usan comúnmente. Una vez compilado, queda almacenado en la base de datos (por eso es también llamado 'Procedimiento almacenado') y puede ser utilizado por múltiples aplicaciones.

- Funciones en PLSQL

Una función es un bloque de código PL/SQL que tiene las mismas características que un procedimiento almacenado. La diferencia estriba en que una función devuelve un valor al retornar. Al devolver un valor puede ser llamada como parte de una expresión.

- Triggers

Un trigger o disparador se ejecuta ante un determinado evento de manera automática. Generalmente se utilizan para garantizar que una determinada acción siempre se realiza después de realizar una tarea determinada. Se debe tener cuidado con este tipo de estructuras puesto que un uso excesivo puede dar lugar a dependencias difíciles de mantener. Además se deben tener muy claras las restricciones de integridad para evitar problemas.

## 9.Herramientas utilizadas

### 9.1.Samsung Galaxy Tab

Este tablet posee un procesador 1GHz Cortex A8 con PowerVR SGX540 y una plataforma Android 2.2 aunque es posible actualizarlo a la versión Android 2.3.3 Gingerbread Posee una memoria RAM de 512MB, una memoria interna de 16GB y ranura para MicroSD de hasta 32GB

La tablet dispone de una pantalla multitouch LCD-TFT de 7 pulgadas con resolución WSVGA (1024 x 600), la calidad de la imagen es muy buena. Dispone de dos cámaras, una detrás de 3MPX con flash de LED y la delantera de 1.3MPX para poder realizar video llamadas. Dispone también de GPS integrado.

Sus dimensiones son de 190 x 120 x 12 mm y un peso de 380 gramos, no es muy pesado, aunque se hecha en falta que sea algo mas fino.

Gracias a su cámara trasera de 3MPX podemos realizar grabaciones de video en resoluciones de 720X480 y permite la reproducción de video Full HD (1080p) .

La duración de su batería es también un punto fuerte ya que permite hasta 7 horas de autonomía reproduciendo videos sin parar.

En lo que se refiere a la conectividad, se dispone de tecnología 3G, que junto con la tecnología HSDPA (High Speed Downlink Packet Access) mejoran notablemente la velocidad de transmisión de la red, y WiFi 802.11b/g para conectarse a cualquier red inalámbrica que esté al alcance. También tiene Bluetooth® 3.0 ,A-GPS y A2DP para

auriculares inalámbricos estéreo.

En el apartado de Redes tanto 2G como 3G, cuenta con las siguientes:

- 2G: GSM/GPRS/EDGE : 850 / 900 / 1800 / 1900 MHz
- 3G: HSUPA/HSDPA: 900 / 1900 / 2100 Mhz

En la pantalla principal tenemos un área de notificaciones, donde nos aparecerán en nuestro móvil y verlos en la mejor calidad posible.

Aparte de todas estas novedades, el dispositivo incorpora los servicios típicos que ofrece todo teléfono móvil, como agenda, alarma, calendario, calculadora, cámara de fotos y de vídeo, mensajería SMS/MMS y gestión de imágenes y música que podemos descargarnos en nuestra tarjeta microSD.

Además, hay otro servicio importante que es Android Market, donde podemos encontrar todas las aplicaciones que los usuarios han creado y han decidido compartir para que todo el mundo pueda usarlas, aprovechando la definición de software libre y haciendo aún más grande la revolución que ya está causando Android. Entre las aplicaciones, podemos encontrar muchísimas gratuitas, como un scanner de códigos de barras, servicio móvil de eBay, servicios de comunicación como Skype, Live Chat, Hi MSN, herramientas de finanzas, bloc de notas, aplicaciones multimedia, noticias y meteorología, viajes, bibliotecas de software, multitud de juegos y cualquier aplicación que nos pueda pasar por nuestra imaginación y queramos subirla a Android Market para que todos los usuarios puedan disfrutar de ella.

## **9.2. Entorno de desarrollo Eclipse Helios**

Eclipse es un entorno de desarrollo integrado de código abierto para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

## 10. Conclusiones y agradecimientos

Como conclusiones finales, cabe decir que ha sido un proyecto interesante, aunque al principio surgieron varios problemas en el tema de como comunicar la aplicación debido a la poca experiencia en este tipo de aplicaciones. Aunque después de esta aplicación en posteriores actualizaciones este proyecto servirá de gran utilidad debido a que se han establecido unas pautas de comunicación con la base de datos y se podrán realizar aplicaciones mas complejas. En otras palabras este proyecto es un acercamiento a las nuevas tecnologías móviles.

