

TFC - Sistemas Operativos y Arquitectura de Computadores

Memoria

**Desarrollo de aplicación móvil utilizando
conceptos de realidad aumentada (AR)**

Autor:

José Antonio Domínguez Medina
jdominguezmed@uoc.edu
UOC-ETIG
(Segundo semestre del 2011)

Consultor:

[Francesc Guim Bernat](#)

Dedicatoria

A mi princesa Ana,
sin ella ando sin rumbo y todo carece de sentido.

1 Índice

<u>1</u>	<u>ÍNDICE</u>	<u>3</u>
<u>2</u>	<u>INTRODUCCIÓN</u>	<u>7</u>
<u>2.1</u>	<u>MOTIVACIÓN</u>	<u>7</u>
<u>2.2</u>	<u>OBJETIVOS</u>	<u>8</u>
<u>2.3</u>	<u>RESULTADO</u>	<u>8</u>
<u>2.4</u>	<u>PLANIFICACIÓN</u>	<u>11</u>
<u>2.5</u>	<u>ESTIMACIÓN DE COSTES</u>	<u>12</u>
<u>3</u>	<u>LA PLATAFORMA ANDROID</u>	<u>13</u>
<u>3.1</u>	<u>INTRODUCCIÓN</u>	<u>13</u>
<u>3.2</u>	<u>ARQUITECTURA</u>	<u>13</u>
<u>3.3</u>	<u>HERRAMIENTAS DE DESARROLLO</u>	<u>17</u>
<u>3.4</u>	<u>DISPOSITIVOS MÓVILES CON ANDROID</u>	<u>18</u>
<u>3.4.1</u>	<u>SMARTPHONES</u>	<u>19</u>
<u>3.4.2</u>	<u>TABLETS</u>	<u>20</u>
<u>4</u>	<u>CONSTRUCCIÓN DE PROYECTO</u>	<u>22</u>
<u>4.1</u>	<u>ANÁLISIS DE REQUERIMIENTOS/FUNCIONALIDADES</u>	<u>22</u>
<u>4.1.1</u>	<u>ESPECIFICACIÓN INICIAL DE REQUERIMIENTOS FUNCIONALES</u>	<u>22</u>

<u>4.1.1.1</u>	<u>ESCENARIO 1: GESTIÓN DE NOTAS</u>	<u>23</u>
<u>4.1.1.2</u>	<u>ESCENARIO 2: BUSCAR NOTAS</u>	<u>24</u>
<u>4.1.1.3</u>	<u>ESCENARIO 3: VISUALIZAR NOTAS</u>	<u>24</u>
<u>4.1.2</u>	<u>CASOS DE USO</u>	<u>25</u>
<u>4.1.2.1</u>	<u>CASO DE USO DE ESCENARIO 1: GESTIÓN DE NOTAS</u>	<u>25</u>
<u>4.1.2.1.1</u>	<u>FUNCIONALIDADES</u>	<u>26</u>
<u>4.1.2.2</u>	<u>CASO DE USO DE ESCENARIO 2: BUSCAR NOTAS</u>	<u>27</u>
<u>4.1.2.2.1</u>	<u>FUNCIONALIDADES</u>	<u>27</u>
<u>4.1.2.3</u>	<u>CASO DE USO DE ESCENARIO 3: VISUALIZAR NOTAS</u>	<u>28</u>
<u>4.1.2.3.1</u>	<u>FUNCIONALIDADES</u>	<u>28</u>
<u>4.1.3</u>	<u>ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES</u>	<u>29</u>
<u>4.1.3.1</u>	<u>REQUERIMIENTOS GENERALES DE USUARIO</u>	<u>29</u>
<u>4.1.3.2</u>	<u>ARQUITECTURA DEL SISTEMA</u>	<u>29</u>
<u>4.1.3.3</u>	<u>OPERATIVA DEL SISTEMA</u>	<u>30</u>
<u>4.1.3.4</u>	<u>CONSIDERACIONES DEL SISTEMA</u>	<u>31</u>
<u>4.2</u>	<u>DISEÑO</u>	<u>31</u>
<u>4.2.1</u>	<u>REALIZACIONES DE CASOS DE USO</u>	<u>32</u>
<u>4.2.1.1</u>	<u>REALIZACIÓN DE CASOS DE USO DEL ESCENARIO 1</u>	<u>32</u>
<u>4.2.1.1.1</u>	<u>DIAGRAMA DE SECUENCIA DE CREARNOTA</u>	<u>32</u>
<u>4.2.1.1.2</u>	<u>DIAGRAMA DE SECUENCIA DE CREARNOTA DESDE VISOR ARVIEWER</u>	<u>33</u>

<u>4.2.1.2</u>	<u>REALIZACIÓN DE CASOS DE USO DEL ESCENARIO 2</u>	<u>34</u>
<u>4.2.1.2.1</u>	<u>DIAGRAMA DE SECUENCIA DE REPRESENTARENMAPA</u>	<u>34</u>
<u>4.2.1.3</u>	<u>REALIZACIÓN DE CASOS DE USO DEL ESCENARIO 3</u>	<u>35</u>
<u>4.2.1.3.1</u>	<u>DIAGRAMA DE SECUENCIA DE PRESENTARVISOR</u>	<u>35</u>
<u>4.2.2</u>	<u>DISEÑO DE PANTALLAS</u>	<u>36</u>
<u>4.2.2.1</u>	<u>PANTALLA DE ACCESO</u>	<u>36</u>
<u>4.2.2.1.1</u>	<u>DESCRIPCIÓN DE COMPONENTES</u>	<u>37</u>
<u>4.2.2.1.2</u>	<u>DESCRIPCIÓN FUNCIONAL</u>	<u>37</u>
<u>4.2.2.2</u>	<u>PANTALLA DE MENÚ PRINCIPAL</u>	<u>38</u>
<u>4.2.2.2.1</u>	<u>DESCRIPCIÓN DE COMPONENTES</u>	<u>38</u>
<u>4.2.2.2.2</u>	<u>DESCRIPCIÓN FUNCIONAL</u>	<u>39</u>
<u>4.2.2.3</u>	<u>PANTALLA DE GESTIÓN DE NOTAS</u>	<u>40</u>
<u>4.2.2.3.1</u>	<u>DESCRIPCIÓN DE COMPONENTES</u>	<u>40</u>
<u>4.2.2.3.2</u>	<u>DESCRIPCIÓN FUNCIONAL</u>	<u>42</u>
<u>4.2.2.4</u>	<u>PANTALLA DE LOCALIZAR NOTAS</u>	<u>43</u>
<u>4.2.2.4.1</u>	<u>DESCRIPCIÓN DE COMPONENTES</u>	<u>43</u>
<u>4.2.2.4.2</u>	<u>DESCRIPCIÓN FUNCIONAL</u>	<u>43</u>
<u>4.2.2.5</u>	<u>PANTALLA DE VISUALIZAR NOTAS</u>	<u>44</u>
<u>4.2.2.5.1</u>	<u>DESCRIPCIÓN DE COMPONENTES</u>	<u>44</u>
<u>4.2.2.5.2</u>	<u>DESCRIPCIÓN FUNCIONAL</u>	<u>45</u>

<u>4.2.2.6</u>	<u>PANTALLA DE MARCAR NOTAS</u>	<u>45</u>
<u>4.2.2.6.1</u>	<u>DESCRIPCIÓN DE COMPONENTES</u>	<u>46</u>
<u>4.2.2.6.2</u>	<u>DESCRIPCIÓN FUNCIONAL</u>	<u>46</u>
<u>4.2.3</u>	<u>MODELO DE BASE DE DATOS</u>	<u>46</u>
<u>4.2.3.1</u>	<u>ENTIDAD USUARIOS</u>	<u>47</u>
<u>4.2.3.2</u>	<u>ENTIDAD NOTAS</u>	<u>48</u>
<u>4.2.3.3</u>	<u>ENTIDAD USUARIOS NOTAS</u>	<u>48</u>
<u>4.2.4</u>	<u>DESARROLLO DE SERVIDOR</u>	<u>48</u>
<u>4.2.4.1</u>	<u>DESCRIPCIÓN FUNCIONAL</u>	<u>49</u>
<u>5</u>	<u>CONCLUSIONES</u>	<u>50</u>
<u>6</u>	<u>GLOSARIO</u>	<u>51</u>
<u>7</u>	<u>BIBLIOGRAFÍA</u>	<u>52</u>

2 Introducció

2.1 Motivació

En mi entorno laboral llevo algunos años manteniendo/desarrollando evolutivos para un aplicativo desarrollado en J2ME CLDC 1.1, MIDP 2.0 y sistema operativo Symbian. Desde hace un tiempo he observado el mercado de la telefonía móvil y he podido ser partícipe de la emergente diversidad de aplicaciones y sistemas operativos de diferentes fabricantes. Pero ha sido uno de ellos, posiblemente ligado al atractivo de basarse en una distribución Linux –con el irremediable viso de rebeldía que estos sistemas representan frente a los productos de pago y a costosas licencias- y su simpático logo, el que ha conseguido que le dedique un tiempo especial no relacionado con mi trabajo profesional. Fue con un terminal Android y, navegando por su interminable repositorio de aplicaciones mediante Market, que entre por primera vez en contacto con el concepto de Realidad Aumentada (a partir de ahora, RA) mediante el aplicativo Layar [Layar 1]. Este browser de RA permite descargar marcadores, o puntos de interés (POIs), que luego pueden ser visualizados/añadidos/superpuestos a imágenes reales captadas mediante la cámara de video/foto del dispositivo, siendo este el principio básico de la RA, es decir, añadir información sintética creada digitalmente a imágenes reales. Posteriormente ha hecho mucho uso de esta tecnología en varios escenarios de ocio de mi vida, como por ejemplo, cuando visito una ciudad y necesito encontrar un determinado servicio (tienda, restaurant, transporte, etc.) , siendo una herramienta que ofrece una ayuda novedosa, efectiva y divertida.

Ha sido esta tecnología por tanto la que me ha animado a realizar mi TFC sobre esta temática, estando toda la tecnología necesaria a nuestro alcance para realizar una pequeña aplicación (adaptada al timing del TFC y mi disponibilidad) que exponga los conceptos básicos de la AR.

2.2 Objetivos

El objetivo principal de este proyecto es el de obtener unos conocimientos detallados del sistema operativo Android y el de ejemplificarlos mediante un aplicativo que haga uso de las características más peculiares de los dispositivos móviles en los que se ejecuta. Para ello el proyecto se dividirá en dos fases, éstas son:

- Fase 1:

Estudio del sistema operativo Android así como de las características generales de los dispositivos móviles que lo incorporan.

- Fase 2:

Desarrollo de un aplicativo en Java para Android que ejemplifique las características más novedosas de los dispositivos móviles que lo incorporan. Para ello, se pretende desarrollar un aplicativo que utilice algunas tecnologías de Realidad Aumentada.

2.3 Resultado

Como resultado del desarrollo del proyecto obtendremos:

- a) Una memoria de todo el proceso del proyecto que deberá incluir:
 - i. Estudio detallado del sistema operativo Android.
 - ii. Estudio de las generalidades/peculiaridades de los dispositivos móviles que incorporan el OS Android.
 - iii. Descripción funcional/técnica y resultado de pruebas de la aplicación seleccionada para ejemplificar los puntos anteriores.

b) Un aplicativo que ejemplificará algunas de las funcionalidades que caracterizan a los dispositivos móviles que ejecutan Android. Dicho aplicativo hará uso de la tecnología de Realidad Aumentada para implementar un sistema de creación de mensajes/notas que podrán ser ubicadas en un espacio virtual tridimensional y que, para su consulta/lectura, será necesario que el usuario destinatario de dicha nota se encuentre físicamente en la posición geográfica en la que se ha creado/emitió la nota virtual. Para ello, la aplicación deberá implementar las siguientes funcionalidades básicas:

1. Dar de alta usuarios en el sistema.
2. Autenticar usuarios en el sistema.
3. Dar de alta notas en el sistema introduciendo el texto y obteniendo las coordenadas de localización de la nota mediante la posición geográfica actual del dispositivo.
4. Dar de alta notas en el sistema introduciendo el texto y seleccionando las coordenadas de localización de la nota mediante un mapa interactivo (permitirá dejar notas virtuales en cualquier parte del mundo).
5. Modificación de notas creadas por un usuario autenticado en el sistema y seleccionables de una lista de notas creadas.
6. Borrado de notas creadas por un usuario autenticado en el sistema y seleccionables de una lista de notas creadas.
7. Búsqueda de notas mediante browser de AR; se visualizarán las del destinatario autenticado en el sistema y cuando se encuentre en el área de visualización de la nota.
8. Selección de notas en el browser, su visualización y marcación como leída.