En lo referente a los agradecimientos, tengo que agradecer a mi familia el apoyo durante estos años de estudios ya que sin su apoyo no lo hubiera logrado, también quiero agradecer a mi novia que durante estos años me ha estado dando ánimos para no decaer y terminar mis estudios y también durante la realización de este PFC. También cabe destacar la ayuda en la realización de este proyecto de Juan Carlos Morales (Director UPV Radiotelevisión ), sin el cual no habría podido realizarse la comunicación de la base de datos con la aplicación con la realización de los procedimientos almacenados que utiliza la aplicación. También tengo que destacar la ayuda de Ramón Molla Vaya(Director del PFC), que me proporciono la herramienta donde probar y desarrollar la aplicación (Samsung Galaxy Tab) y superviso la realización de la documentación y el proceso de desarrollo del código de la aplicación. Un agradecimiento especial a Ivan Gallego por realizar los iconos de la aplicación y su opinión en lo referente al apartado visual de la aplicación.

## Apéndice A:

### Manual de instalación del entorno de desarrollo

Primeramente instalaremos el SDK de Android para su posterior inserción en nuestro entorno de desarrollo.

Para ello, lo descargamos de la pagina oficial de programadores de Android, que esta a continuación:

<http://developer.android.com/sdk/index.html>

En esta pagina disponemos de dependiendo del S.O que se utilice distintas descargas, en este proyecto he utilizado el de Windows

**Download the Android SDK**

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the S

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD M* starter package. See [Adding SDK Components](#).

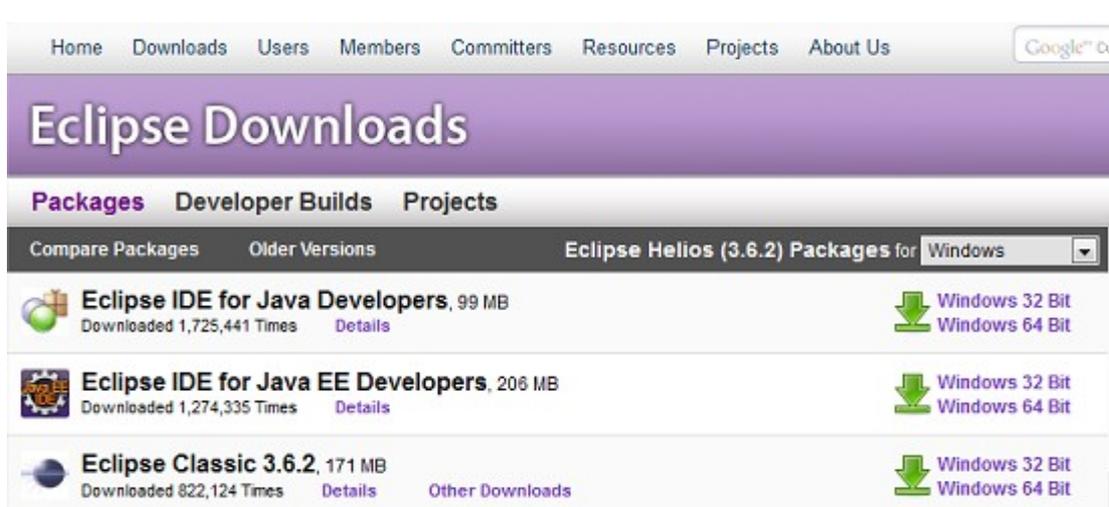
Platform	Package	Size	MD5 Checksum
Windows	<a href="#">android-sdk_r11-windows.zip</a>	32837554 bytes	0a2c52b8f8d97a4871ce8b3eb38e3072
	<a href="#">installer_r11-windows.exe</a> (Recommended)	32883649 bytes	3dc8a29ae5afed97b40910ef153caa2b
Mac OS X (intel)	<a href="#">android-sdk_r11-mac_x86.zip</a>	28844968 bytes	85bed5ed25aea51f6a447a674d637d1e
Linux (i386)	<a href="#">android-sdk_r11-linux_x86.tgz</a>	26984929 bytes	026c67f82627a3a70efb197ca3360d0a

Una vez descargado( podemos elegir si instalamos o descargamos el zip), vamos a donde lo hemos descargado y ejecutamos el SDK Manager), el cual nos dirá que tenemos que instalar bastantes actualizaciones de librerías y diversas cosas necesarias. Este proceso es lento y tarda su tiempo, no os desesperéis.

Una vez realizado esto ya podéis pasar a configurar el entorno de desarrollo(Eclipse)

Francisco Miguel Pardo Veses

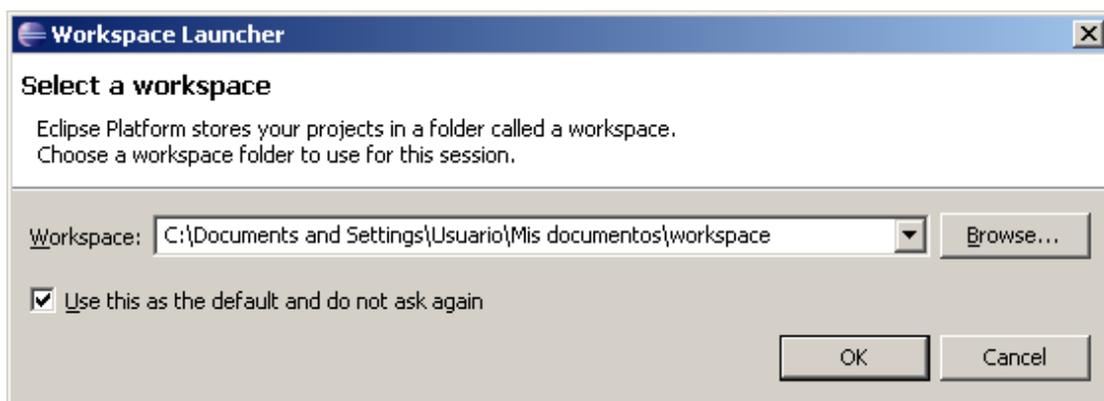
Para poder empezar a desarrollar una aplicación para Android, necesitaremos el entorno de desarrollo Eclipse. En primer lugar descargaremos el entorno de desarrollo de la pagina oficial de eclipse (<http://www.eclipse.org/downloads/>)



Personalmente utilizo Eclipse IDE for Java Developers(Galileo), ya que es una versión muy completa y suficiente para nuestro cometido.

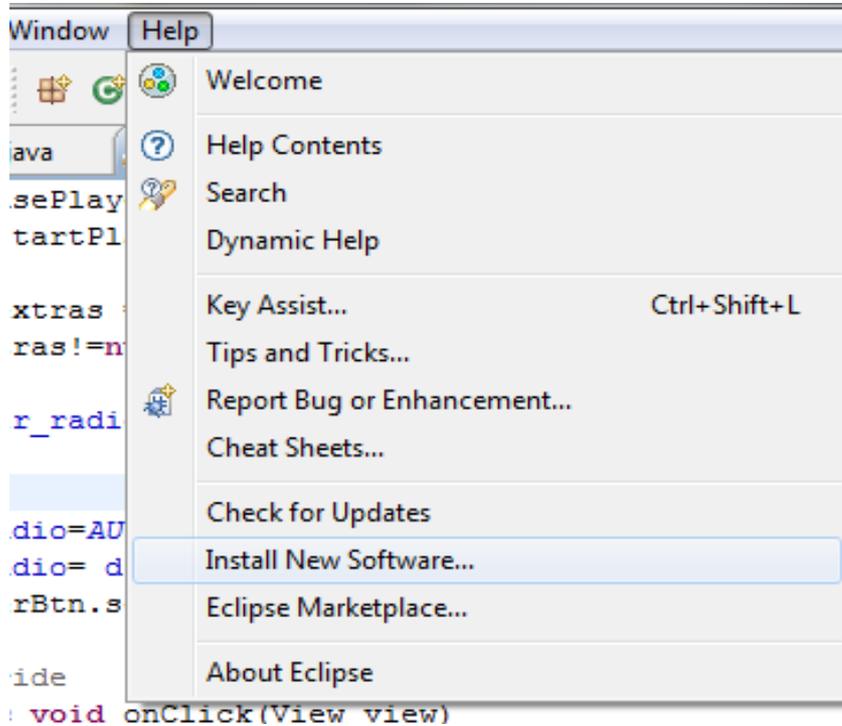
Una vez descargado( en versión de 32 o 64 Bits) lo descomprimimos en el disco duro y ya esta listo para utilizar, no necesita instalación.