Nota: Todas las funcionalidades serán detalladas en el Análisis de Requerimientos y posterior Diseño Funcional incorporados a la memoria entregable.

El sistema de notas virtuales estará dividido en dos entornos que dará lugar a dos aplicaciones, estos son:

Entono Servidor:

Una aplicación del lado del Servidor (serán Servlets con acceso a base de datos), que permitirá gestionar las notas virtuales creadas por los usuarios. Por tanto, el Servidor dispondrá de una serie de servicios que accederán a un base de datos para registra y gestionar las notas. Este registro dispondrá de la información necesaria para ubicar una nota en un espacio tridimensional (geonodes), así como el contenido de la nota y con un usuario creador y destinatario de la misma. Es por esto que también deberá proporcionar los servicios requeridos para realizar alta y baja de usuarios. Por tanto, los servicios básicos que realizará serán:

- Alta, baja y modificación de registro de notas asociada a usuarios.
- Alta y baja de usuarios.

El servidor de aplicaciones a utilizar será Apache Tomcat [Apac 1] y los servicios se implementarán con tecnología servlet. La base de datos podrá ser una Oracle Express Edition [Orac 1].

Entorno dispositivo móvil:

Una aplicación en el dispositivo móvil que nos permita logarnos/autenticarnos en el sistema Servidor y poder posteriormente:

- Obtener una lista de todas la notas creadas por el usuario.
- Obtener una lista de todos los usuarios del sistema (en un principio no va haber relación de amigos).
- Modificar o eliminar notas que no hayan sido leídas.
- Crear una nueva nota para un usuario concreto, situándola en un punto geoespacial.

- Visualizar y seleccionar notas en browser AR que estén asociadas al usuario logado y que se encuentren en el campo de visión de éstas.

Nota: Todas las funcionalidades serán detalladas en el Análisis de Requerimientos y posterior Diseño Funcional incorporados a la memoria entregable.

El entorno o ecosistema de AR a utilizar será posiblemente ARviewer [Libr 1], pero debe estudiarse también la posibilidad de otros entornos [Will 1], como puede ser Mixare [Mixa 1].

2.4 Planificación

El siguiente diagrama Gantt expone una primera aproximación/estimación de la planificación del proyecto. El calendario del diagrama ha sido adaptado para que la jornada laboral sea de 4h, que es la media de horas que podré dedicar al TFC cada día de la semana, es decir, de lunes a viernes:

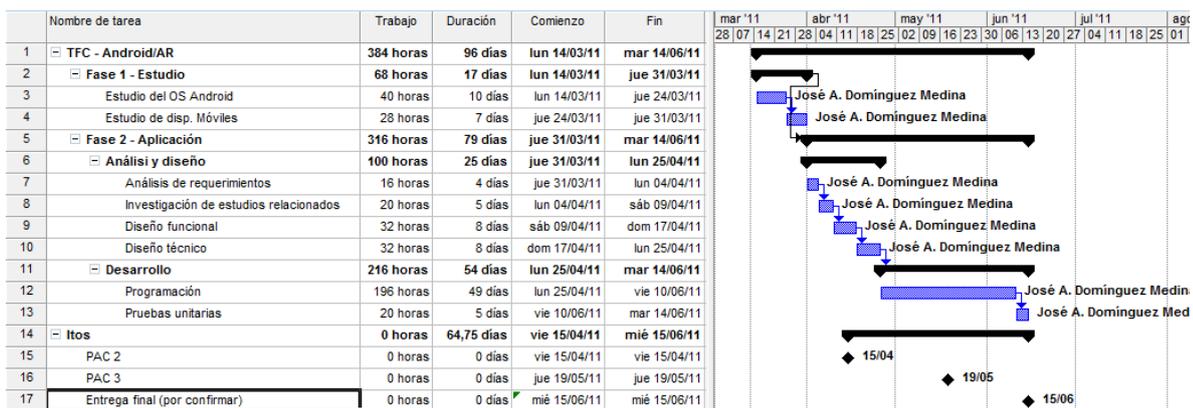


Fig 1. Cronograma Gantt del proyecto.

En la sección de Desarrollo no se ha especificado/desglosado las tareas de Programación, ésta será desarrollada y expuesta en un nuevo Gantt después de haber elaborado el Análisis de Requerimientos, obteniendo de esta forma una información más concreta de lo que se pretende implementar y, por tanto, de las tareas de desarrollo que se deberán realizar. En cualquier caso, sí que se ha estimado el número de horas que se debería emplear en su desarrollo, estimado a partir de las *a priori* funcionalidades básicas expuestas en los puntos anteriores. Después del Análisis de Requerimientos puede que estas funcionalidades se vean modificadas, pero en cualquier caso deberemos ajustarnos al timing que nos marca el TFC.

2.5 Estimación de costes

La estimación de costes asociado al proyecto se ha realizado teniendo en cuenta exclusivamente el factor humano, es decir, las horas de trabajo necesarias para la construcción del proyecto. A continuación se muestra una relación de los diferentes precios por hora dependiendo del perfil del trabajador:

Jefe de proyecto: 72 euros/hora

Analista sénior: 55 euros/hora

Analista programador: 40 euros/hora

Programador: 36 euros/hora

Por tanto, teniendo en cuenta el coste por hora de los perfiles expuesto y las horas incurridas por cada uno de ellos en la [planificación](#) del punto anterior, el coste total de proyecto será de 22.164 euros.

3 La plataforma Android

3.1 Introducción

Android es un sistema operativo (OS) o más extensamente considerado, como una plataforma emergente para dispositivos móviles que está obteniendo una gran aceptación. Su núcleo está basado en una versión modificada de Linux v2.6 [Lin 1] y su código fuente está disponible para la comunidad de desarrolladores para su libre customización. La empresa desarrolladora de esta plataforma es Google Inc. [Wiki 1], que con su proyecto Android Open Source Project (AOSP) [Goo 1] tiene como objetivo evolucionar este sistema proporcionando una plataforma de excelencia para el desarrollo de aplicaciones que mejore las experiencias de usuario de dispositivos móviles. Este proyecto está auspiciado por la Open Handset Alliance [OHA 1].

3.2 Arquitectura

La plataforma Android está constituida por una serie de capas software [Wiki 2] que cooperan entre sí para obtener una funcionalidad específica y dando lugar a lo que se denomina *a software stack* [PCMA 1]. Más específicamente, la plataforma Android está compuesta por un sistema operativo, software específico de conectividad entre módulos lógicos/software heterogéneos, denominado middleware [Wiki 3] y, una serie de programas clave o esenciales.

El siguiente diagrama de bloques muestra la jerarquía de cooperación de las diferentes capas que conforman el *software stack*, siendo las capas superiores dependientes de las inferiores.

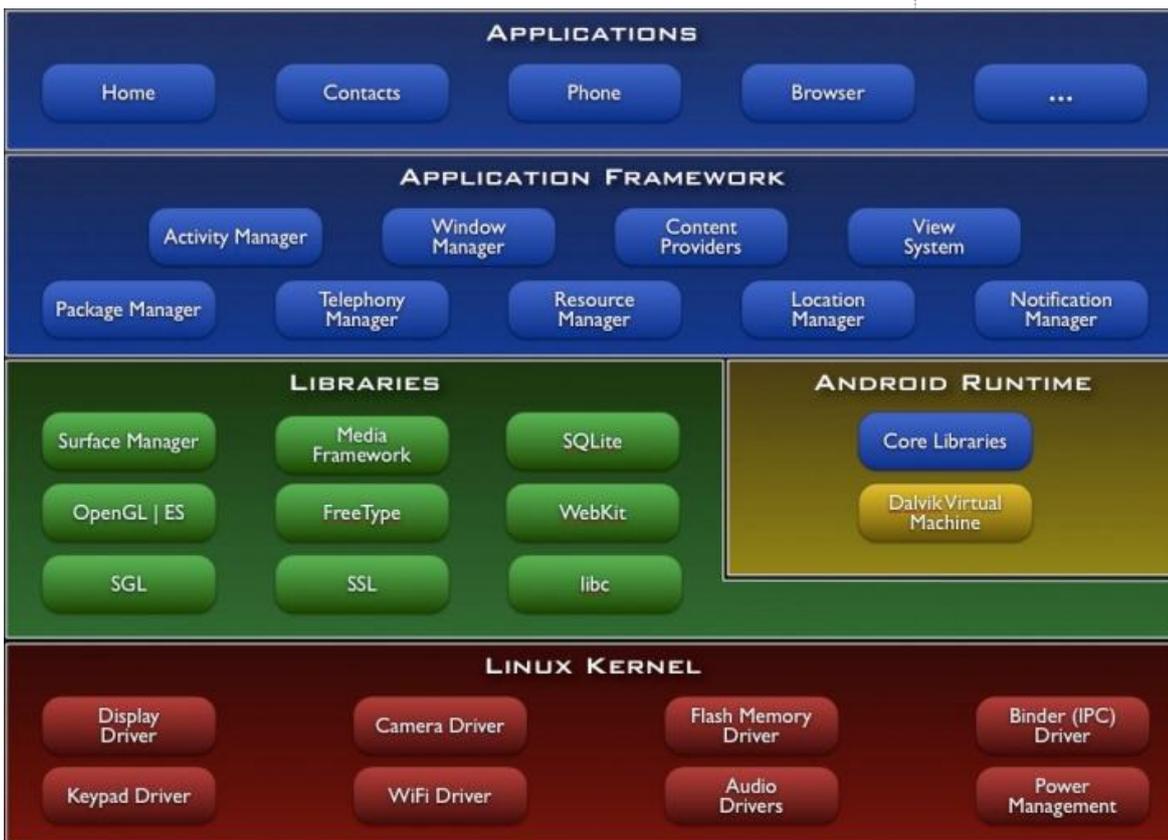


Fig 2. Arquitectura de la plataforma Android (obtenida de: <http://developer.android.com/guide/basics/what-is-android.html>).

Linux kernel

En la base de la jerarquía se encuentra el Linux Kernel, que es el conjunto de módulos encargados de interactuar de forma directa con el hardware del dispositivo y el de proporcionar los servicios más básicos o fundamentales del sistema, como puede ser la gestión de permisos de acceso a recursos, gestión de procesos [Wiki 4] y memoria disponible para llevar a cabo las tareas, así como el control del acceso a redes.

Android Runtime

La Runtime es el entorno o contenedor en dónde toda aplicación Android es ejecutada y le es permitido interactuar con el sistema/dispositivo. Este entorno o contenedor de ejecución está conformado por la máquina virtual Java [Wiki 5]

específica de Android, denominada Dalvik VM [Wiki 6], y el conjunto de librerías que forman el lenguaje básico/fundamental de computación Java, también denominado *Java core*. Todo programa/clase para poder ser ejecutado por Dalvik, necesita ser transformado a un código optimizado para una mejor gestión de la memoria; estos programas poseen la extensión *.dex* (Dalvik Executable). Cada aplicación Android es ejecutada en su propia máquina virtual, permitiendo el OS Android la ejecución concurrente de diversas VM.

Libraries

Conjunto de librerías específicas desarrolladas en C/C++ disponibles para los desarrolladores de aplicaciones a través del framework [Wiki 7] que dispone Android. Estas librerías ofrecen clases con funcionalidades muy diversas, como pueden ser: clases para la gestión gráfica en 2D (SGL) [Goo 2] y 3D (OpenGL) [Khro 1], clases para la gestión del gestor de base de datos SQLite [SQLi 1] o clases para gestionar los archivos media (audio, video e imagen) de formatos tales como MPEG4, H.264, MP3, AAC, AMR, JPG, y PNG; estas librerías están basadas en PacketVideo's OpenCORE (<http://www.pv.com/products/android/index.html>).