Entra en el directorio *eclipse* y ejecuta *eclipse.exe* para lanzar la aplicación. La primera vez que arranques, te pedirá que le indiques un directorio donde colocar tu espacio de trabajo (workspace). Selecciona la ruta que quieras, que es donde se guardarán tus proyectos. Marca la opción “Use this as the default...” para que no te vuelva a preguntar más y pulsa OK.



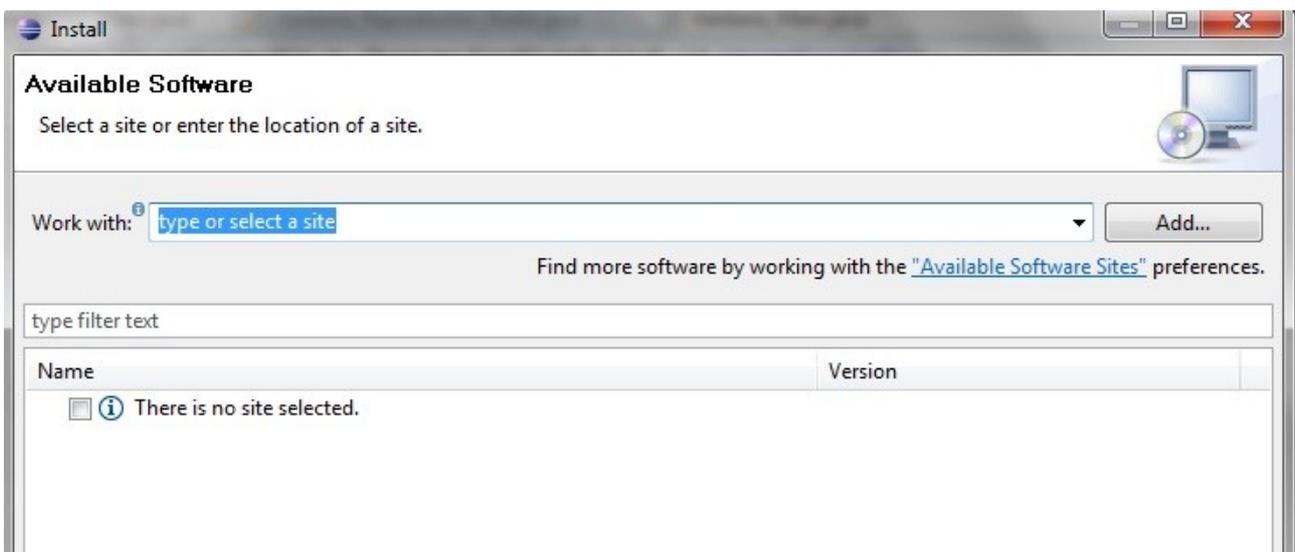
**Instalación de las Android Development Tools (ADT)**

Dentro de Eclipse, selecciona Install New Software, en el menú Help



Dentro de Install New Software:

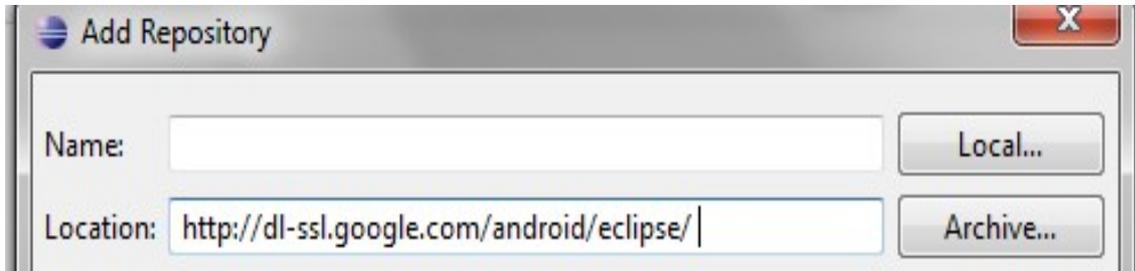
- En la parte derecha, haz click en Add.



Francisco Miguel Pardo Veses

Nos aparecerá una ventana en donde pondremos :

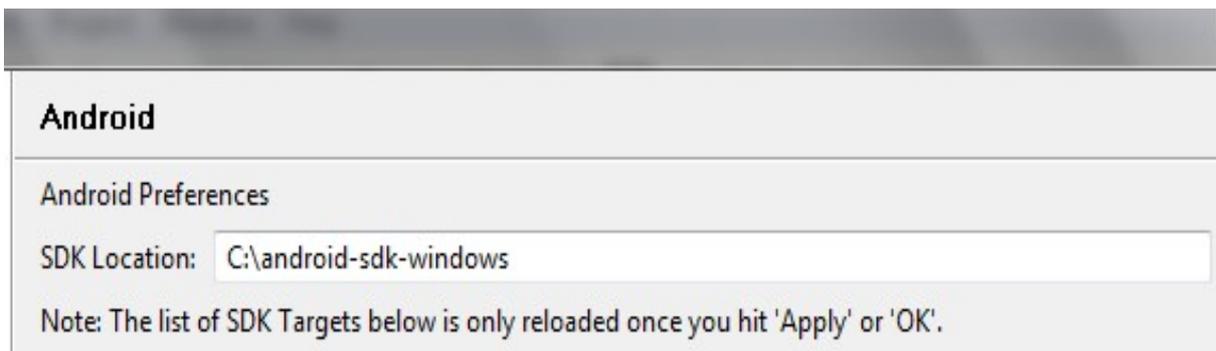
“http://dl-ssl.google.com/android/eclipse/ ”en el apartado de “Location”



Una vez hecho esto, instalaremos **Android DDMS** y **Android Development Tools**



Acabada la instalación, reinicia Eclipse. Una vez reiniciado, debemos configurar Eclipse para que acceda al SDK de Android ya instalado. Abre el menú Window y selecciona Preferences. En el apartado Android, en SDK Location debemos introducir la ruta donde hemos descomprimido el SDK de Android..



Presiona Ok para cerrar la ventana de preferencias. Ya tienes el SDK de Android integrado en Eclipse para empezar a programar.

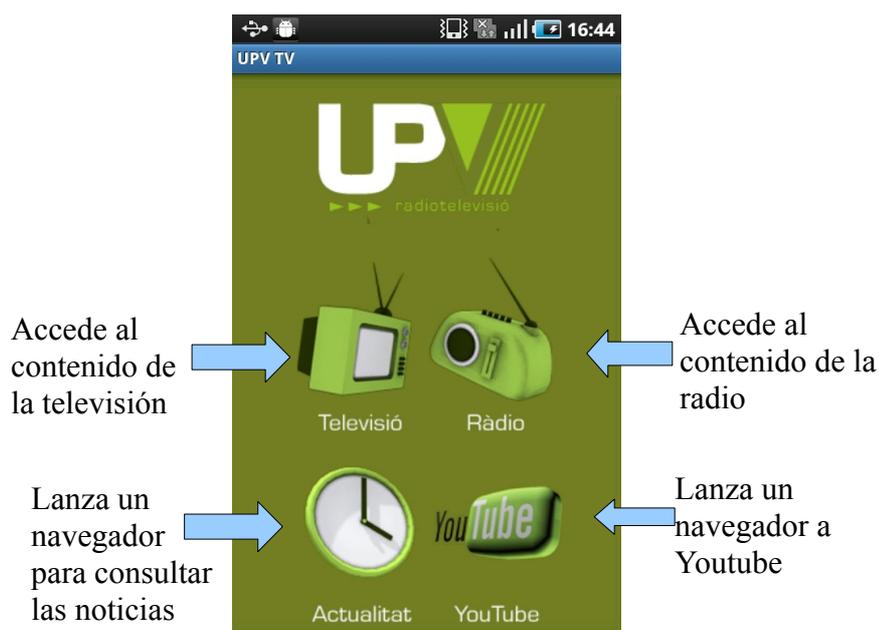
## Apéndice B:

### Manual de usuario de la aplicación

En este apéndice realizaremos un breve manual para todos aquellos usuarios de la aplicación, al tratarse de una aplicación que visualiza videos y programas de radio no conlleva excesiva dificultad, pero aun así lo comentaremos breve mente.