Application Framework

Es el conjunto de librerías de alto nivel que Android dispone a la comunidad de desarrolladores para implementar aplicaciones con un alto contenido visual e interactivo. Estas librerías utilizan el mismo conjunto de APIs de bajo nivel que las utilizadas por el núcleo del sistema operativo. La arquitectura de las aplicaciones en Android facilita que las funcionalidades que presentan puedan ser compartidas desde otras aplicaciones, mejorando de esta forma la reusabilidad de componentes [Wiki 8]. Esta arquitectura basada en componentes permite que cualquier componente pueda ser reemplazado por el usuario.

Los servicios y sistemas generalmente usados y que están disponibles en el Application Framework son:

- Windows manager: Proporciona los controles gráficos que serán utilizados en las aplicaciones para interactuar con el usuario. Estos controles pueden ser muy diversos como: botones, listas, cuadrículas (grids), navegador Web, etc.
- Content Providers: Es el mecanismo proporcionado para compartir datos/información entre diferentes aplicaciones. Cada tipo de dato es gestionado por un *provider*, existiendo para los tipos más comunes (audio, video, imagen, información personal de contacto, etc.).
- Resource Manager: Proporciona el acceso y uso de recursos externos a la aplicación. Estos recursos pueden ser imágenes, ficheros de diseño de pantallas (layouts), etc, que pueden ser organizados para su mejor selección y, por ejemplo, teniendo en cuenta el modelo en dónde se va a ejecutar el aplicativo.
- Notification Manager: Es el mecanismo proporcionado para display/mostrar mensajes de alerta en la barra de estado.
- Activity Manager: Encargado de gestionar el ciclo de vida de las aplicaciones mediante el control de las actividades (pantallas) que se van sucediendo en la aplicación y las acciones que va ejecutando el usuario sobre ellas. Este ciclo de vida se establece mediante eventos que se ejecutan, por ejemplo, al crearse la actividad, al parar la actividad o al eliminarla.
- Applications: Es el marco o nivel en dónde ejecutará la aplicación objetivo de este trabajo TFC. En este nivel encontramos, por tanto,

todas aquellas aplicaciones que son directamente invocadas por el usuario y dispuestas para su uso, como pueden ser: navegador Web (browser), cliente de correo, servicio de SMS, calendario, Maps, etc. Todas estas aplicaciones están desarrolladas en java.

3.3 Herramientas de desarrollo

Para el desarrollo de aplicaciones en Android, la plataforma dispone de una serie de herramienta y recomendaciones para facilitar las implementaciones. Estas herramientas serán las utilizadas para el desarrollo de la aplicación objetivo de este trabajo TFC. A continuación se comentan las diferentes herramientas, dispuestas en el orden que deberían instalarse en nuestro computador de desarrollo:

Java SE JDK (http://en.wikipedia.org/wiki/Java_Development_Kit): Es el conjunto de recursos Java, o plataforma, que conforman la denominada Java Standard Edition, fundamental para el desarrollo de aplicaciones Java. Este conjunto de recursos no es propio de la plataforma Android, pero debe ser instalada en el computador para poder desarrollar aplicaciones.

Eclipse IDE (<http://www.eclipse.org/downloads/>): Es el entorno de desarrollo recomendado por Android Developers para la implementación de aplicaciones. Existe una versión de Eclipse que ya viene preparada para facilitar el desarrollo de aplicaciones en lenguaje Java. Se recomienda la versión Eclipse 3.5 (Gallileo) o superior. También es posible obtener la versión clásica de este IDE a la cual se le debería añadir el módulo JDT (<http://www.eclipse.org/jdt/>) para obtener un entorno de trabajo orientado a Java.

Android SDK (<http://developer.android.com/sdk/index.html>): Es el conjunto de recursos fundamentales que dispone la plataforma Android para el desa-

rollo de aplicaciones Java. Posteriormente deberemos configurar el IDE Eclipse para que localice el SDK de Android y disponga de sus recursos en el entorno de desarrollo. Entre los recursos de que dispone el SDK de Android se encuentra un emulador de dispositivos, para probar los desarrollos sin

necesidad de realizar instalaciones sobre un dispositivo físico (<http://developer.android.com/guide/developing/tools/emulator.html>).

ADT Plugin For Eclipse (<http://developer.android.com/sdk/eclipse-adt.html#installing>): Plugin que extiende las funcionalidades de Eclipse para proporcionar un entorno de desarrollo orientado a Android, disponiendo de toda una serie de utilidades como: componentes basados en el UI de Android, herramientas de debugger del SDK de Android, generación de aplicación en formato .apk (http://en.wikipedia.org/wiki/APK_%28file_format%29), etc.

Crear un Android Virtual Device (<http://developer.android.com/guide/developing/devices/index.html>): Es la configuración de dispositivo que será utilizada por el emulador de Android. Mediante AVD podremos definir las características hardware y software del dispositivo que vamos a emular y para el cuál queremos ejecutar la aplicación en desarrollo.

3.4 Dispositivos móviles con Android

Actualmente existen dos tipos de dispositivos, en el ámbito doméstico, que disponen de la plataforma Android, estos son: los smartphones [Wiki 9] y las mediatables o simplemente, tablets [Wiki 10].

3.4.1 Smartphones

Son teléfonos móviles que han evolucionado en capacidades de computación, interacción con el usuario y conectividad, con capacidad de ejecutar aplicaciones informáticas de recursos reducidos, en cuanto a memoria, capacidad de cómputo y almacenamiento interno. Estas aplicaciones están gestionadas por un sistema operativo completo que las ejecuta y les proporciona diferentes servicios para la interacción con el

sistema smartphone. Los siguientes son algunos de los smartphones más representativos de la escena Android:

HTC

Desire

HD

(<http://www.htc.com/es/product/desirehd/specification.html>):

Es uno de los smartphones que posee una pantalla de gran tamaño, con una cámara de altas prestaciones de 8 megapíxels y que es capaz de grabar video en 720p HD. Las imágenes tomadas son tipificadas mediante Geotagging, que añade información de posicionamiento geográfico en los metadatos. La distribución de Android que incorpora es la 2.2 (Froyo). Posee una CPU de 1GHz, GPS y sensores *G-Sensor* [Wiki 11], *digital compass*, *proximity sensor* y *ambient light sensor*.

Es el dispositivo físico sobre el que realizaré las pruebas, una vez hayan sido validadas en el emulador.



Samsung

Galaxy

S

(<http://galaxys.samsungmobile.com/specification/spec.html>):

Este dispositivo posee una pantalla con unas dimensiones algo menores que la del HTC Desire HD, aunque de igual resolución, y utiliza tecnología Super AMOLED [Wiki 12] de muy bajo consumo y calidad de imagen superior. La resolución máxima de la cámara es algo menor, de 5 megapíxels, y las imágenes tomadas son tipificadas mediante Geotagging. La distribución de Android que incorpora es la 2.1 (Eclair) pero actualizable a 2.2 (Froyo). Posee una CPU de 1GHz, GPS y sensores *Accelerometer Sensor* [Wiki 11], *digital compass*, *proximity sensor*.



3.4.2 Tablets

El tablet es un dispositivo que tomó relevancia con la advenimiento del Apple iPad, anunciado el 27 de enero del 2010 (<http://en.wikipedia.org/wiki/IPad>). Como principal funcionalidad permite ejecutar aplicaciones, proporcionan conectividad a internet. Se caracteriza por tener un tamaño entre 6 y 10 pulgadas y una interfaz táctil (casi ningún tablet moderno trae puntero).

Estos dispositivos cubren las necesidades del segmento de mandos intermedios que necesitan un dispositivo ligero para ejecutar aplicaciones en un ambiente de trabajo en movilidad o, simplemente como dispositivo de ocio doméstico. Los siguientes son algunos de los tablets más representativos de la escena Android:

Samsung Galaxy Tab 10.1 (<http://galaxytab.samsungmobile.com/>):

Samsung ha aparecido en el mercado de las tablets en el 2010 como el principal competidor del iPad. Ha podido diseñar y lanzar un modelo representativo del sistema operativo Android que ya posee una versión ampliada en 10" y que prevé otros modelos a ser lanzados durante el 2011. La versión de 10" se anuncia con OS Android 3.0 (Honeycomb) customizado por Samsung. Posee una pantalla TFT LCD de 10,1 pulgadas, con una resolución de 1280x800 (WXGA) y un procesador de doble núcleo a 1 GHz.



Motorola Xoom (<http://www.motorola.com/Consumers/US-EN/Consumer-Product-and-Services/Tablets/ci.MOTOROLA-XOOM-US-EN.alt>):

Motorola, con el respaldo de Google, ha lanzado al mercado este tablet con el sistema operativo Android 3.0 (Honeycomb), customizado para su uso en Tablets. A esto se suma un hardware competente con un procesador de doble núcleo a 1 GHz y una apuesta por el mercado de uso personal, al igual que el iPad. Su resolución de 1280x800 en una pantalla de 10.1 pulgadas es una de las mejores, como la del modelo de Samsung. Posee sensores *sensor Proximity*, *sensor ambient light*, *barometer*, y *gyroscope*.



4 Construcción de proyecto

El resto de este trabajo de final de carrera se va a basar en la construcción de un aplicativo para dispositivos móviles para la plataforma Android. El ciclo de vida del proyecto estará basado en el clásico de cascada en el cuál, la conclusión de cada una de las etapas marcará el inicio de la siguiente. A grandes rasgos este ciclo de vida lo conforman dos grandes fases: la fase de análisis y la fase de desarrollo y pruebas. Iniciamos, sin más dilaciones, la construcción del proyecto.

4.1 Análisis de requerimientos/funcionalidades

4.1.1 Especificación inicial de requerimientos funcionales

Se requiere construir un aplicativo para dispositivo móvil y para la plataforma Android que disponga de las siguientes especificaciones:

Se desea disponer de una aplicación que permita a un usuario, previamente autenticado en el sistema, elaborar una nota o mensaje y asignárselo a otro usuario que esté registrado en el sistema. Por tanto, la aplicación debe permitir registrar usuarios mediante un nombre identificativo y una contraseña. Esta nota creada y, asignada a un usuario concreto, sólo debe poder ser visible mediante la cámara de video del dispositivo y cuando el usuario destinatario de la nota se encuentre en la proximidad de las coordenadas geoespaciales en dónde se ha dejado la nota. Una vez el usuario destinatario dispone de la nota visible en su visualizador (viewer), podrá seleccionarla para su consulta. Esta acción de consulta deberá marcar la nota como leída, por tanto, las notas deberán disponer de estado que permita a los usuarios saber si disponen de alguna nota por leer cuando éstas son visibles a través del visualizador. Para poder localizar las notas asignadas para un determinado usuario, se deberá poder disponer de una lista de notas y sus estados. Si el usuario seleccionase alguna de estas notas de la

lista, ésta deberá representarse en un mapa mediante las coordenadas geoespaciales que tiene asignada cada nota individual para su localización; esto permitirá al usuario desplazarse hasta el lugar para poder acceder a sus notas. Para poder crear y asignar una nota, es necesario que la aplicación disponga de otra lista con todos los usuarios registrados por el sistema y posibles destinatarios. También deberá ser posible eliminar o modificar una nota previamente creada por un usuario concreto por lo que también se hace necesario disponer de una lista de notas creadas para su selección y modificación o borrado. Para poder localizar una nota, una vez el usuario destinatario se encuentra en la zona, el dispositivo móvil deberá presentar un mapa, cuando éste se encuentre en una posición horizontal con respecto al suelo, y situar las notas circundantes cuando éstas se encuentren en un radio de visualización determinado y establecido en el visualizador.

4.1.1.1 Escenario 1: Gestión de notas

Un usuario, previamente autenticado en el sistema, dispone de la opción de gestión de notas. Seleccionando esta opción se le debe presentar unas nuevas opciones para crear una nueva nota o consultar, modificar y eliminar una existente. Para esto último es necesario que las notas creadas por el usuario estén disponibles en una lista para su selección. Las notas deben poder ser asignadas a un usuario registrado en el sistema, por lo que estos usuarios deben estar disponible en otra lista para su selección, además, la nota creada debe poseer información referente a la posición geoespacial en dónde se desea que quede ubicada para su posterior consulta mediante un visualizador. Estas coordenadas geoespaciales deberán poder ser asignadas de forma automática mediante la posición actual del dispositivo o podrán ser asignadas mediante la señalización de una posición en un mapa.

4.1.1.2 Escenario 2: Buscar notas

Un usuario, previamente autenticado en el sistema, dispone de la opción de buscar notas. Seleccionando esta opción se le debe presentar una lista de notas asignadas, leídas y no leídas, para su selección. Al seleccionar una nota de la lista, ésta debe ser representada en un mapa para identificar su localización geográfica.

4.1.1.3 Escenario 3: Visualizar notas

Un usuario, previamente autenticado en el sistema dispone de la opción de visualizar notas. Seleccionando esta opción se le debe presentar, si el dispositivo está en posición vertical, un visualizador de notas virtuales a través de la cámara de video. Si el usuario se encuentra en la cercanía de una nota, ésta debe hacerse visible en el visualizador si el usuario se encuentra orientado hacia ella y quedar seleccionable para su consulta. Si el dispositivo se encuentra en posición horizontal, se debe presentar un mapa con las notas circundantes situadas sobre él para su seguimiento/localización. Estas dos funcionalidades deben conmutar de forma automática si el dispositivo pasa de una a otra posición. Una vez se ha consultado una nota, ésta debe cambiar su estado a ‘leída’.

4.1.2 Casos de uso

A continuación se exponen los casos de usos de los escenarios descritos.

4.1.2.1 Caso de uso de Escenario 1: Gestión de notas

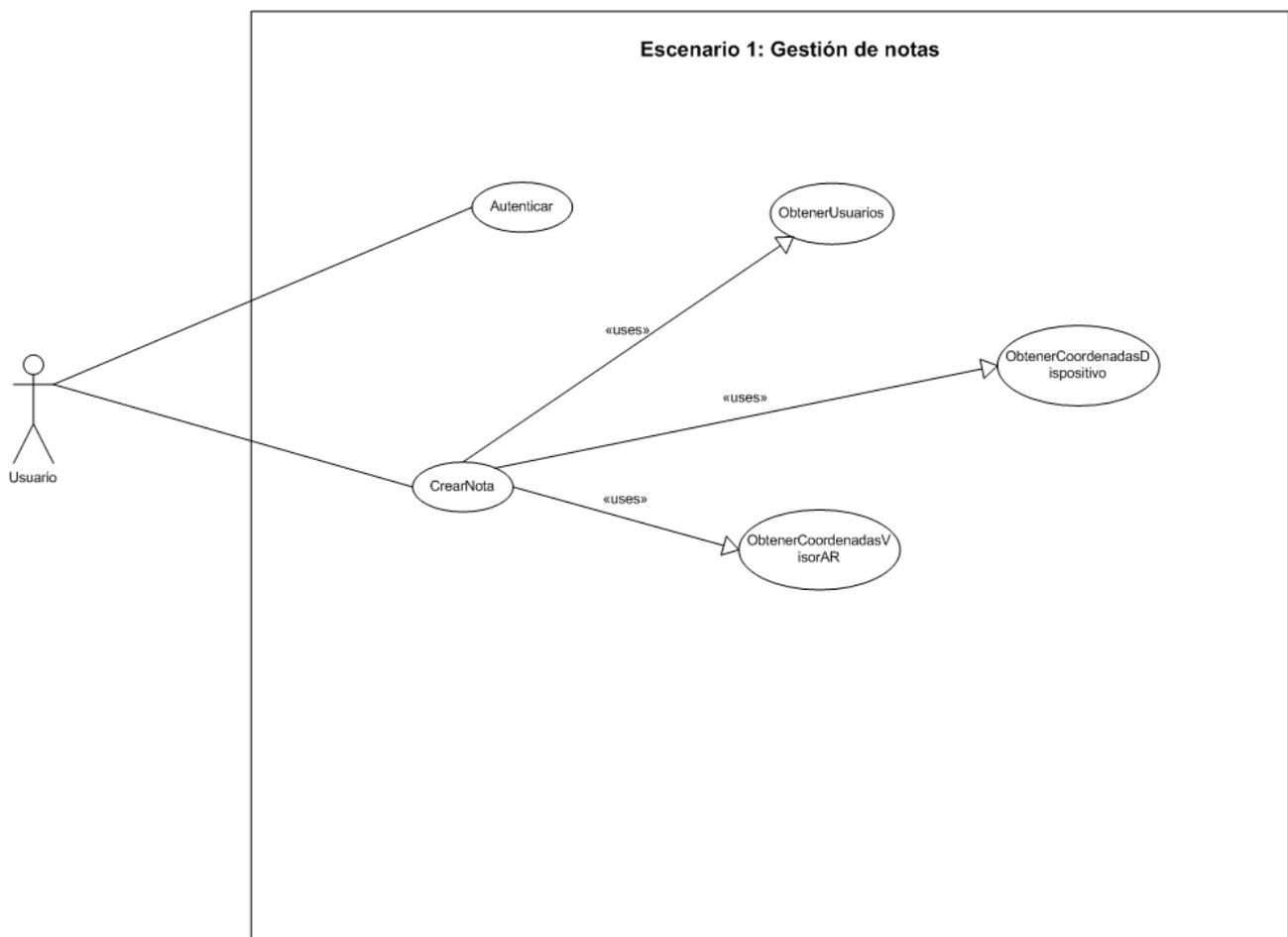


Fig 3. Caso de uso de Gestión de Notas.

4.1.2.1.1 Funcionalidades

- **Autenticar:** Permite el acceso de un usuario registrado en el sistema.
- **Crear nota:** Permite a un usuario crear una nueva nota y asignarla a varios usuarios registrados en el sistema. A la nota se le debe asignar unas coordenadas GPS que podrán ser seleccionadas mediante la posición actual o mediante un visor de AR.
- **Obtener usuarios:** Permite obtener una lista de los usuarios registrados en el sistema a los cuales se les podrá enviar la nota.
- **Obtener coordenadas de dispositivo:** Permite obtener las coordenadas GPS actuales en donde se encuentre el dispositivo móvil.
- **Obtener coordenadas de visor AR:** Permite obtener las coordenadas GPS del visor de realidad aumentada utilizado. Desde este visor también se obtendrá el texto de la nota a enviar.

4.1.2.2 Caso de uso de Escenario 2: Buscar notas

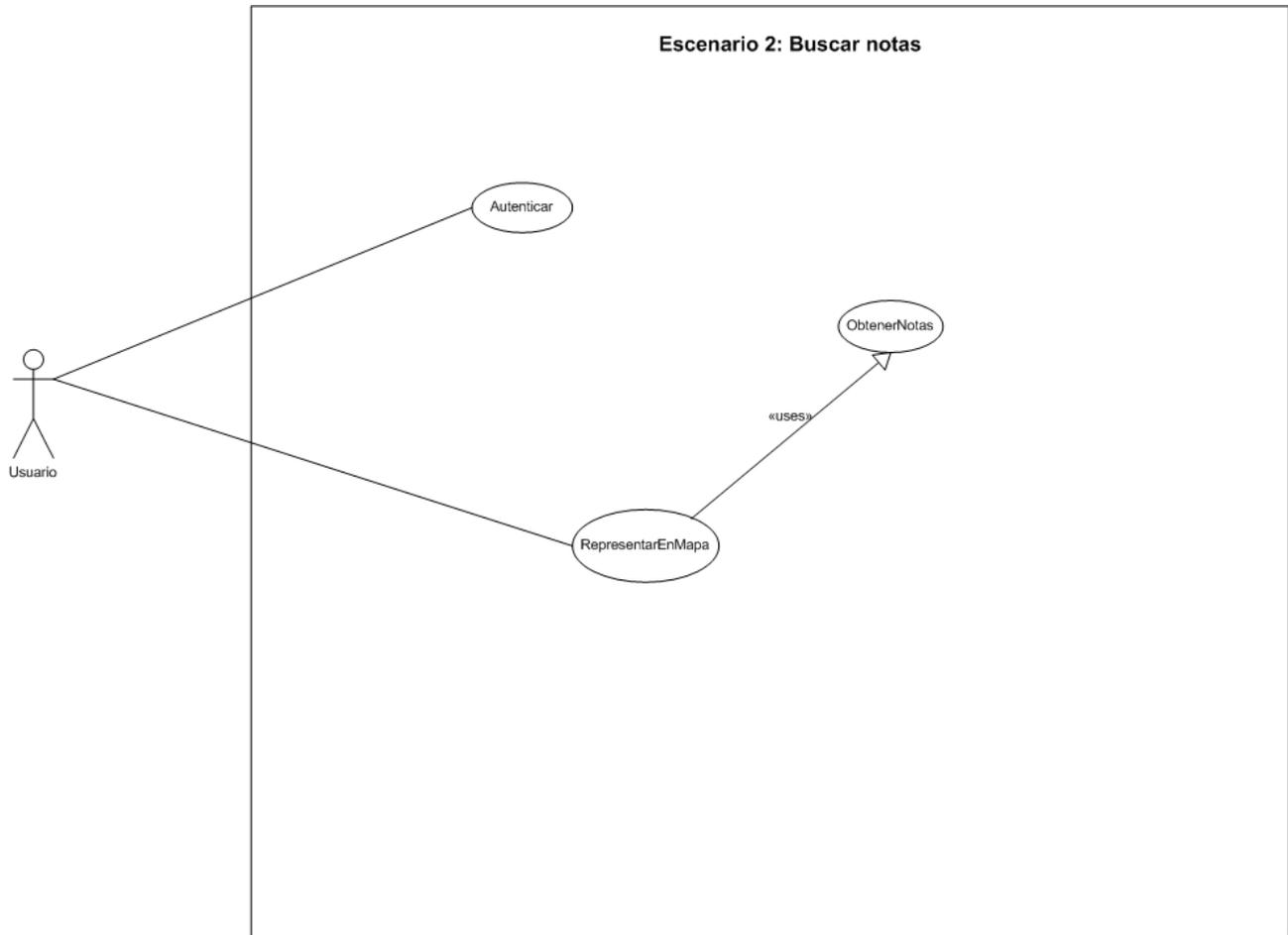


Fig 4. Caso de uso de Buscar Notas.

4.1.2.2.1 Funcionalidades

- **Autenticar:** Permite el acceso de un usuario registrado en el sistema.
- **Obtener notas:** Permite obtener una lista de las notas que tiene asignadas un usuario.
- **Representar en mapa:** Permite al usuario representar las notas obtenidas en un mapa para su localización.