Primero deberemos descargarnos la aplicación del Android Market, que instalara de forma automática en el móvil.

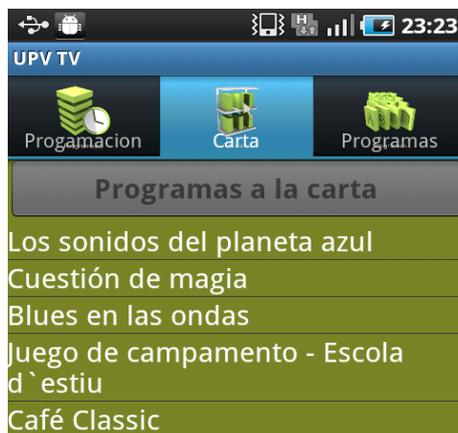
Una vez iniciamos la aplicación, nos encontramos con este menú principal. El procedimiento es el mismo para televisión que para radio, así que lo realizaremos para radio y luego en televisión es seguir los mismo pasos.



Cuando accedemos a televisión o la radio, obtenemos una estructura en 3 pestañas que distribuye la información muy claramente.



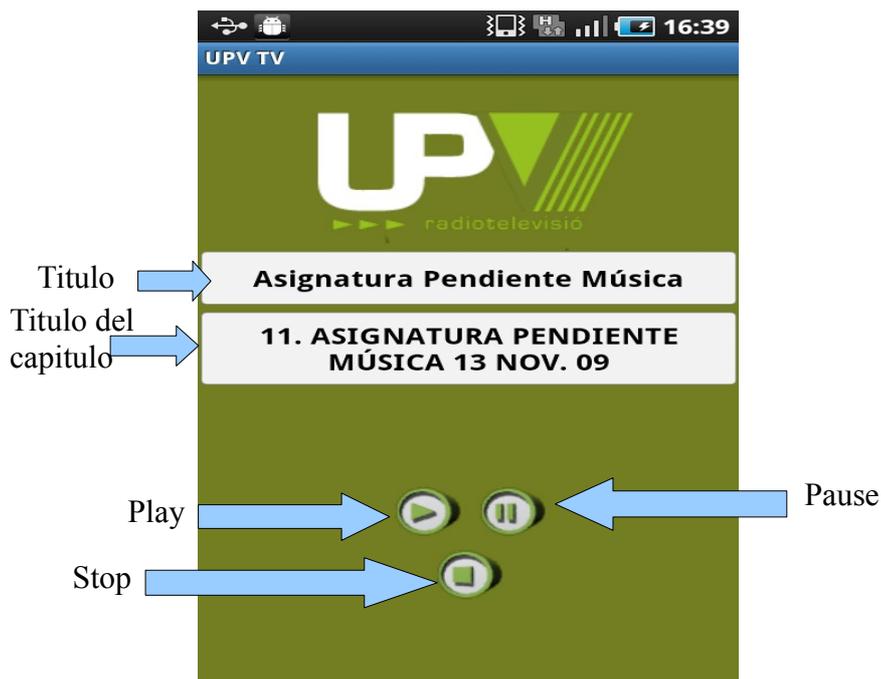
La primera pestaña que corresponde a la Programación, es tan solo meramente informativa de la programación que podemos disfrutar durante todo el día, solamente muestra la del día actual. En esta pantalla la información se obtiene consultando un XML que estructura la programación y que es parseado por la aplicación y recoge los datos necesarios y los muestra en una lista.



Las otras dos, que son “Carta” y “Programas”, nos muestra una lista de los programa que al seleccionar uno de ellos, nos traslada a una nueva pantalla en donde elegiremos un capitulo de dicho programa. Ambas pantallas consiguen su información accediendo a unos XML formados por los procedimientos almacenados correspondientes que tiene la base de datos, los cuales son parseados y recogidos los datos y mostrándolos en unas listas.



Una vez elegido el capítulo, el cual sigue el mismo patrón que las demás pantallas, nos lanza otra pantalla en donde veremos el reproductor de radio, que nos muestra el título del programa, el título del capítulo y los botones de reproducción.