4.1.2.3 Caso de uso de Escenario 3: Visualizar notas

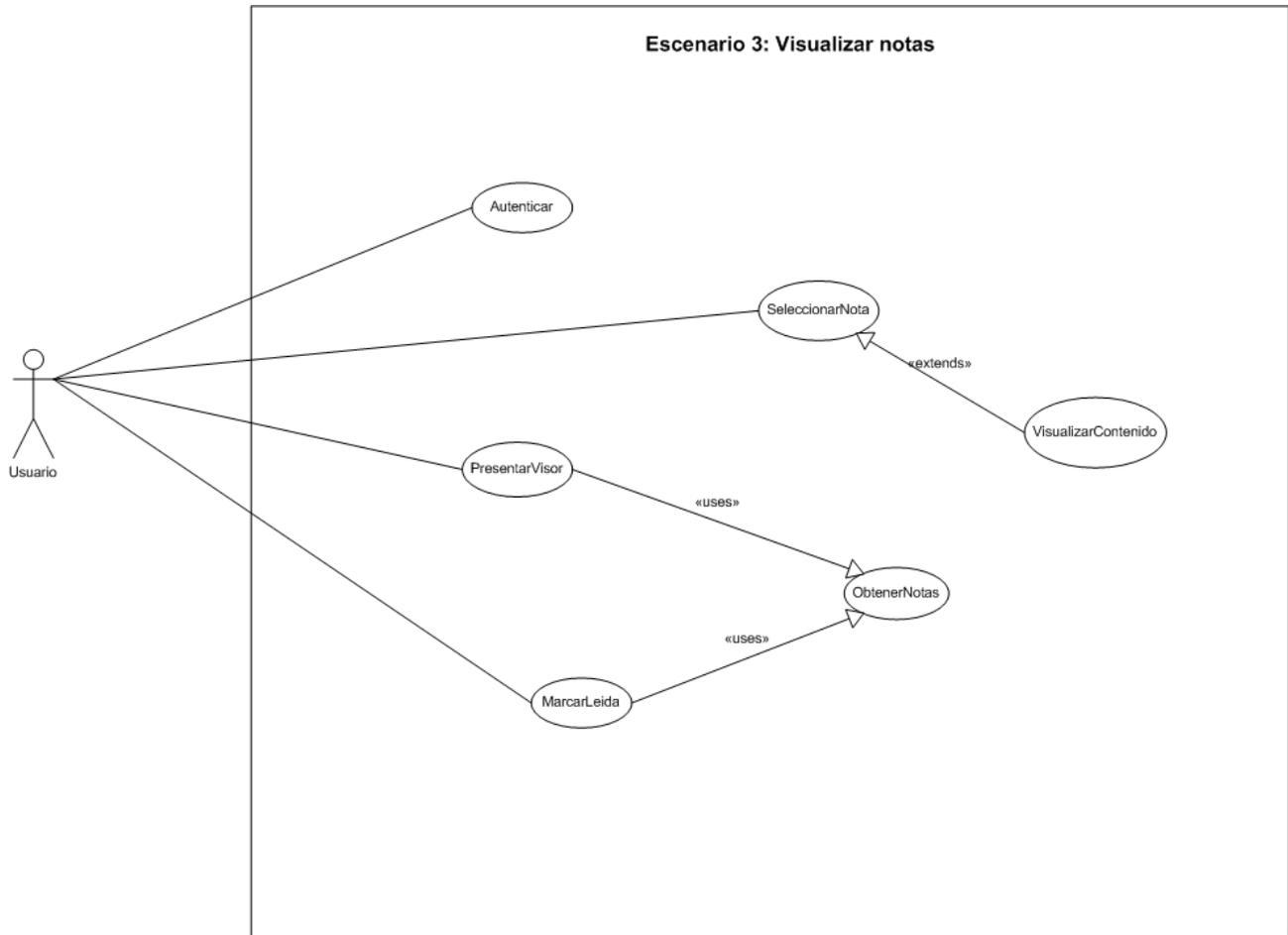


Fig 5. Caso de uso de Visualizar Notas.

4.1.2.3.1 Funcionalidades

- **Autenticar:** Permite el acceso de un usuario registrado en el sistema.
- **Seleccionar nota:** Permite seleccionar una nota asignada del visualizador ARViewer para su visualización.
- **Presentar visor:** Mediante la captura de señales de los sensores, se presentará la información (notas) en el visualizado ARViewer.

- **Marcar leída:** Permite marcar/desmarca las notas que han sido leídas desde una lista de notas asignadas. Las notas marcadas como leídas no so representables en el mapa ni en el visor de AR.

4.1.3 Especificación de requerimientos no funcionales

4.1.3.1 Requerimientos generales de usuario

- Se deberá disponer de un dispositivo smartphone o tablet con plataforma Android 2.2 (Froyo) o superior (ver apartado [2.4 Dispositivos móviles con Android](#)).
- El dispositivo deberá disponer de conexión a internet.

4.1.3.2 Arquitectura del sistema

La siguiente figura muestra la arquitectura del sistema a implementar y que permitirá la interacción del dispositivo móvil con el sistema servidor que almacena las notas de los usuarios registrados.

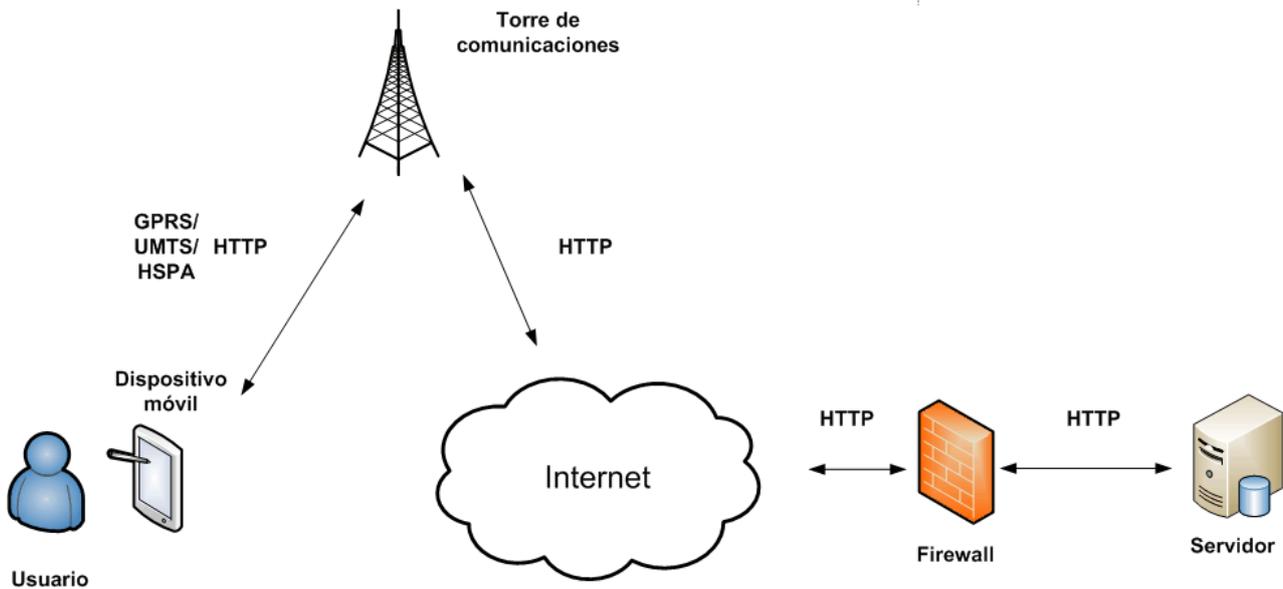


Fig 6. Arquitectura del sistema.

4.1.3.3 Operativa del sistema

Un usuario, mediante un dispositivo móvil (ver [requerimientos generales](#)), ejecuta la aplicación ARNotes4U! El usuario realiza operaciones de registro de usuario, alta de notas, etc. (ver [la especificación inicial de requerimientos funcionales](#)), mediante peticiones HTTP [Wiki 13]. Estas peticiones serán encaminadas, en primera instancia, mediante redes de radio de tecnología de transmisión de paquetes (GPRS/UMTS/HSPA) [Wiki 14], gestionadas por un operador de telefonía que nos proporciona el servicio y que, posteriormente, las encauzará a la red Internet (ISP) [Wiki 15]. Internet, mediante diferentes elementos de enrutamiento, encamina las peticiones hasta entregarlas al Servidor. El Servidor atiende las peticiones, ejecutando los servicios solicitados, generando una respuesta. Esta respuesta realiza el camino inverso al que fue realizado por la petición del usuario del dispositivo móvil. Una vez que la respuesta es recibida por el dispositivo móvil, ésta es procesada obteniendo los resultados del Servidor.

4.1.3.4 Consideraciones del sistema

- El dispositivo móvil se conectará a Internet utilizando tecnología inalámbrica de transmisión de radio (GPRS/UMTS/HSPA).
- Las peticiones del dispositivo serán realizadas mediante protocolo HTTP.
- El sistema Servidor dispondrá de un servidor de aplicaciones Apache Tomcat (<http://tomcat.apache.org/>) y los servicios serán proporcionados mediante tecnología Java Servlet [Wiki 16].
- El gestor de base de datos gestionado por los servicios del Servidor de aplicaciones será el Oracle Express Edition (<http://www.oracle.com/technetwork/database/express-edition/overview/index.html>).

4.2 Diseño

En este punto se analizarán los diseños más importantes del sistema mediante la realización de algunos [casos de uso documentados](#) en los escenarios anteriores. La realización de los casos de uso va a consistir en el diseño de diagramas de secuencia que expongan las interacciones entre las diferentes clases del modelo de negocio a implementar y que conforman cada caso de uso.

4.2.1 Realizaciones de casos de uso

4.2.1.1 Realización de casos de uso del escenario 1

4.2.1.1.1 Diagrama de secuencia de CrearNota

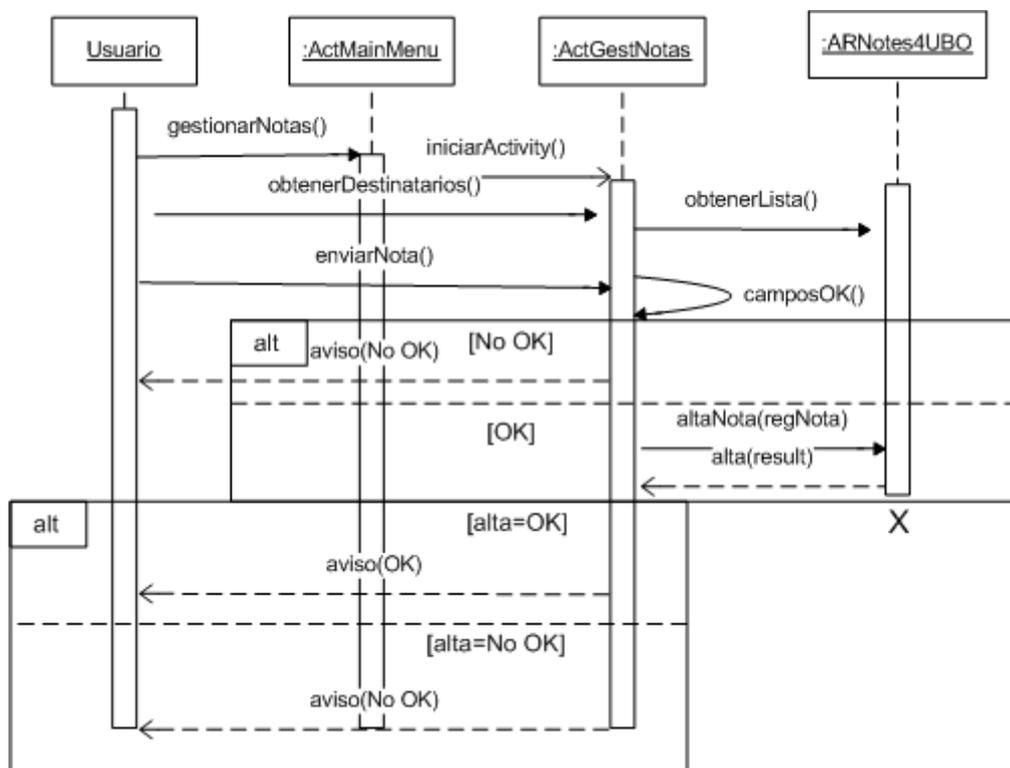


Fig. 7. Diagrama de secuencia de CrearNota.

Descripción:

Un usuario desde el menú principal de la aplicación invoca la pantalla de gestión de notas. Una vez que ha cumplimentado los campos realiza una petición para obtener una lista desde donde podrá seleccionar los destinatarios de la nota. Posteriormente realiza una petición para registrar las notas en el servidor. Si alguno de los campos obligatorios no han sido debidamente cumplimentados,

se dará una aviso al usuario indicando este hecho y la nota no será enviada. Si los campos están correctamente cumplimentados se realizará una petición para registrar la nota en el servidor a través de una clase de negocio especializada. El resultado de la operación de registro es informado al usuario.