En esta pantalla, reproduce en Streaming el archivo seleccionado, en este ejemplo es un programa de radio, pero en el caso de un video, se empieza a reproducir automáticamente, mientras que en los capítulos de radio, debemos apretar el botón “Play” para comenzar su reproducción.

## Apéndice C:

### Subida aplicación a Android Market

En este apéndice, explicaremos como subir una aplicación al Android Market, primeramente se necesita una cuenta en [Google Checkout](#) y pagar \$25 para toda la vida, frente a los \$100 anuales que se tiene que pagar en Apple.

El formulario de subida de app consta de tres partes:

- Assets
- Listing details
- Publishing options

#### Assets

En esta parte subirás el .apk, las imágenes de promoción, un icono enorme grande, un gráfico promocional, y si lo deseas, un video de Youtube. Es todo muy fácil y sencillo de rellenar.

ANDROID market

pacosal@gmail.com | Home | Help | Android.com | Sign out

### Upload an Application

#### Upload assets

Draft application .apk file click the 'publish' button to publish draft apk file	Upload an .apk file: <input type="text"/>	Examine...	Upload
Screenshots at least 2	Add a screenshot: <input type="text"/>	Examine...	Upload
High Resolution Application Icon <a href="#">[Learn More]</a>	Add a hi-res application icon: <input type="text"/>	Examine...	Upload
Promotional Graphic optional	Add a promotional graphic: <input type="text"/>	Examine...	Upload
Feature Graphic optional	Add a feature graphic: <input type="text"/>	Examine...	Upload
Promotional Video optional	Add a promotional video link: <input type="text" value="http://"/>		Upload
Marketing Opt-Out	<input type="checkbox"/> Do not promote my application except in Android Market and in any Google-owned online or mobile properties. I understand that any changes to this preference may take sixty days to take effect.		

**Screenshots:**  
320 x 480, 480 x 800, or 480 x 854  
24 bit PNG or JPEG (no alpha)  
Full bleed, no border in art  
You may upload screenshots in landscape orientation. The thumbnails will appear to be cropped, but the actual images will be preserved.

**High Resolution Application Icon:**  
512w x 512h  
32 bit PNG or JPEG  
Maximum: 1024 KB

**Promo Graphic:**  
180w x 120h  
24 bit PNG or JPEG (no alpha)  
Full bleed, no border in art

**Feature Graphic:**  
1024w x 500h  
24 bit PNG or JPEG (no alpha)  
Will be downsized to mini or micro

**Promotional Video:**  
Enter YouTube URL.

El siguiente paso después de rellenar el primer formulario, toca rellenar los detalles de la aplicación.

## Listing details

Aquí va la descripción de la aplicación en diferentes idiomas.

**Listing details**

Language | \*English (en) |  
[add language](#) | Star sign (\*) indicates the default language.

Title (en)   
0 characters (30 max)

Description (en)   
0 characters (4000 max)

Recent Changes (en)   
[\[Learn More\]](#)  
0 characters (500 max)

Promo Text (en)   
0 characters (80 max)

Application Type

Category

Price  Free  EUR   
Selling the price to Free is permanent; you cannot change to a price later. [\[Learn More\]](#)

## Publishing options

Esta es la parte que incorporó hace poco Google donde se clasifica la app por edades y por supuesto, se proporciona la información de contacto y aceptaciones correspondientes.

**Publishing options**

Copy Protection  Off (Application can be copied from the device)  
 On (Helps prevent copying of this application from the device. Increases the amount of memory on the phone required to install the application.)  
The copy protection feature will be deprecated soon, please use [licensing.santoc](#) instead.

Content Rating [\[Learn More\]](#)  
 Mature  
 Teen  
 Pre-Teen  
 All

Locations Select locations to list in:  
 All locations  
(Includes more countries than those listed below. As the developer, you are responsible for complying with country-specific laws related to the distribution or sale of your application into that country, including your home country.)

**Contact information**

Website   
Email   
Phone

**Consent**

This application meets [Android Content Guidelines](#)

I acknowledge that my software application may be subject to United States export laws, regardless of my location or nationality. I agree that I have complied with all such laws, including any requirements for software with encryption functions. I hereby certify that my application is authorized for export from the United States under these laws. [\[Learn More\]](#)