4.2.1.1.2 Diagrama de secuencia de CrearNota desde visor ARViewer

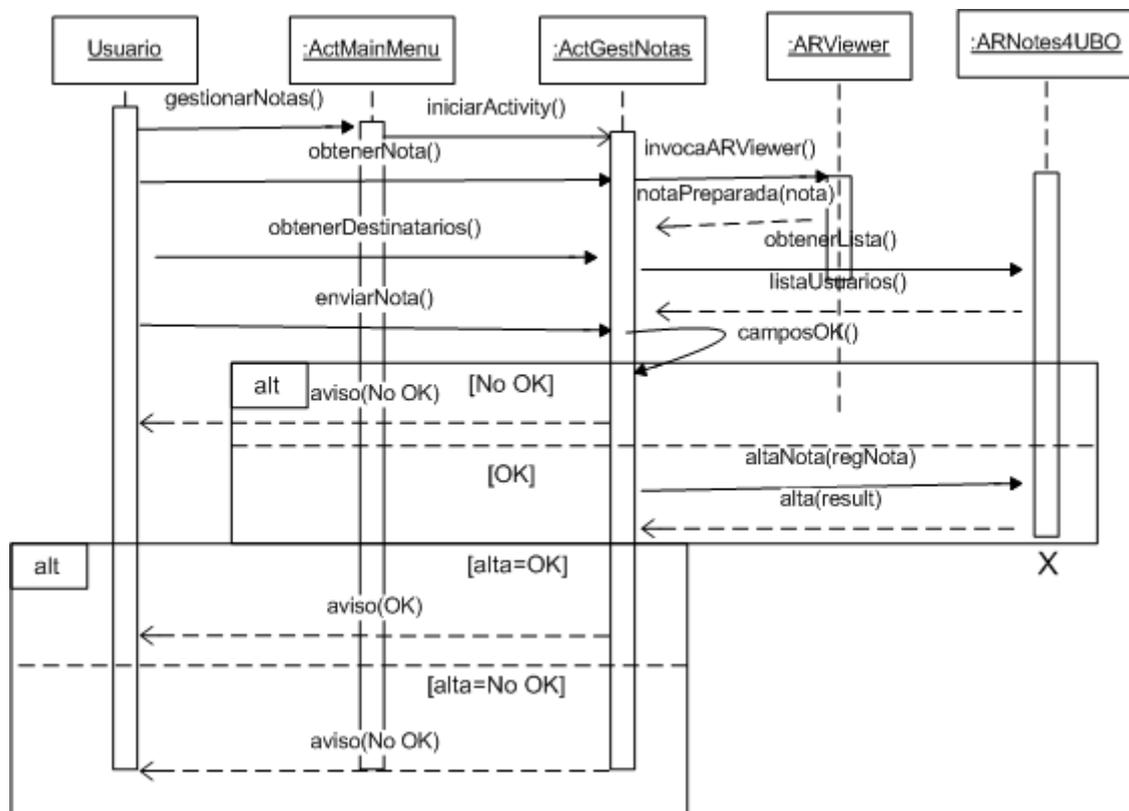


Fig. 8. Diagrama de secuencia de CrearNota con ARViewer.

Descripción:

Un usuario desde el menú principal de la aplicación invoca la pantalla de gestión de notas. Para cumplimentar los campos de la nota realiza una petición al visor de realidad aumentada ARViewer. Una vez ha preparado la nota mediante el visor, ésta es devuelta a la pantalla de gestión de notas. Después de cumplimentarse los campos obligatorios mediante la nota devuelta por el visor de forma automática, se debe realiza una petición para obtener una lista desde donde se podrá seleccionar los destinatarios de la nota. Posteriormente realiza una petición para registrar las notas en el servidor. Si alguno de los campos obligatorios no han sido debidamente cumplimentados se dará una aviso al usuario indicando este hecho y la nota no será enviada. Si los campos están correctamente cumplimentados se realizará una petición para registrar la nota en el servidor a través de una clase de negocio especializada. El resultado de la operación de registro es informado al usuario.

4.2.1.2 Realización de casos de uso del escenario 2

4.2.1.2.1 Diagrama de secuencia de RepresentarEnMapa

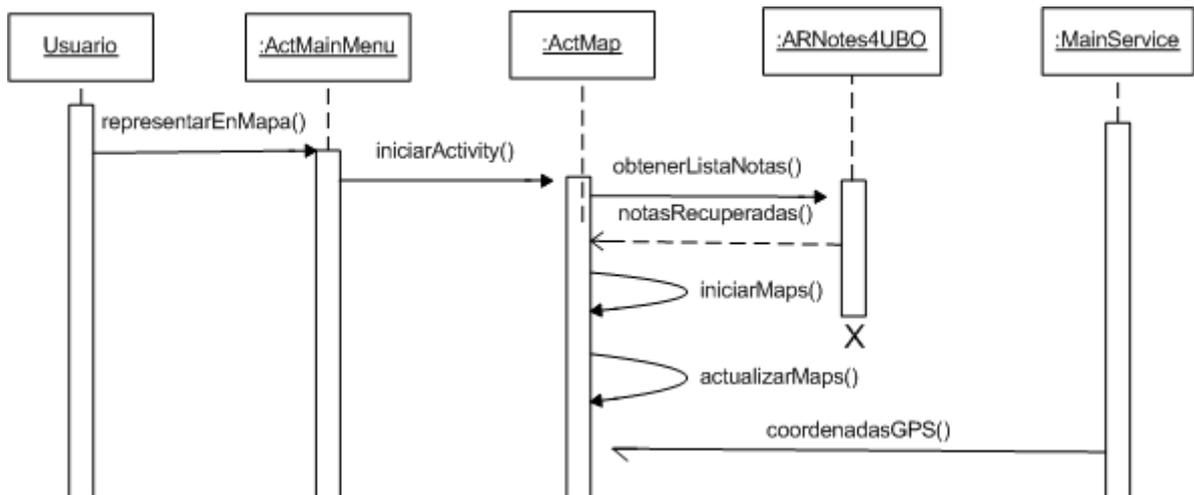


Fig. 9. Diagrama de secuencia de RepresentarEnMapa.

Descripción:

Un usuario desde el menú principal de la aplicación invoca la pantalla de representación las notas en un mapa. La pantalla de representación de notas en mapa realiza una petición para obtener las notas del usuario logado en el sistema y que se encuentren en estado no leída. Una vez que las notas han sido recuperadas son proporcionadas a maps para que las represente. La pantalla de visualización de maps está constantemente siendo informada cada vez que el dispositivo obtiene unas nuevas coordenadas GPS. Estas coordenadas son utilizadas para actualizar la porción de mapa a visualizar. El usuario, una vez dispone de maps, puede navegar a través de él en busca de las notas asignadas. El mapa constantemente regresará a nuestra posición actual. Esto permite que si vamos caminando, el mapa nos irá enseñando nuestro entorno, a la vez que nos permite movernos por él para localizar las notas.

4.2.1.3 Realización de casos de uso del escenario 3

4.2.1.3.1 Diagrama de secuencia de PresentarVisor

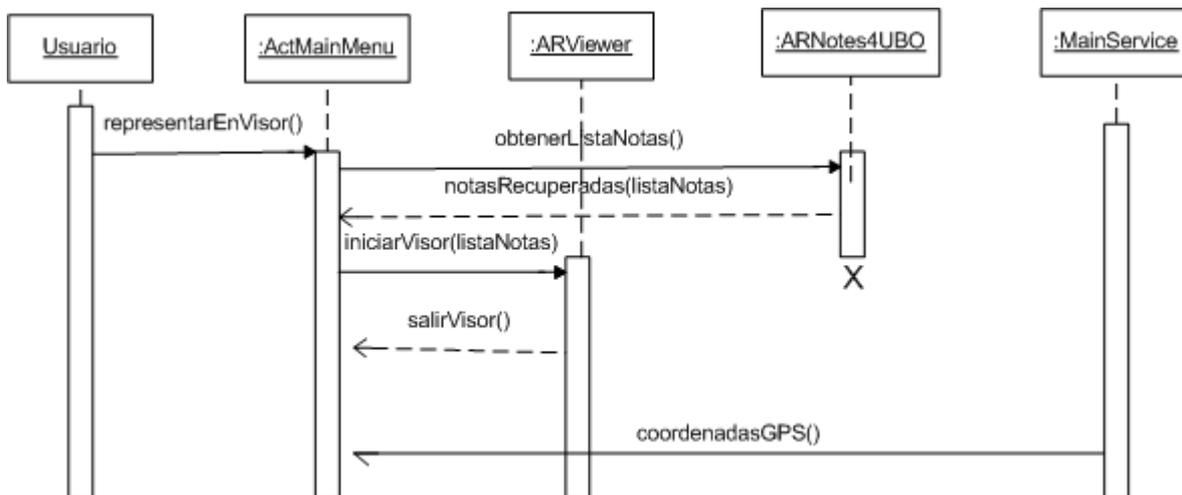


Fig. 10. Diagrama de secuencia de PresentarEnVisor.

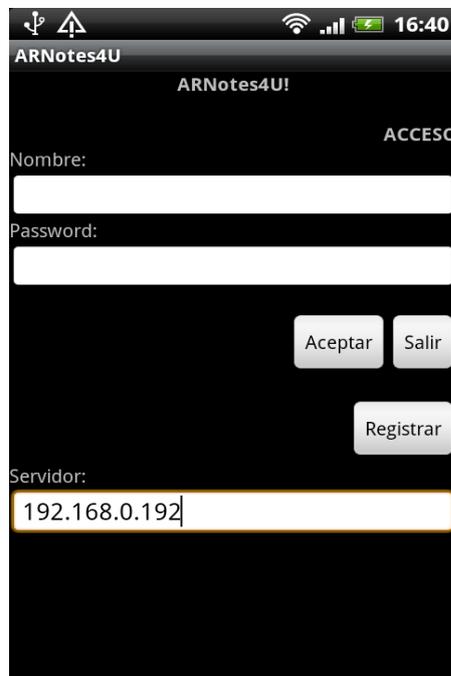
Descripción:

Un usuario desde el menú principal de la aplicación invoca la pantalla de representación las notas mediante el visor de AR. La pantalla de menú principal realiza una petición para obtener las notas del usuario logado en el sistema y que se encuentren en estado no leída. Una vez que las notas han sido recuperadas, son proporcionadas al visor de realidad aumentada ARViewer para que las visualice. Al pulsar el botón retroceder del dispositivo regresaremos a la pantalla de menú principal.

4.2.2 Diseño de pantallas

En los siguientes puntos se describen algunas las pantallas de la aplicación que deben ser implementadas.

4.2.2.1 Pantalla de Acceso



Pant. 1. Pantalla de Acceso.

4.2.2.1.1 Descripción de componentes

Control	Tipo	Descripción
Nombre:	Label	Etiqueta.
Password:	Label	Etiqueta.
Servidor:	Label	Etiqueta.
Editable (nombre)	EditText	El nombre del usuario que desea logarse.
Editable (password)	EditText	El password del usuario que desea logarse.
Editable (servidor)	EditText	Es el nombre de dominio o IP del servidor de gestión de notas.
Aceptar	Button	Invoca la acción de logar al usuario en el sistema.
Salir	Button	Pide confirmación para salir. Si se acepta, la aplicación nos devuelve al sistema.
Registrar	Button	Invoca la acción de registrar un nuevo usuario.

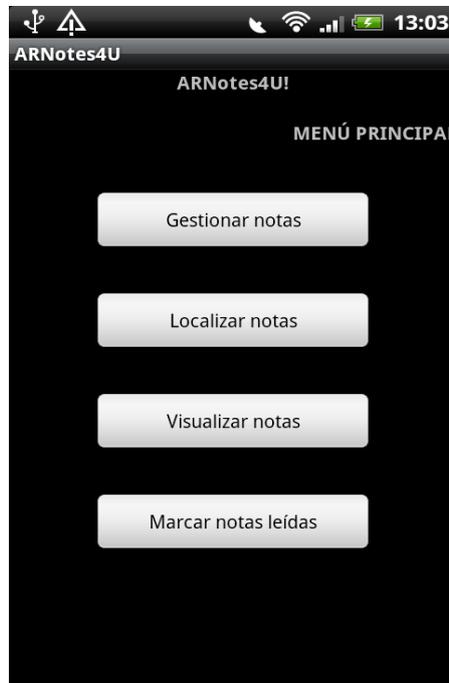
4.2.2.1.2 Descripción funcional

Al invocar la aplicación desde el dispositivo se nos muestra la pantalla de acceso. La pantalla dispone de unos campos para permitir el login de un usuario en el sistema.

En el campo (Servidor) debemos indicar la dirección IP o nombre de dominio del servidor gestor de notas.

Al introducir un usuario no registrado en el sistema, la aplicación nos avisará y nos propondrá darlo de alta mediante la pulsación del botón [Registrar]. Si el usuario existe en el sistema éste el logado accediendo al menú principal del sistema.

4.2.2.2 Pantalla de Menú Principal



Pant. 2. Pantalla de Menú Principal.

4.2.2.2.1 Descripción de componentes

Control	Tipo	Descripción
Gestionar notas	Button	Invoca la acción de acceder a la pantalla de gestión

		de notas.
Localizar notas	Button	Invoca la acción de acceder a la pantalla de localización de notas mediante maps.
Visualizar notas	Button	Invoca la acción de acceder a la pantalla de localización de notas mediante ARViewer.
Marcar notas leídas	Button	Invoca la acción de presentar una lista de selección/deselección de notas.

4.2.2.2 Descripción funcional

Después de habernos logado correctamente desde la pantalla Acceso, se nos muestra la pantalla del menú principal de la aplicación. En ella hay una botonera que nos va a permitir realizar las funciones que en ella se describen. Esta pantalla siempre es presentada una vez se vuelve de algunas de las opciones disponibles.

4.2.2.3 Pantalla de Gestión de Notas



Pant. 3. Pantalla de Gestión de Notas.

4.2.2.3.1 Descripción de componentes

Control	Tipo	Descripción
Título:	Label	Etiqueta.
Nota:	Label	Etiqueta.
Posición de la nota	Label	Etiqueta.
Longitud:	Label	Etiqueta.
Latitud:	Label	Etiqueta.

Editable (título)	EditText	Es el título que se le dará a la nota.
Editable (Nota)	EditText	Es el cuerpo de la nota.
Editable (longitud)	EditText	Longitud de las coordenadas GPS obtenidas.
Editable (latitud)	EditText	latitud de las coordenadas GPS obtenidas.
Destinatarios	Button	Invoca una lista de posibles destinatarios a enviar la nota.
Volver	Button	Regresa a la pantalla de menú principal.
Nueva nota	Button	Limpia todos los campos y los prepara para una nueva nota.
Enviar nota	Button	Invoca la acción de dar de alta la nota en el sistema y se la asigna a la lista de destinatarios.
Actual	Button	Actualiza los campos 'longitud' y 'latitud' con las coordenadas de la posición actual del dispositivo.
Obtener de ARViewer	Button	Invoca al visor de AR para la elaboración de la nota mediante la cámara del dispositivo.

4.2.2.3.2 Descripción funcional

Al pulsar el botón [Gestionar notas] desde el menú principal, se nos presenta la siguiente pantalla de gestión de notas. En ella hay cuatro campos que son obligatorios.

Para realizar una nota se debe facilitar: Un título, el cuerpo de la nota y las coordenadas GPS de longitud y latitud expresadas en grados. Si el dispositivo lo permite, se registra también la altitud.

Si pulsamos [Nueva nota] los campos son inicializados y preparados para introducir una nueva nota. Si pulsamos [Volver] regresaremos a la pantalla de Menú principal.

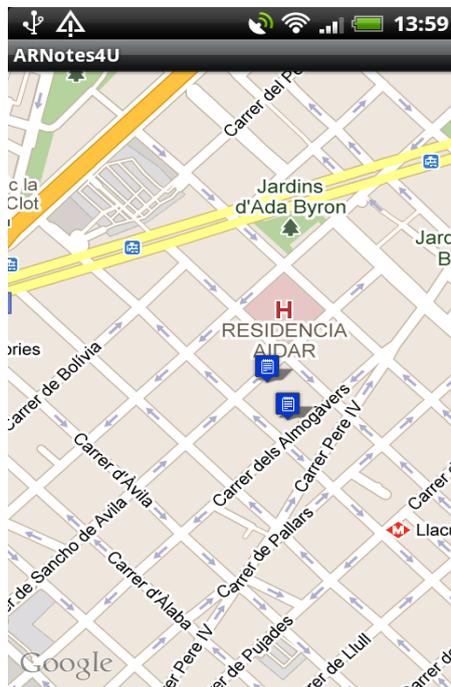
Al pulsar el botón [Enviar nota] con campos no cumplimentados, el sistema nos avisa y la nota no es enviada.

Es posible crear una nueva nota cumplimentado cada uno de los campos que presenta la pantalla. Al pulsar el botón [Actualizar], obtendremos las coordenadas GPS de nuestra posición actual y, serán en estas coordenadas, dónde situaremos la nota virtual a crear.

Al pulsar el botón [Destinatarios], obtendremos una lista de todos los usuarios registrados en el sistema. Pulsando sobre la lista podremos seleccionar/deseleccionar los posibles destinatarios. Para ocultar la lista con la selección realizada, deberemos pulsar sobre el botón de volver del dispositivo.

Al pulsar el botón [Obtener de ARViewer] es posible la creación de notas desde el visor de realidad aumentada ARViewer. Desde este entorno podremos crear notas, asignándolas directamente a una ubicación geoposicional registrada por el visor.

4.2.2.4 Pantalla de Localizar Notas



Pant. 4. Pantalla de Localizar Notas.

4.2.2.4.1 Descripción de componentes

Control	Tipo	Descripción
Visor de mapa	maps	Visualiza un mapa con las notas que se le suministran.

4.2.2.4.2 Descripción funcional

Al pulsar la opción [Localizar notas] del menú principal, se nos presentará un mapa, en dónde aparecerán marcadas todas las notas que tenemos asignadas y pendientes de leer (no las hemos marcado como leídas). El mapa constantemente regresará a nuestra

posición actual. Esto permite que si vamos caminando, el mapa nos irá enseñando nuestro entorno, a la vez que nos permite movernos por él para localizar las notas.

4.2.2.5 Pantalla de Visualizar Notas

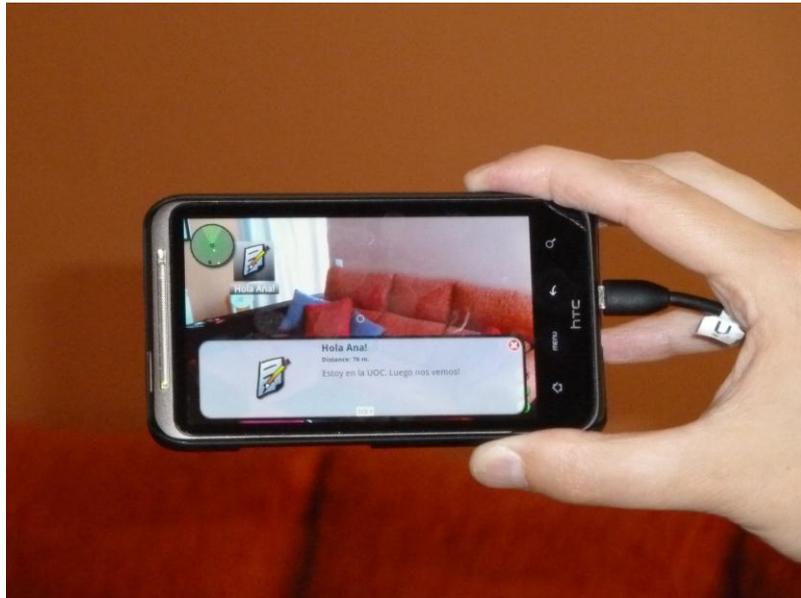


Foto. 1. Pantalla de visualizar Notas.

4.2.2.5.1 Descripción de componentes

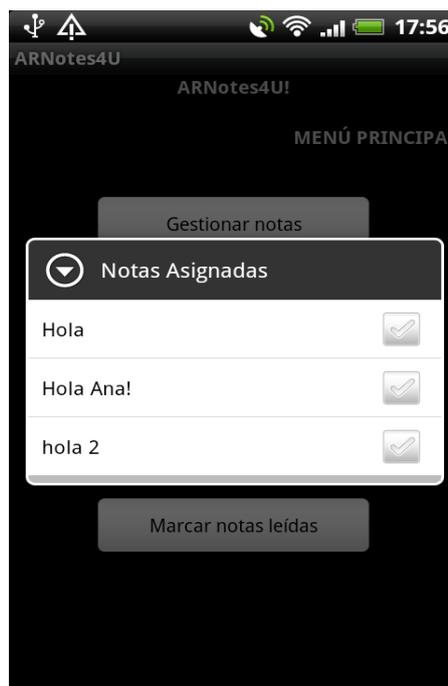
Control	Tipo	Descripción
Visor de AR ARViewer	ARViewer	Visualiza mediante la cámara del sistema, notas que se le suministren, situándolas en las coordenadas geoposicionales en las que fueron creadas.

4.2.2.5.2 Descripción funcional

Al pulsar la opción [Visualizar notas] del menú principal, se iniciará el visor de AR ARViewer, que nos mostrará en la pantalla, junto con imágenes reales, las notas que tengamos asignadas pendientes de leer (no han sido marcadas como leídas). La nota muestra el título en la parte inferior.

Al pulsar sobre la nota, en la parte inferior de la pantalla se nos mostrará el contenido de la misma.

4.2.2.6 Pantalla de Marcar Notas



Pant. 5. Pantalla de Marcar Notas.

4.2.2.6.1 Descripción de componentes

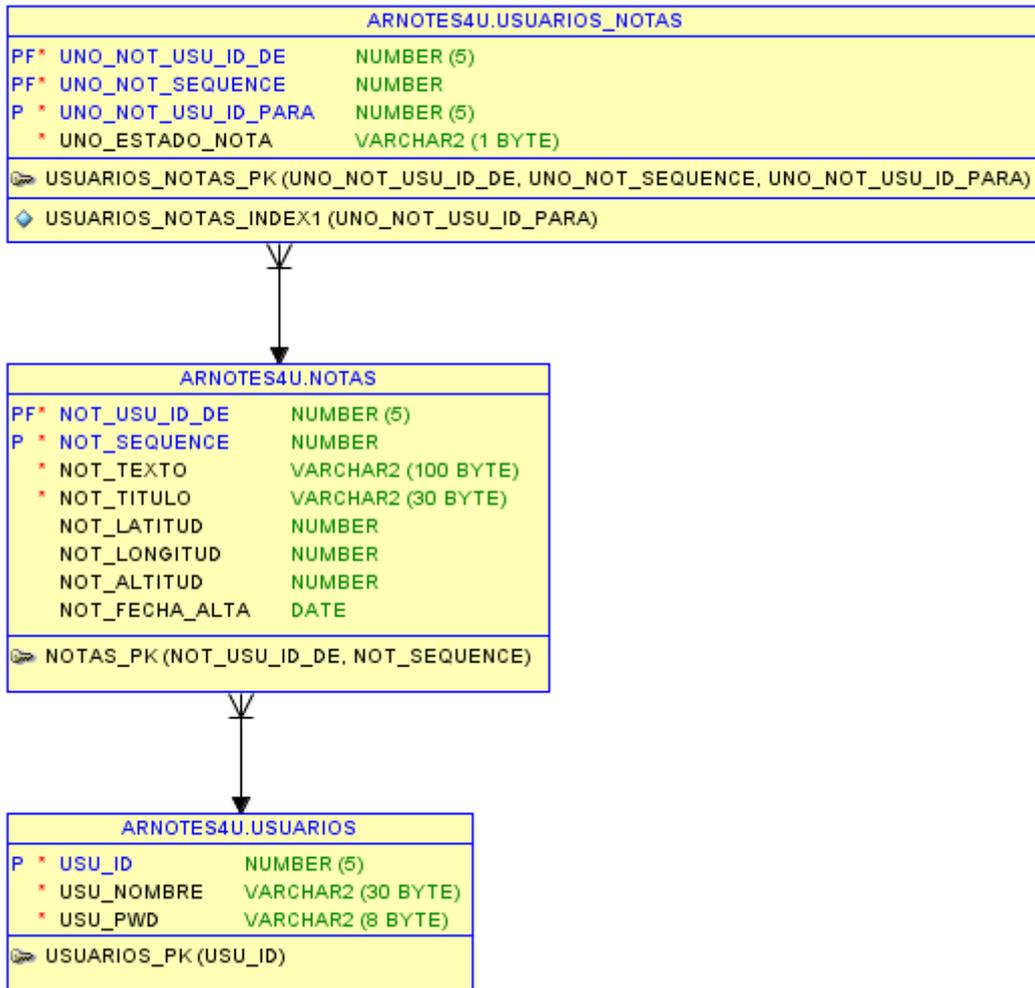
Control	Tipo	Descripción
Notas Asignadas	AlertDialog	Muestra una lista de nota asignadas y permite marcar/desmarcar para cambiarles el estado de leída a no leída.

4.2.2.6.2 Descripción funcional

Al pulsar la opción [Marcar notas leídas] del menú principal, se nos presentará una lista con todas las notas que tenemos asignadas (sólo los títulos). Mediante esta lista dispondremos de la opción de marcarlas como leídas o no. Si una nota se marca como leída, ésta ya no podrá ser consultada, al no aparecer en el mapa ni en el visor de AR. Las notas que han sido marcadas podrán volver a ser desmarcadas para su posterior consulta.

4.2.3 Modelo de base de datos

El siguiente diagrama entidad-relación [Wiki 17] muestra las tablas que forman parte del modelo de base de datos que da soporte a la gestión de usuarios y notas.



Diag. 1. Diagrama entidad-relación del modelo de datos.

4.2.3.1 Entidad USUARIOS

Es la entidad que almacena la información relacionada con los usuarios registrados en el sistema. Dispone de un ID (primary key) que es alimentado por una secuencia, un nombre y un password.

4.2.3.2 Entidad NOTAS

Es la entidad que almacena la información relacionada con las notas dadas de alta en el sistema. Dispone de un ID formado por el identificador del usuario que crea la nota más una secuencia. El resto de campos son específicos de una nota, como son: el título, el texto, la latitud, longitud y altitud. Finalmente dispone de una fecha que registra cuándo se da la nota de alta en el sistema.

4.2.3.3 Entidad USUARIOS_NOTAS

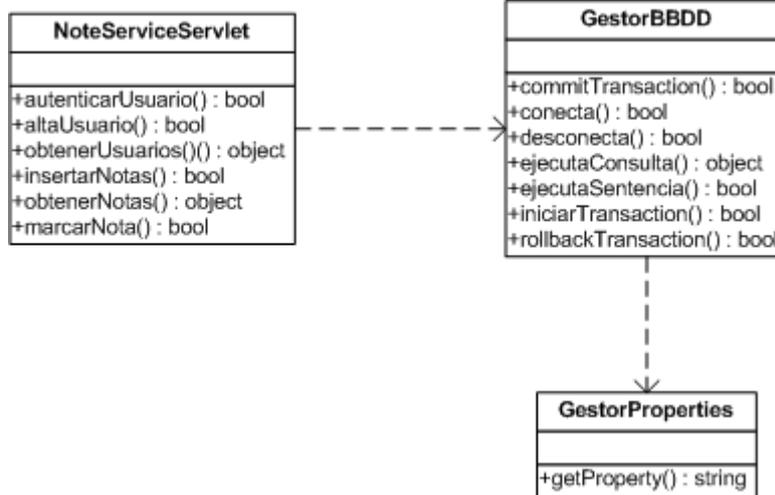
Esta entidad es la que va a permitir asociar diferentes usuarios a una misma nota. Para ello, para una misma nota creada por un usuario (UNO_NOT_USU_ID_DE, UNO_NOT_SEQUENCE) podemos asignarle varios usuarios (UNO_NOT_USU_ID_PARA).

Es esta entidad la que dispone del atributo que indica si una determinada nota ha sido leída (UNO_ESTADO_NOTA), pues es posible que usuarios diferentes dispongan, para una misma nota, estados diferentes.

4.2.4 Desarrollo de servidor

Para la realización del BackOffice se debe desarrollar una serie de servicios de servidor accesibles desde el dispositivo móvil. Estos servicios se construirán con tecnología Servelt de Java y deberán poder acceder a una base de datos relacional en dónde se dispondrá del [modelo de base de datos descrito](#).

La estructura básica de estos servicios deberá ser la siguiente:



Diag. 2. Diagrama estático de clases de la aplicación de servidor.

4.2.4.1 Descripción funcional

Existe una clase servlet que recibe las peticiones de los dispositivos móviles. En las peticiones realizadas se deberá indicar qué servicio se desea ejecutar, de los disponibles en la clase. La clase servlet hace uso de una clase gestora de base de datos para resolver las diferentes peticiones que debe servir. Las operaciones disponibles en la clase gestora de base de datos son las clásicas para realizar consultas o ejecutar operaciones de manipulación de datos (insert, update). También dispone de los mecanismos necesarios para el control de transacciones, pudiendo deshacer un conjunto de operaciones si éstas no se realiza correctamente de forma atómica.

5 Conclusiones

El presente trabajo ha consistido en el estudio del sistema operativo Android, en dónde se han expuesto los elementos característicos del sistema y su organización interna. También se han comentado algunos modelos de dispositivos móviles que incorporan Android y que a día de hoy presentan las tecnologías más avanzadas (como pueden ser la diversidad de sensores), potenciando las funcionalidades del sistema. Posteriormente se ha realizado un proyecto, denominado ARNotes4U, que ejemplifica algunos conceptos novedosos soportados por Android, como es el Realidad Aumentada.

Concluir comentando que el desarrollo de la aplicación ARNotes4U ha sido muy trabajoso, pues es mi primera incursión en OS Android y en su framework de trabajo. Pero también ha sido muy gratificante pues, de igual forma, también ha sido mi primer contacto con la Realidad Aumentada.

Muchas de las dudas que se me presentaron fueron resueltas principalmente mediante la Web de desarrolladores de Android: <http://developer.android.com/index.html>

También me gustaría agradecer a Raúl Román, componente del equipo desarrollador del visor ARViewer, el haber respondido a las cuestiones que le realicé respecto a su funcionamiento (<http://www.libregeosocial.org/node/3>) de forma totalmente desinteresada.

6 Glosario

API: Es la especificación de uso de un determinado conjunto de recursos software (http://en.wikipedia.org/wiki/Application_programming_interface).

CLDC: La especificación CLDC (Connected Limited Device Configuration) define el conjunto básico de librerías, interfaces (core) y máquina virtual disponibles en un entorno J2ME.

Customizar: Realizar una configuración no estándar sobre un producto base.

IDE: Entorno integrado que dispone de todas las herramientas necesarias para el desarrollo de aplicaciones (http://en.wikipedia.org/wiki/Integrated_development_environment).

J2ME: Java 2 Micro Edition es la plataforma Java para el desarrollo de aplicaciones embebidas en dispositivos con recursos limitados, como por ejemplo teléfonos móviles.

Layout: Recurso sobre el cuál se realiza el diseño gráfico o especificación de una pantalla de usuario (http://en.wikipedia.org/wiki/Layout_%28computing%29).

MIDP: El MIDP (Mobile Information Device Profile) es el perfil o framework que hace uso de la especificación CLDC para proporcionar un entorno de desarrollo y marco de ejecución de aplicaciones para dispositivos móviles como teléfonos y PDA's.

POIs: Los puntos de interés (POIs) son información estructurada utilizada por sistemas GIS y GPS para localizar/situar información de interés en un mapa.

RA: La RA (Realidad Aumentada) es una tecnología que combina imágenes reales capturadas mediante cámara de video con imágenes sintéticas creadas por computador para añadir información y/o interactuar con ellas.

WEB: La World Wide Web o, también conocida como Web, es un sistema distribuido de gestión de información accesible desde Internet (http://en.wikipedia.org/wiki/World_Wide_Web).

7 Bibliografía

Todas las consultas online fueron realizadas, en un momento u otro, durante el transcurso de desarrollo del proyecto, entre los meses de marzo y junio del 2011.

[Layar 1]: *Augmented Reality* [on line], Layar Inc., <<http://www.layar.com/>>.

[Apac 1]: *Apache Tomcat* [on line], The Apache Software Foundation, <<http://tomcat.apache.org/>> .

[Orac 1]: *Oracle Database 10g Express Edition* [on line], Oracle Inc., <<http://www.oracle.com/technetwork/database/express-edition/overview/index.html>>.

[Libr 1]: ARViewer [on line], *LibreGeoSocial*, <<http://www.libregeosocial.org/node/24>>.

[Will 1]: *A Who's Who Guide to Open Source Augmented Reality Apps* [on line], Nathan Willis, linux.com, <<http://www.linux.com/learn/tutorials/418753-a-whos-who-guide-to-open-source-augmented-reality-apps>>.

[Mixa 1]: *Augmented Reality Engine* [on line], Mixare Inc., <<http://www.mixare.org/>>.

[Lin 1]: *Linux* [on line], The Linux Foundation, <<http://www.linux.com/>>.

[Wiki 1]: *Google* [on line], Wikimedia Foundation, <<http://en.wikipedia.org/wiki/Google>>.

[Goo 1]: *Open Source Project* [on line], Google Inc., <<http://source.android.com/about/index.html>>.

[OHA 1]: *Open Handset Alliance* [on line], Open Handset Alliance Group, <<http://www.openhandsetalliance.com/>>.

[Wiki 2]: *Software* [on line], Wikimedia Foundation, <<http://es.wikipedia.org/wiki/Software>>.

[PCMA 1]: *Software stack* [on line], pcmag.com, Enciclopedia,
<http://www.pcmag.com/encyclopedia_term/0,2542,t=software+stack&i=51702,00.asp>.

[Wiki 3]: *Middleware* [on line], Wikimedia Foundation,
<<http://en.wikipedia.org/wiki/Middleware>>.

[Wiki 4]: *Scheduling (computing)* [on line], Wikimedia Foundation,
<http://en.wikipedia.org/wiki/Scheduling_%28computing%29>.

[Wiki 5]: *Máquina virtual Java* [on line], Wikimedia Foundation,
<http://es.wikipedia.org/wiki/M%C3%A1quina_virtual_Java>.

[Wiki 6]: *Dalvik (software)* [on line], Wikimedia Foundation,
<http://en.wikipedia.org/wiki/Dalvik_%28software%29>.

[Wiki 7]: *Software framework* [on line], Wikimedia Foundation,
<http://en.wikipedia.org/wiki/Software_framework>.

[Goo 2]: *Skia, 2D Graphics Library* [on line], Google Inc.,
<<http://code.google.com/p/skia/>>.

[Khro 1]: *OpenGL* [on line], Khronos Group, <<http://www.opengl.org/>>.

[SQLi 1]: *SQLite database* [on line], SQLite Consortium , <<http://www.sqlite.org/>>.

[Wiki 8]: *Component-based software engineering* [on line], Wikimedia Foundation,
<http://en.wikipedia.org/wiki/Component-based_software_engineering>.

[Wiki 9]: *Smartphone* [on line], Wikimedia Foundation,
<<http://en.wikipedia.org/wiki/Smartphone>>.

[Wiki 10]: *Tablet computer* [on line], Wikimedia Foundation,
<http://en.wikipedia.org/wiki/Tablet_computer>.

[Wiki 11]: *Accelerometer* [on line], Wikimedia Foundation,
<<http://en.wikipedia.org/wiki/Accelerometer>>.

[Wiki 12]: *AMOLED* [on line], Wikimedia Foundation,
<<http://es.wikipedia.org/wiki/AMOLED>>.

[Wiki 13]: *Hypertext Transfer Protocol* [on line], Wikimedia Foundation,
<http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol>.

[Wiki 14]: *Universal Mobile Telecommunications System* [on line], Wikimedia
Foundation,
<http://en.wikipedia.org/wiki/Universal_Mobile_Telecommunications_System>.

[Wiki 15]: *Internet service provider* [on line], Wikimedia Foundation,
<http://en.wikipedia.org/wiki/Internet_service_provider>.

[Wiki 16]: *Java Servlet* [on line], Wikimedia Foundation,
<http://en.wikipedia.org/wiki/Java_Servlet>.

[Wiki 17]: *Entity-relationship model* [on line], Wikimedia Foundation,
<http://en.wikipedia.org/wiki/Entity-relationship_model>.